INTRODUCTION TO SOCIAL NETWORK METHODS

Robert Hanneman & Mark Riddle
University of California, Riverside

LibreTexts™

University of California, Riverside

Introduction to Social Network Methods

Robert Hanneman & Mark Riddle

The LibreTexts libraries are Powered by NICE CXOne and are supported by the Department of Education Open Textbook Pilot Project, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

TABLE OF CONTENTS

1: Social Network Data

2: Why Formal Methods?

3: Using Graphs to Represent Social Relations

4: Working with Netdraw to Visualize Graphs

5: Using Matrices to Represent Social Relations

6: Working with Network Data

Detailed Licensing

Licensing

A detailed breakdown of this resource's licensing can be found in ***Back Matter/Detailed Licensing***.

CHAPTER OVERVIEW

1: Social Network Data

1.1: Introduction - What's different about social network data?

On one hand, there really isn't anything about social network data that is all that unusual. Social network analysts do use a specialized language for describing the structure and contents of the sets of observations that they use. But, network data can also be described and understood using the ideas and concepts of more familiar methods, like cross-sectional survey research.

On the other hand, the data sets that social network analysts develop usually end up looking quite different from the conventional rectangular data array so familiar to survey researchers and statistical analysts. The differences are quite important because they lead us to look at our data in a different way -- and even lead us to think differently about how to apply statistics.

"Conventional" social science data consist of a rectangular array of measurements. The rows of the array are the cases, or subjects, or observations. The columns consist of scores (quantitative or qualitative) on attributes, or variables, or measures. A simple example is shown as figure 1.1. Each cell of the array then describes the score of some actor (row) on some attribute (column). In some cases, there may be a third dimension to these arrays, representing panels of observations or multiple groups.

Figure 1.1. Example of rectangular data array

Name	***Sex***	***Age***	***In-Degree***
Bob	Male	32	2
Carol	Female	27	1
Ted	Male	29	1
Alice	Female	28	3

The fundamental data structure is one that leads us to compare how actors are similar or dissimilar to each other across attributes (by comparing rows). Or, perhaps more commonly, we examine how variables are similar or dissimilar to each other in their distributions across actors (by comparing or correlating columns).

"Network" data (in their purest form) consist of a square array of measurements. The rows of the array are the cases, or subjects, or observations. The columns of the array are -- and note the key difference from conventional data -- the same set of cases, subjects, or observations. In each cell of the array describes a relationship between the actors. A simple example is shown as figure 1.2, which describes the network of friendship relations among four people.

Figure 1.2. Example of square array of network data

Who reports liking whom?				
	Choice:			
Chooser:	Bob	Carol	Ted	Alice
Bob	---	0	1	1

Carol	1	---	0	1
Ted	0	1	---	1
Alice	1	0	0	---

We could look at this data structure the same way as with attribute data. By comparing rows of the array, we can see which actors are similar to which other actors in whom they choose. By looking at the columns, we can see who is similar to whom in terms of being chosen by others. These are useful ways to look at the data, because they help us to see which actors have similar positions in the network. This is the first major emphasis of network analysis: seeing how actors are located or "embedded" in the overall network.

But a network analyst is also likely to look at the data structure in a second way -- holistically. The analyst might note that there are about equal numbers of ones and zeros in the matrix. This suggests that there is a moderate "density" of liking overall. The analyst might also compare the cells above and below the diagonal to see if there is reciprocity in choices (e.g. Bob chose Ted, did Ted choose Bob?). This is the second major emphasis of network analysis: seeing how the whole pattern of individual choices gives rise to more holistic patterns.

It is quite possible to think of the network data set in the same terms as "conventional data." One can think of the rows as simply a listing of cases, and the columns as attributes of each actor (i.e. the relations with other actors can be thought of as "attributes" of each actor). Indeed, many of the techniques used by network analysts (like calculating correlations and distances) are applied exactly the same way to network data as they would be to conventional data.

While it is possible to describe network data as just a special form of conventional data (and it is), network analysts look at the data in some rather fundamentally different ways. Rather than thinking about how an actor's ties with other actors describes the attributes of "ego," network analysts instead see a structure of connections, within which the actor is embedded. Actors are described by their relations, not by their attributes. And, the relations themselves are just as fundamental as the actors that they connect.

The major difference between conventional and network data is that conventional data focuses on actors and attributes; network data focus on actors and relations. The difference in emphasis is consequential for the choices that a researcher must make in deciding on research design, in conducting sampling, developing measurement, and handling the resulting data. It is not that the research tools used by network analysts are different from those of other social scientists (they mostly are not). But the special purposes and emphases of network research do call for some different considerations.

In this chapter, we will take a look at some of the issues that arise in design, sampling, and measurement for social network analysis. Our discussion will focus on the two parts of network data: nodes (or actors) and edges (or relations). We will try to show some of the ways in which network data are similar to, and different from more familiar actor by attribute data. We will introduce some new terminology that makes it easier to describe the special features of network data. Lastly, we will briefly discuss how the differences between network and actor-attribute data are consequential for the application of statistical tools.

1.2: Nodes

Network data are defined by actors and by relations (or "nodes" and "edges"). The nodes or actors part of network data would seem to be pretty straight-forward. Other empirical approaches in the social sciences also think in terms of cases or subjects or sample elements and the like. There is one difference with most network data, however, that makes a big difference in how such data are usually collected -- and the kinds of samples and populations that are studied.

Network analysis focuses on the relations among actors, and not individual actors and their attributes. This means that the actors are usually not sampled independently, as in many other kinds of studies (most typically, surveys). Suppose we are studying friendship ties, for example. John has been selected to be in our sample. When we ask him, John identifies seven friends. We need to track down each of those seven friends and ask them about their friendship ties, as well. The seven friends are in our sample because John is (and vice-versa), so the "sample elements" are no longer "independent."

The nodes or actors included in non-network studies tend to be the result of independent probability sampling. Network studies are much more likely to include all of the actors who occur within some (usually naturally occurring) boundary. Often network studies don't use "samples" at all, at least in the conventional sense. Rather, they tend to include all of the actors in some population or populations. Of course, the populations included in a network study may be a sample of some larger set of populations. For example, when we study patterns of interaction among students in a classrooms, we include all of the children in a classroom (that is, we study the whole population of the classroom). The classroom itself, though, might have been selected by probability methods from a population of classrooms (say all of those in a school).

The use of whole populations as a way of selecting observations in (many) network studies makes it important for the analyst to be clear about the boundaries of each population to be studied, and how individual units of observation are to be selected within that population. Network data sets also frequently involve several levels of analysis, with actors embedded at the lowest level (i.e. network designs can be described using the language of "nested" designs).

Populations, samples, and boundaries

Social network analysts rarely draw samples in their work. Most commonly, network analysts will identify some population and conduct a census (i.e. include all elements of the population as units of observation). A network analyst might examine all of the nouns and objects occurring in a text, all of the persons at a birthday party, all members of a kinship group, of an organization, neighborhood, or social class (e.g. landowners in a region, or royalty).

Survey research methods usually use a quite different approach to deciding which nodes to study. A list is made of all nodes (sometimes stratified or clustered), and individual elements are selected by probability methods. The logic of the method treats each individual as a separate "replication" that is, in a sense, interchangeable with any other.

Because network methods focus on relations among actors, actors cannot be sampled independently to be included as observations. If one actor happens to be selected, then we must also include all other actors to whom our ego has (or could have) ties. As a result, network approaches tend to study whole populations by means of census, rather than by sample (we will discuss a number of exceptions to this shortly, under the topic of sampling ties).

The populations that network analysts study are remarkably diverse. At one extreme, they might consist of symbols in texts or sounds in verbalizations; at the other extreme, nations in the world system of states might constitute the population of nodes. Perhaps most common, of course, are populations of individual persons. In each case, however, the elements of the population to be studied are defined by falling within some boundary.

The boundaries of the populations studied by network analysts are of two main types. Probably most commonly, the boundaries are those imposed or created by the actors themselves. All the members of a classroom, organization, club, neighborhood, or community can constitute a population. These are naturally occurring clusters, or networks. So, in a sense, social network studies often draw the boundaries around a population that is known, *a priori*, to be a network. Alternatively, a network analyst might take a more "demographic" or "ecological" approach to defining population boundaries. We might draw observations by contacting all of the people who are found in a bounded spatial area, or who meet some criterion (having gross family incomes over $1,000,000 per year). Here, we might have reason to suspect that networks exist, but the entity being studied is an abstract aggregation imposed by the investigator -- rather than a pattern of institutionalized social action that has been identified and labeled by its participants.

Network analysts can expand the boundaries of their studies by replicating populations. Rather than studying one neighborhood, we can study several. This type of design (which could use sampling methods to select populations) allows for replication and for testing of hypotheses by comparing populations. A second, and equally important way that network studies expand their scope is by the inclusion of multiple levels of analysis, or modalities.

Modality and levels of analysis

The network analyst tends to see individual people nested within networks of face-to-face relations with other persons. Often these networks of interpersonal relations become "social facts" and take on a life of their own. A family, for example, is a network of close relations among a set of people. But this particular network has been institutionalized and given a name and reality beyond that of its component nodes. Individuals in their work relations may be seen as nested within organizations; in their leisure relations they may be nested in voluntary associations. Neighborhoods, communities, and even societies are, to varying degrees, social entities in and of themselves. And, as social entities, they may form ties with the individuals nested within them, and with other social entities.

Often network data sets describe the nodes and relations among nodes for a single bounded population. If I study the friendship patterns among students in a classroom, I am doing a study of this type. But a classroom exists within a school - which might be thought of as a network relating classes and other actors (principals, administrators, librarians, etc.). And most schools exist within school districts, which can be thought of as networks of schools and other actors (school boards, research wings, purchasing and personnel departments, etc.). There may even be patterns of ties among school districts (say by the exchange of students, teachers, curricular materials, etc.).

Most social network analysts think of individual persons as being embedded in networks that are embedded in networks that are embedded in networks. Network analysts describe such structures as "multi-modal." In our school example, individual students and teachers form one mode, classrooms a second, schools a third, and so on. A data set that contains information about two types of social entities (say persons and organizations) is a two mode network.

Of course, this kind of view of the nature of social structures is not unique to social network analystst. Statistical analysts deal with the same issues as "hierarchical" or "nested" designs. Theorists speak of the macro-meso-micro levels of analysis, or develop schema for identifying levels of analysis (individual, group, organization, community, institution, society, global order being perhaps the most commonly used system in sociology). One advantage of network thinking and method is that it naturally predisposes the analyst to focus on multiple levels of analysis simultaneously. That is, the network analyst is always interested in how the individual is embedded within a structure and how the structure emerges from the micro-relations between individual parts. The ability of network methods to map such multi-modal relations is, at least potentially, a step forward in rigor.

Having claimed that social network methods are particularly well suited for dealing with multiple levels of analysis and multi-modal data structures, it must immediately be admitted that social network analysis rarely actually takes much advantage. Most network analyses does move us beyond simple micro or macro reductionism -- and this is good. Few, if any, data sets and analyses, however, have attempted to work at more than two modes simultaneously. And, even when working with two modes, the most common strategy is to examine them more or less separately (one exception to this is the conjoint analysis of two mode networks). In chapter 17, we'll take a look at some methods for multi-mode networks.

1.3: Relations

The other half of the design of network data has to do with what ties or relations are to be measured for the selected nodes. There are two main issues to be discussed here. In many network studies, all of the ties of a given type among all of the selected nodes are studied -- that is, a census is conducted. But, sometimes different approaches are used (because they are less expensive, or because of a need to generalize) that sample ties. There is also a second kind of sampling of ties that always occurs in network data. Any set of actors might be connected by many different kinds of ties and relations (e.g. students in a classroom might like or dislike each other, they might play together or not, they might share food or not, etc.). When we collect network data, we are usually selecting, or sampling, from among a set of kinds of relations that we might have measured.

Sampling ties

Given a set of actors or nodes, there are several strategies for deciding how to go about collecting measurements on the relations among them. At one end of the spectrum of approaches are "full network" methods. This approach yields the maximum of information, but can also be costly and difficult to execute, and may be difficult to generalize. At the other end of the spectrum are methods that look quite like those used in conventional survey research. These approaches yield considerably less information about network structure, but are often less costly, and often allow easier generalization from the observations in the sample to some larger population. There is no one "right" method for all research questions and problems.

Full network methods require that we collect information about each actor's ties with all other actors. In essence, this approach is taking a census of ties in a population of actors -- rather than a sample. For example we could collect data on shipments of copper between all pairs of nation states in the world system from International Monetary Fund records; we could examine the boards of directors of all public corporations for overlapping directors; we could count the number of vehicles moving between all pairs of cities; we could look at the flows of e-mail between all pairs of employees in a company; we could ask each child in a play group to identify their friends.

Because we collect information about ties between all pairs or dyads, full network data give a complete picture of relations in the population. Most of the special approaches and methods of network analysis that we will discuss in the remainder of this text were developed to be used with full network data. Full network data is necessary to properly define and measure many of the structural concepts of network analysis (e.g. between-ness).

Full network data allows for very powerful descriptions and analyses of social structures. Unfortunately, full network data can also be very expensive and difficult to collect. Obtaining data from every member of a population, and having every member rank or rate every other member can be very challenging tasks in any but the smallest groups. The task is made more manageable by asking respondents to identify a limited number of specific individuals with whom they have ties. These lists can then be compiled and cross-connected. But, for large groups (say all the people in a city), the task is practically impossible.

In many cases, the problems are not quite as severe as one might imagine. Most persons, groups, and organizations tend to have limited numbers of ties -- or at least limited numbers of strong ties. This is probably because social actors have limited resources, energy, time, and cognitive capacity -- and cannot maintain large numbers of strong ties. It is also true that social structures can develop a considerable degree of order and solidarity with relatively few connections.

Snowball methods begin with a focal actor or set of actors. Each of these actors is asked to name some or all of their ties to other actors. Then, all the actors named (who were not part of the original list) are tracked down and asked for some or all of their ties. The process continues until no new actors are identified, or until we decide to stop (usually for reasons of time and resources, or because the new actors being named are very marginal to the group we are trying to study).

The snowball method can be particularly helpful for tracking down "special" populations (often numerically small sub-sets of people mixed in with large numbers of others). Business contact networks, community elites, deviant sub-cultures, avid stamp collectors, kinship networks, and many other structures can be pretty effectively located and described by snowball methods. It is sometimes not as difficult to achieve closure in snowball "samples" as one might think. The limitations on the numbers of strong ties that most actors have, and the tendency for ties to be reciprocated often make it fairly easy to find the boundaries.

There are two major potential limitations and weaknesses of snowball methods. First, actors who are not connected (i.e. "isolates") are not located by this method. The presence and numbers of isolates can be a very important feature of populations for some analytic purposes. The snowball method may tend to overstate the "connectedness" and "solidarity" of populations of actors. Second, there is no guaranteed way of finding all of the connected individuals in the population. Where does one start the snowball rolling? If we start in the wrong place or places, we may miss whole sub-sets of actors who are connected -- but not attached to our starting points.

Snowball approaches can be strengthened by giving some thought to how to select the initial nodes. In many studies, there may be a natural starting point. In community power studies, for example, it is common to begin snowball searches with the chief executives of large economic, cultural, and political organizations. While such an approach will miss most of the community (those who are "isolated" from the elite network), the approach is very likely to capture the elite network quite effectively.

Ego-centric networks (with alter connections)

In many cases it will not be possible (or necessary) to track down the full networks beginning with focal nodes (as in the snowball method). An alternative approach is to begin with a selection of focal nodes (egos), and identify the nodes to which they are connected. Then, we determine which of the nodes identified in the first stage are connected to one another. This can be done by contacting each of the nodes; sometimes we can ask ego to report which of the nodes that it is tied to are tied to one another.

This kind of approach can be quite effective for collecting a form of relational data from very large populations, and can be combined with attribute-based approaches. For example, we might take a simple random sample of male college students and ask them to report who are their close friends, and which of these friends know one another. This kind of approach can give us a good and reliable picture of the kinds of networks (or at least the local neighborhoods) in which individuals are embedded. We can find out such things as how many connections nodes have, and the extent to which these nodes are close-knit groups. Such data can be very useful in helping to understand the opportunities and constraints that ego has as a result of the way they are embedded in their networks.

The ego-centered approach with alter connections can also give us some information about the network as a whole, though not as much as snowball or census approaches. Such data are, in fact, micro-network data sets -- samplings of local areas of larger networks. Many network properties -- distance, centrality, and various kinds of positional equivalence cannot be assessed with ego-centric data. Some properties, such as overall network density can be reasonably estimated with ego-centric data. Some properties -- such as the prevalence of reciprocal ties, cliques, and the like can be estimated rather directly.

Ego-centric networks (ego only)

Ego-centric methods really focus on the individual, rather than on the network as a whole. By collecting information on the connections among the actors connected to each focal ego, we can still get a pretty good picture of the "local" networks or "neighborhoods" of individuals. Such information is useful for understanding how networks affect individuals, and they also give a (incomplete) picture of the general texture of the network as a whole.

Suppose, however, that we only obtained information on ego's connections to alters -- but not information on the connections among those alters. Data like these are not really "network" data at all. That is, they cannot be represented as a square actor-by-actor array of ties. But doesn't mean that ego-centric data without connections among the alters are of no value for analysts seeking to take a structural or network approach to understanding actors. We can know, for example, that some actors have many close friends and kin, and others have few. Knowing this, we are able to understand something about the differences in the actors places in social structure, and make some predictions about how these locations constrain their behavior. What we cannot know from ego-centric data with any certainty is the nature of the macro-structure or the whole network.

In ego-centric networks, the alters identified as connected to each ego are probably a set that is unconnected with those for each other ego. While we cannot assess the overall density or connectedness of the population, we can sometimes be a bit more general. If we have some good theoretical reason to think about alters in terms of their social roles, rather than as individual occupants of social roles, ego-centered networks can tell us a good bit about local social structures. For example, if we identify each of the alters connected to an ego by a friendship relation as "kin," "co-worker," "member of the same church," etc., we can build up a picture of the networks of social positions (rather than the networks of individuals) in which egos are embedded. Such an approach, of course, assumes that such categories as "kin" are real and meaningful determinants of patterns of interaction.

Multiple relations

In a conventional actor-by-trait data set, each actor is described by many variables (and each variable is realized in many actors). In the most common social network data set of actor-by-actor ties, only one kind of relation is described. Just as we often are interested in multiple attributes of actors, we are often interested in multiple kinds of ties that connect actors in a network.

In thinking about the network ties among faculty in an academic department, for example, we might be interested in which faculty have students in common, serve on the same committees, interact as friends outside of the workplace, have one or more areas of expertise in common, and co-author papers. The positions that actors hold in the web of group affiliations are multi-faceted. Positions in one set of relations may re-enforce or contradict positions in another (I might share friendship ties with one set of people with whom I do not work on committees, for example). Actors may be tied together closely in one relational network, but be quite distant from one another in a different relational network. The locations of actors in multi-relational networks and the structure of networks composed of multiple relations are some of the most interesting (and still relatively unexplored) areas of social network analysis.

When we collect social network data about certain kinds of relations among actors we are, in a sense, sampling from a population of possible relations. Usually our research question and theory indicate which of the kinds of relations among actors are the most relevant to our study, and we do not sample -- but rather select -- relations. In a study concerned with economic dependency and growth, for example, I could collect data on the exchange of performances by musicians between nations -- but it is not really likely to be all that relevant.

If we do not know what relations to examine, how might we decide? There are a number of conceptual approaches that might be of assistance. Systems theory, for example, suggests two domains: material and informational. Material things are "conserved" in the sense that they can only be located at one node of the network at a time. Movements of people between organizations, money between people, automobiles between cities, and the like are all examples of material things which move between nodes -- and hence establish a network of material relations. Informational things, to the systems theorist, are "non-conserved" in the sense that they can be in more than one place at the same time. If I know something and share it with you, we both now know it. In a sense, the commonality that is shared by the exchange of information may also be said to establish a tie between two nodes. One needs to be cautious here, however, not to confuse the simple possession of a common attribute (e.g. gender) with the presence of a tie (e.g. the exchange of views between two persons on issues of gender).

Methodologies for working with multi-relational data are not as well developed as those for working with single relations. Many interesting areas of work such as network correlation, multi-dimensional scaling and clustering, and role algebras have been developed to work with multi-relational data. For the most part, these topics are beyond the scope of the current text, and are best approached after the basics of working with single relational networks are mastered. We will look at some methods for multi-relational (a.k.a. "multiplex" network data in chapter 16).

1.4: Scales of Measurement

Like other kinds of data, the information we collect about ties between actors can be measured (i.e. we can assign scores to our observations) at different "levels of measurement." The different levels of measurement are important because they limit the kinds of questions that can be examined by the researcher. Scales of measurement are also important because different kinds of scales have different mathematical properties, and call for different algorithms in describing patterns and testing inferences about them.

It is conventional to distinguish nominal, ordinal, and interval levels of measurement (the ratio level can, for all practical purposes, be grouped with interval). It is useful, however, to further divide nominal measurement into binary and multi-category variations; it is also useful to distinguish between full-rank ordinal measures and grouped ordinal measures. We will briefly describe all of these variations, and provide examples of how they are commonly applied in social network studies.

Binary measures of relations**:** By far the most common approach to scaling (assigning numbers to) relations is to simply distinguish between relations being absent (coded zero), and ties being present (coded one). If we ask respondents in a survey to tell us "which other people on this list do you like?" we are doing binary measurement. Each person from the list that is selected is coded one. Those who are not selected are coded zero.

Much of the development of graph theory in mathematics, and many of the algorithms for measuring properties of actors and networks have been developed for binary data. Binary data is so widely used in network analysis that it is not unusual to see data that are measured at a "higher" level transformed into binary scores before analysis proceeds. To do this, one simply selects some "cut point" and re-scores cases as below the cut-point (zero) or above it (one). Dichotomizing data in this way is throwing away information. The analyst needs to consider what is relevant (i.e. what is the theory about? is it about the presence and pattern of ties, or about the strengths of ties?), and what algorithms are to be applied in deciding whether it is reasonable to recode the data. Very often, the additional power and simplicity of analysis of binary data is "worth" the cost in information lost.

Multiple-category nominal measures of relations: In collecting data we might ask our respondents to look at a list of other people and tell us: "for each person on this list, select the category that describes your relationship with them the best: friend, lover, business relationship, kin, or no relationship." We might score each person on the list as having a relationship of type "1" type "2" etc. This kind of a scale is nominal or qualitative -- each person's relationship to the subject is coded by its type, rather than its strength. Unlike the binary nominal (true-false) data, the multiple category nominal measure is multiple choice.

The most common approach to analyzing multiple-category nominal measures is to use it to create a series of binary measures. That is, we might take the data arising from the question described above and create separate sets of scores for friendship ties, for lover ties, for kin ties, etc. This is very similar to "dummy coding" as a way of handling multiple choice types of measures in statistical analysis. In examining the resulting data, however, one must remember that each node was allowed to have a tie in at most one of the resulting networks. That is, a person can be a friendship tie or a lover tie -- but not both -- as a result of the way we asked the question. In examining the resulting networks, densities may be artificially low, and there will be an inherent negative correlation among the matrices.

This sort of multiple choice data can also be "binarized." That is, we can ignore what kind of tie is reported, and simply code whether a tie exists for a dyad, or not. This may be fine for some analyses -- but it does waste information. One might also wish to regard the types of ties as reflecting some underlying continuous dimension (for example, emotional intensity). The types of ties can then be scaled into a single grouped ordinal measure of tie strength. The scaling, of course, reflects the predispositions of the analyst -- not the reports of the respondents.

Grouped ordinal measures of relations: One of the earliest traditions in the study of social networks asked respondents to rate each of a set of others as "liked" "disliked" or "neutral." The result is a grouped ordinal scale (i.e., there can be more than one "liked" person, and the categories reflect an underlying rank order of intensity). Usually, this kind of three point scale was coded -1, 0, and +1 to reflect negative liking, indifference, and positive liking. When scored this way, the pluses and minuses make it fairly easy to write algorithms that will count and describe various network properties (e.g. the structural balance of the graph).

Grouped ordinal measures can be used to reflect a number of different quantitative aspects of relations. Network analysts are often concerned with describing the "strength" of ties. But, "strength" may mean (some or all of) a variety of things. One dimension is the frequency of interaction -- do actors have contact daily, weekly, monthly, etc. Another dimension is "intensity," which usually reflects the degree of emotional arousal associated with the relationship (e.g. kin ties may be infrequent, but carry a high "emotional charge" because of the highly ritualized and institutionalized expectations). Ties may be said to be stronger if they involve many different contexts or types of ties. Summing nominal data about the presence or absence of multiple types of ties gives rise to an ordinal (actually, interval) scale of one dimension of tie strength. Ties are also said to be stronger to the extent that they are reciprocated. Normally we would assess reciprocity by asking each actor in a dyad to report their feelings about the other. However, one might also ask each actor for their perceptions of the degree of reciprocity in a relation: Would you say that neither of you like each other very much, that you like X more than X likes you, that X likes you more than you like X, or that you both like each other about equally?

Ordinal scales of measurement contain more information than nominal. That is, the scores reflect finer gradations of tie strength than the simple binary "presence or absence." This would seem to be a good thing, yet it is frequently difficult to take advantage of ordinal data. The most commonly used algorithms for the analysis of social networks have been designed for binary data. Many have been adapted to continuous data -- but for interval, rather than ordinal scales of measurement. Ordinal data, consequently, are often binarized by choosing some cut-point and re-scoring. Alternatively, ordinal data are sometimes treated as though they really were interval. The former strategy has some risks, in that choices of cut-points can be consequential; the latter strategy has some risks, in that the intervals separating points on an ordinal scale may be very heterogeneous.

Full-rank ordinal measures of relations: Sometimes it is possible to score the strength of all of the relations of an actor in a rank order from strongest to weakest. For example, I could ask each respondent to write a "1" next to the name of the person in the class that you like the most, a "2" next to the name of the person you like next most, etc. The kind of scale that would result from this would be a "full rank order scale." Such scales reflect differences in degree of intensity, but not necessarily equal differences -- that is, the difference between my first and second choices is not necessarily the same as the difference between my second and third choices. Each relation, however, has a unique score (1st, 2nd, 3rd, etc.).

Full rank ordinal measures are somewhat uncommon in the social networks research literature, as they are in most other traditions. Consequently, there are relatively few methods, definitions, and algorithms that take specific and full advantage of the information in such scales. Most commonly, full rank ordinal measures are treated as if they were interval. There is probably somewhat less risk in treating fully rank ordered measures (compared to grouped ordinal measures) as though they were interval, though the assumption is still a risky one. Of course, it is also possible to group the rank order scores into groups (i.e. produce a grouped ordinal scale) or dichotomize the data (e.g. the top three choices might be treated as ties, the remainder as non-ties). In combining information on multiple types of ties, it is frequently necessary to simplify full rank order scales. But, if we have a number of full rank order scales that we may wish to combine to form a scale (i.e. rankings of people's likings of other in the group, frequency of interaction, etc.), the sum of such scales into an index is plausibly treated as a truly interval measure.

Interval measures of relations: The most "advanced" level of measurement allows us to discriminate among the relations reported in ways that allow us to validly state that, for example, "this tie is twice as strong as that tie." Ties are rated on scales in which the difference between a "1" and a "2" reflects the same amount of real difference as that between "23" and "24."

True interval level measures of the strength of many kinds of relationships are fairly easy to construct, with a little imagination and persistence. Asking respondents to report the details of the frequency or intensity of ties by survey or interview methods, however, can be rather unreliable -- particularly if the relationships being tracked are not highly salient and infrequent. Rather than asking whether two people communicate, one could count the number of email, phone, and inter-office mail deliveries between them. Rather than asking whether two nations trade with one another, look at statistics on balances of payments. In many cases, it is possible to construct interval level measures of relationship strength by using artifacts (e.g. statistics collected for other purposes) or observation.

Continuous measures of the strengths of relationships allow the application of a wider range of mathematical and statistical tools to the exploration and analysis of the data. Many of the algorithms that have been developed by social network analysts, originally for binary data, have been extended to take advantage of the information available in full interval measures. Whenever possible, connections should be measured at the interval level -- as we can always move to a less refined approach later; if data are collected at the nominal level, it is much more difficult to move to a more refined level.

Even though it is a good idea to measure relationship intensity at the most refined level possible, most network analysis does not operate at this level. The most powerful insights of network analysis, and many of the mathematical and graphical tools used by network analysts were developed for simple graphs (i.e. binary, undirected). Many characterizations of the embeddedness of actors in their networks, and of the networks themselves are most commonly thought of in discrete terms in the research literature. As a result, it is often desirable to reduce even interval data to the binary level by choosing a cutting -point, and coding tie strength above that point as "1" and below that point as "0." Unfortunately, there is no single "correct" way to choose a cut-point. Theory and the purposes of the analysis provide the best guidance. Sometimes examining the data can help (maybe the distribution of tie strengths really is discretely bi-modal, and displays a clear cut point; maybe the distribution is highly skewed and the main feature is a distinction between no tie and any tie). When a cut-point is chosen, it is wise to also consider alternative values that are somewhat higher and lower, and repeat the analyses with different cut-points to see if the substance of the results is affected. This can be very tedious, but it is very necessary. Otherwise, one may be fooled into thinking that a real pattern has been found, when we have only observed the consequences of where we decided to put our cut-point.

1.5: A note on Statistics and Social Network Data

Social network analysis is more a branch of "mathematical" sociology than of "statistical or quantitative analysis," though social network analysts most certainly practice both approaches. The distinction between the two approaches is not clear-cut. Mathematical approaches to network analysis tend to treat the data as "deterministic." That is, they tend to regard the measured relationships and relationship strengths as accurately reflecting the "real" or "final" or "equilibrium" status of the network. Mathematical types also tend to assume that the observations are not a "sample" of some larger population of possible observations; rather, the observations are usually regarded as the population of interest. Statistical analysts tend to regard the particular scores on relationship strengths as stochastic or probabilistic realizations of an underlying true tendency or probability distribution of relationship strengths. Statistical analysts also tend to think of a particular set of network data as a "sample" of a larger class or population of such networks or network elements -- and have a concern for the results of the current study would be reproduced in the "next" study of similar samples.

In the chapters that follow in this text, we will mostly be concerned with the "mathematical" rather than the "statistical" side of network analysis (again, it is important to remember that I am over-drawing the differences in this discussion). Before passing on to this, we should note a couple main points about the relationship between the material that you will be studying here, and the main statistical approaches in sociology. In chapter 18, we will explore some of the basic ways in which statistical tools have been adapted to study social network data.

In one way, there is little apparent difference between conventional statistical approaches and network approaches. Univariate, bi-variate, and even many multivariate descriptive statistical tools are commonly used in the describing, exploring, and modeling social network data. Social network data are, as we have pointed out, easily represented as arrays of numbers -- just like other types of sociological data. As a result, the same kinds of operations can be performed on network data as on other types of data. Algorithms from statistics are commonly used to describe characteristics of individual observations (e.g. the median tie strength of actor X with all other actors in the network) and the network as a whole (e.g. the mean of all tie strengths among all actors in the network). Statistical algorithms are very heavily used in assessing the degree of similarity among actors, and if finding patterns in network data (e.g. factor analysis, cluster analysis, multi-dimensional scaling). Even the tools of predictive modeling are commonly applied to network data (e.g. correlation and regression).

Descriptive statistical tools are really just algorithms for summarizing characteristics of the distributions of scores. That is, they are mathematical operations. Where statistics really become "statistical" is on the inferential side. That is, when our attention turns to assessing the reproducibility or likelihood of the pattern that we have described. Inferential statistics can be, and are, applied to the analysis of network data. But, there are some quite important differences between the flavors of inferential statistics used with network data, and those that are most commonly taught in basic courses in statistical analysis in sociology.

Probably the most common emphasis in the application of inferential statistics to social science data is to answer questions about the stability, reproducibility, or generalizability of results observed in a single sample. The main question is: if I repeated the study on a different sample (drawn by the same method), how likely is it that I would get the same answer about what is going on in the whole population from which I drew both samples? This is a really important question -- because it helps us to assess the confidence (or lack of it) that we ought to have in assessing our theories and giving advice.

To the extent the observations used in a network analysis are drawn by probability sampling methods from some identifyable population of actors and/or ties, the same kind of question about the generalizability of sample results applies. Often this type of inferential question is of little interest to social network researchers. In many cases, they are studying a particular network or set of networks, and have no interest in generalizing to a larger population of such networks (either because there isn't any such population, or we don't care about generalizing to it in any probabilistic way). In some other cases we may have an interest in generalizing, but our sample was not drawn by probability methods. Network analysis often relies on artifacts, direct observation, laboratory experiments, and documents as data sources -- and usually there are no plausible ways of identifying populations and drawing samples by probability methods.

The other major use of inferential statistics in the social sciences is for testing hypotheses. In many cases, the same or closely related tools are used for questions of assessing generalizability and for hypothesis testing. The basic logic of hypothesis testing is to compare an observed result in a sample to some null hypothesis value, relative to the sampling variability of the result under the assumption that the null hypothesis is true. If the sample result differs greatly from what was likely to have been observed under the assumption that the null hypothesis is true -- then the null hypothesis is probably not true.

The key link in the inferential chain of hypothesis testing is the estimation of the standard errors of statistics. That is, estimating the expected amount that the value a a statistic would "jump around" from one sample to the next simply as a result of accidents of sampling. We rarely, of course, can directly observe or calculate such standard errors -- because we don't have replications. Instead, information from our sample is used to estimate the sampling variability.

With many common statistical procedures, it is possible to estimate standard errors by well validated approximations (e.g. the standard error of a mean is usually estimated by the sample standard deviation divided by the square root of the sample size). These approximations, however, hold when the observations are drawn by independent random sampling. Network observations are almost always non-independent, by definition. Consequently, conventional inferential formulas do not apply to network data (though formulas developed for other types of dependent sampling may apply). It is particularly dangerous to assume that such formulas do apply, because the non-independence of network observations will usually result in under-estimates of true sampling variability -- and hence, too much confidence in our results.

The approach of most network analysts interested in statistical inference for testing hypotheses about network properties is to work out the probability distributions for statistics directly. This approach is used because: 1) no one has developed approximations for the sampling distributions of most of the descriptive statistics used by network analysts and 2) interest often focuses on the probability of a parameter relative to some theoretical baseline (usually randomness) rather than on the probability that a given network is typical of the population of all networks.

Suppose, for example, that I was interested in the proportion of the actors in a network who were members of cliques (or any other network statistic or parameter). The notion of a clique implies structure -- non-random connections among actors. I have data on a network of ten nodes, in which there are 20 symmetric ties among actors, and I observe that there is one clique containing four actors. The inferential question might be posed as: how likely is it, if ties among actors were purely random events, that a network composed of ten nodes and 20 symmetric ties would display one or more cliques of size four or more? If it turns out that cliques of size four or more in random networks of this size and degree are quite common, I should be very cautious in concluding that I have discovered "structure" or non-randomness. If it turns out that such cliques (or more numerous or more inclusive ones) are very unlikely under the assumption that ties are purely random, then it is very plausible to reach the conclusion that there is a social structure present.

But how can I determine this probability? The method used is one of simulation -- and, like most simulation, a lot of computer resources and some programming skills are often necessary. In the current case, I might use a table of random numbers to distribute 20 ties among 10 actors, and then search the resulting network for cliques of size four or more. If no clique is found, I record a zero for the trial; if a clique is found, I record a one. The rest is simple. Just repeat the experiment several thousand times and add up what proportion of the "trials" result in "successes." The probability of a success across these simulation experiments is a good estimator of the likelihood that I might find a network of this size and density to have a clique of this size "just by accident" when the non-random causal mechanisms that I think cause cliques are not, in fact, operating.

This may sound odd, and it is certainly a lot of work (most of which, thankfully, can be done by computers). But, in fact, it is not really different from the logic of testing hypotheses with non-network data. Social network data tend to differ from more "conventional" survey data in some key ways: network data are often not probability samples, and the observations of individual nodes are not independent. These differences are quite consequential for both the questions of generalization of findings, and for the mechanics of hypothesis testing. There is, however, nothing fundamentally different about the logic of the use of descriptive and inferential statistics with social network data.

The application of statistics to social network data is an interesting area, and one that is, at the time of this writing, at a "cutting edge" of research in the area. Since this text focuses on more basic and commonplace uses of network analysis, we won't have very much more to say about statistics beyond this point. You can think of much of what follows here as dealing with the "descriptive" side of statistics (developing index numbers to describe certain aspects of the distribution of relational ties among actors in networks). For those with an interest in the inferential side, a good place to start is with the second half of the excellent Wasserman and Faust textbook.

CHAPTER OVERVIEW

2: Why Formal Methods?

The basic idea of a social network is very simple. A social network is a set of actors (or points, or nodes, or agents) that may have relationships (or edges, or ties) with one another. Networks can have few or many actors, and one or more kinds of relations between pairs of actors. To build a useful understanding of a social network, a complete and rigorous description of a pattern of social relationships is a necessary starting point for analysis.

2.1: Introduction

The basic idea of a social network is very simple. A social network is a set of actors (or points, or nodes, or agents) that may have relationships (or edges, or ties) with one another. Networks can have few or many actors, and one or more kinds of relations between pairs of actors. To build a useful understanding of a social network, a complete and rigorous description of a pattern of social relationships is a necessary starting point for analysis. That is, ideally we will know about all of the relationships between each pair of actors in the population.

The amount of information that we need to describe even small social networks can be quite great. Managing these data, and manipulating them so that we can see patterns of social structure can be tedious and complicated. All of the tasks of social network methods are made easier by using tools from mathematics. For the manipulation of network data, and the calculation of indexes describing networks, it is most useful to record information as matrices. For visualizing patterns, graphs are often useful.

2.2: Efficiency

One reason for using mathematical and graphical techniques in social network analysis is to represent the descriptions of networks compactly and systematically. This also enables us to use computers to store and manipulate the information quickly and more accurately than we can by hand. For small populations of actors (e.g. the people in a neighborhood, or the business firms in an industry), we can describe the pattern of social relationships that connect the actors rather completely and effectively using words. To make sure that our description is complete, however, we might want to list all logically possible pairs of actors, and describe each kind of possible relationship for each pair. This can get pretty tedious if the number of actors and/or number of kinds of relations is large. Formal representations ensure that all the necessary information is systematically represented, and provides rules for doing so in ways that are much more efficient than lists.

2.3: Using computers

A related reason for using (particularly mathematical) formal methods for representing social networks is that mathematical representations allow us to apply computers to the analysis of network data. Why this is important will become clearer as we learn more about how structural analysis of social networks occurs. Suppose, for a simple example, we had information about trade-flows of 50 different commodities (e.g. coffee, sugar, tea, copper, bauxite) among the 170 or so nations of the world system in a given year. Here, the 170 nations can be thought of as actors or nodes, and the amount of each commodity exported from each nation to each of the other 169 can be thought of as the strength of a directed tie from the focal nation to the other. A social scientist might be interested in whether the "structures" of trade in mineral products are more similar to one another than, the structure of trade in mineral products are to vegetable products. To answer this fairly simple (but also pretty important) question, a huge amount of manipulation of the data is necessary. It could take, literally, years to do by hand; it can be done by a computer in a few minutes.

2.4: Seeing patterns

The third, and final reason for using "formal" methods (mathematics and graphs) for representing social network data is that the techniques of graphing and the rules of mathematics themselves suggest things that we might look for in our data — things that might not have occurred to us if we presented our data using descriptions in words. Again, allow me a simple example.

Suppose we were describing the structure of close friendship in a group of four people: Bob, Carol, Ted, and Alice. This is easy enough to do with words. Suppose that Bob likes Carol and Ted, but not Alice; Carol likes Ted, but neither Bob nor Alice; Ted likes all three of the other members of the group; and Alice likes only Ted (this description should probably strike you as being a description of a very unusual social structure).

We could also describe this pattern of liking ties with an actor-by-actor matrix where the rows represent choices by each actor. We will put in a "1" if an actor likes another, and a "0" if they don't. Such a matrix would look like figure 2.1.

Figure 2.1. Matrix representation of "liking" relation among four actors

	Bob	Carol	Ted	Alice
Bob	---	1	1	0
Carol	0	---	1	0
Ted	1	1	---	1
Alice	0	0	1	---

There are lots of things that might immediately occur to us when we see our data arrayed in this way, that we might not have thought of from reading the description of the pattern of ties in words. For example, our eye is led to scan across each row; we notice that Ted likes more people than Bob, than Alice and Carol. Is it possible that there is a pattern here? Are men are more likely to report ties of liking than women are (actually, research literature suggests that this is not generally true). Using a "matrix representation" also immediately raises a question: the locations on the main diagonal (e.g. Bob likes Bob, Carol likes Carol) are empty. Is this a reasonable thing? Or, should our description of the pattern of liking in the group include some statements about "self-liking"? There isn't any right answer to this question. My point is just that using a matrix to represent the pattern of ties among actors may let us see some patterns more easily, and may cause us to ask some questions (and maybe even some useful ones) that a verbal description doesn't stimulate.

2.5: Summary

There are three main reasons for using "formal" methods in representing social network data:

- Matrices and graphs are compact and systematic: They summarize and present a lot of information quickly and easily; and they force us to be systematic and complete in describing patterns of social relations.
- Matrices and graphs allow us to apply computers to analyzing data: This is helpful because doing systematic analysis of social network data can be extremely tedious if the number of actors or number of types of relationships among the actors is large. Most of the work is dull, repetitive, and uninteresting, but requires accuracy; exactly the sort of thing that computers do well, and we don't.
- Matrices and graphs have rules and conventions: Sometimes these are just rules and conventions that help us communicate clearly. But sometimes the rules and conventions of the language of graphs and mathematics themselves lead us to see things in our data that might not have occurred to us to look for if we had described our data only with words.

So, we need to learn the basics of representing social network data using matrices and graphs. The next several chapters (3, 4, 5, and 6) introduce these basic tools.

CHAPTER OVERVIEW

3: Using Graphs to Represent Social Relations

3.1: Introduction - Representing networks with graphs

Social network analysts use two kinds of tools from mathematics to represent information about patterns of ties among social actors: graphs and matrices. On this page, we we will learn enough about graphs to understand how to represent social network data. On the next page, we will look at matrix representations of social relations. With these tools in hand, we can understand most of the things that network analysts do with such data (for example, calculate precise measures of "relative density of ties").

There is a lot more to these topics than we will cover here; mathematics has whole sub-fields devoted to "graph theory" and to "matrix algebra." Social scientists have borrowed just a few things that they find helpful for describing and analyzing patterns of social relations.

A word of warning: there is a lot of specialized terminology here that you do need to learn. its worth the effort, because we can represent some important ideas about social structure in quite simple ways, once the basics have been mastered.

3.2: Graphs and Sociograms

There are lots of different kinds of "graphs." Bar-charts, pie-charts, line and trend charts, and many other things are called graphs and/or graphics. Network analysis uses (primarily) one kind of graphic display that consists of points (or nodes) to represent actors and lines (or edges) to represent ties or relations. When sociologists borrowed this way of graphing things from the mathematicians, they re-named their graphics "socio-grams." Mathematicians know the kind of graphic displays by the names of "directed graphs" "signed graphs" or simply "graphs."

There are a number of variations on the theme of socio-grams, but they all share the common feature of using a labeled circle for each actor in the population we are describing, and line segments between pairs of actors to represent the observation that a tie exists between the two. Let's suppose that we are interested in summarizing who nominates whom as being a "friend" in a group of four people (Bob, Carol, Ted, and Alice). We would begin by representing each actor as a "node" with a label (sometimes notes are represented by labels in circles or boxes). Figure 3.1 shows a graph with four labeled nodes, but no connections.

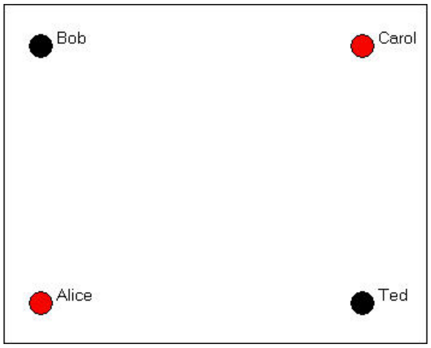


Figure 3.1. *Nodes for a simple graph*

In this example, we've also indicated an "attribute" of each actor by coloring the node (black for males, red for females). Coloring, shading, or different shapes and sizes are often used to represent attributes of the individual nodes.

We collected our data about friendship ties by asking each member of the group (privately and confidentially) who they regarded as "close friends" from a list containing each of the other members of the group. Each of the four people could choose none to all three of the others as "close friends." As it turned out, in our (fictitious) case, Bob chose Carol and Ted, but not Alice; Carol chose only Ted; Ted chose Bob and Carol and Alice; and Alice chose only Ted. We would represent this information by drawing an arrow from the chooser to each of the chosen, as in figure 3.2.

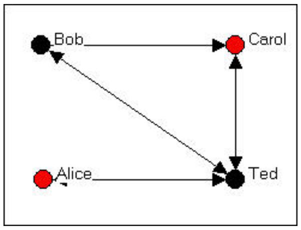


Figure 3.2*. A directed graph of friendship ties*

To reduce visual clutter, a double-headed arrow has been used when the relationship between two node is "reciprocated" (i.e. each actor chooses the other).

Let's suppose that we had also taken note of a second kind of relation - whether persons share the relationship "spouse" with one another. In our example, Bob and Carol are spouses, as are Ted and Alice. We can also represent this kind of a "bonded tie" with a directed graph as in figure 3.3.

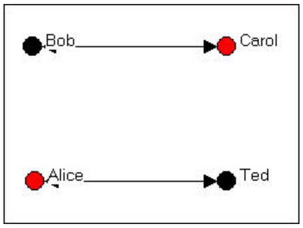


Figure 3.3. *A directed graph of spousal ties*

Where a tie is necessarily reciprocated (see the discussion of "bonded ties, below), a "simple" graph is often used instead of a "directed" graph. In a simple graph, relations are simply present of absent, and the relations are indicated by lines without arrow heads.

We can also represent multiple relations (multiplex relations) using graphs -- though with larger numbers of actors or many relations, the results may not be very easy to read. Let's combine the graphs of both "friendship" and "spousal" relations, as in figure 3.4.

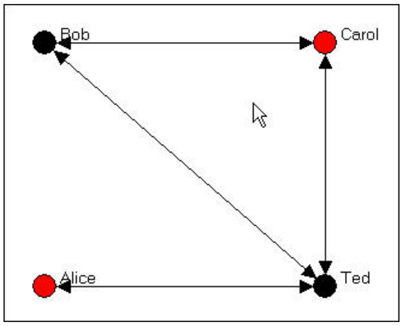


Figure 3.4. *A directed graph of multiplex relations (friendship and spouse)*

In this figure, a tie is shown between two nodes whenever there is either a friendship tie, or a spousal tie, or both. This helps us to see that Bob, Carol, and Ted form a "clique" (i.e. each is connected to each of the others), and Alice is a "pendant" (tied to the group by only one connection).

This particular way for drawing the multiplex relation, however, loses some information about which ties connect which actors. As an alternative, one might want to superimpose the two single-relation graphs, and show multiple lines (or different color lines, or some dashed lines) to show the different kinds of connections.

3.3: Kinds of Graphs

Now we need to introduce some terminology to describe different kinds of graphs. Figure 3.2 is an example of a *binary* (as opposed to a signed or ordinal or valued) and *directed* (as opposed to a co-occurrence or co-presence or bonded-tie) graph. Figure 3.3 is an example of a "co-occurrence" or "co-presence" or "bonded-tie" graph that is *binary* and *undirected* (or simple). The social relations being described here are also *simplex* (in Figures 3.2 and 3.3). Figure 3.4 is an example of one method of representing *multiplex* relational data with a single graph.

Let's take a moment to review some of this terminology in a little more detail.

Levels of Measurement: Binary, Signed, and Valued Graphs

In describing the pattern of who describes whom as a close friend, we could have asked our question in several different ways. If we asked each respondent "is this person a close friend or not," we are asking for a binary choice: each person is or is not chosen by each interviewee. Many social relationships can be described this way: the only thing that matters is whether a tie exists or not. When our data are collected this way, we can graph them simply: an arrow represents a choice that was made, no arrow represents the absence of a choice. But, we could have asked the question a second way: "for each person on this list, indicate whether you like, dislike, or don't care." We might assign a + to indicate "liking," zero to indicate "don't care" and - to indicate dislike. This kind of data is called "signed" data. The graph with signed data uses a + on the arrow to indicate a positive choice, a - to indicate a negative choice, and no arrow to indicate neutral or indifferent. Yet another approach would have been to ask: "rank the three people on this list in order of who you like most, next most, and least." This would give us "rank order" or "ordinal" data describing the strength of each friendship choice. Lastly, we could have asked: "on a scale from minus one hundred to plus one hundred - where minus 100 means you hate this person, zero means you feel neutral, and plus 100 means you love this person - how do you feel about...". This would give us information about the value of the strength of each choice on a (supposedly, at least) ratio level of measurement. With either an ordinal or valued graph, we would put the measure of the strength of the relationship on the arrow in the diagram.

Directed or "bonded" ties in the graph

In our example, we asked each member of the group to choose which others in the group they regarded as close friends. Each person (ego) then is being asked about ties or relations that they themselves direct toward others (alters). Each alter does not necessarily feel the same way about each tie as ego does: Bob may regard himself as a good friend to Alice, but Alice does not necessarily regard Bob as a good friend. It is very useful to describe many social structures as being composed of "directed" ties (which can be binary, signed, ordered, or valued). Indeed, most social processes involve sequences of directed actions. For example, suppose that person A directs a comment to B, then B directs a comment back to A, and so on. We may not know the order in which actions occurred (i.e. who started the conversation), or we may not care. In this example, we might just want to know that "A and B are having a conversation." In this case, the tie or relation "in conversation with" necessarily involves both actors A and B. Both A and B are "co-present" or "co-occurring" in the relation of "having a conversation." Or, we might also describe the situation as being one of an the social institution of a "conversation" that by definition involves two (or more) actors "bonded" in an interaction (Berkowitz).

"Directed" graphs use the convention of connecting nodes or actors with arrows that have arrow heads, indicating who is directing the tie toward whom. This is what we used in the graphs above, where individuals (egos) were directing choices toward others (alters). "Simple" or "Co-occurrence" or "co-presence" or "bonded-tie" graphs use the convention of connecting the pair of actors involved in the relation with a simple line segment (no arrow head). Be careful here, though. In a directed graph, Bob could choose Ted, and Ted choose Bob. This would be represented by headed arrows going from Bob to Ted, and from Ted to Bob, or by a double-headed arrow. But, this represents a different meaning from a graph that shows Bob and Ted connected by a single line segment without arrow heads. Such a graph would say "there is a relationship called close friend which ties Bob and Ted together." The distinction can be subtle, but it is important in some analyses.

Simplex or multiplex relations in the graph

Social actors are often connected by more than one kind of relationship. In our simple example, we showed two graphs of simple (sometimes referred to as "simplex" to differentiate from "multiplex") relations. The friendship graph (figure 3.2) showed a single relation (that happened to be binary and directed). The spouse graph (figure 3.3) showed a single relation (that happened to be binary and un-directed). Figure 3.4 combines information from two relations into a "multiplex" graph.

There are, potentially, different kinds of multiplex graphs. We graphed a tie if there was either a friendship or spousal relation. But, we could have graphed a tie only if there were both a friendship and spousal tie (what would such a graph look like?).

We also combined the information about multiple ties into a single line. Alternatively, one might use different symbols, colors, line widths, or other devices to keep all of the information about multiple relations visible in a multiplex graph -- but the result can often be too complicated to be useful.

3.4: Summary

A graph (sometimes called a sociogram) is composed of nodes (or actors or points) connected by edges (or relations or ties). A graph may represent a single type of relations among the actors (simplex), or more than one kind of relation (multiplex). Each tie or relation may be directed (i.e. originates with a source actor and reaches a target actor), or it may be a tie that represents co-occurrence, co-presence, or a bonded-tie between the pair of actors. Directed ties are represented with arrows, bonded-tie relations are represented with line segments. Directed ties may be reciprocated (A chooses B and B chooses A); such ties can be represented with a double-headed arrow. The strength of ties among actors in a graph may be nominal or binary (represents presence or absence of a tie); signed (represents a negative tie, a positive tie, or no tie); ordinal (represents whether the tie is the strongest, next strongest, etc.); or valued (measured on an interval or ratio level). In speaking the position of one actor or node in a graph to other actors or nodes in a graph, we may refer to the focal actor as "ego" and the other actors as "alters."

3.E: Using graphs to represent social relations (Exercises)

Review Questions

1. What are "nodes" and "edges"? In a sociogram, what is used for nodes? for edges?

2. How do valued, binary, and signed graphs correspond to the "nominal" "ordinal" and "interval" levels of measurement?

3. Distinguish between directed relations or ties and "bonded" relations or ties.

4. How does a reciprocated directed relation differ from a "bonded" relation?

5. Give and example of a multi-plex relation. How can multi-plex relations be represented in graphs?

Application Questions

1. Think of the readings from the first part of the course. Did any studies present graphs? If they did, what kinds of graphs were they (that is, what is the technical description of the kind of graph or matrix). Pick one article and show what a graph of its data would look like.

2. Suppose that I was interested in drawing a graph of which large corporations were networked with one another by having the same persons on their boards of directors. Would it make more sense to use "directed" ties, or "bonded" ties for my graph? Can you think of a kind of relation among large corporations that would be better represented with directed ties?

3. Think of some small group of which you are a member (maybe a club, or a set of friends, or people living in the same apartment complex, etc.). What kinds of relations among them might tell us something about the social structures in this population? Try drawing a graph to represent one of the kinds of relations you chose. Can you extend this graph to also describe a second kind of relation? (e.g. one might start with "who likes whom?" and add "who spends a lot of time with whom?").

4. Make graphs of a "star" network, a "line" and a "circle." Think of real world examples of these kinds of structures where the ties are directed and where they are bonded, or undirected. What does a strict hierarchy look like? What does a population that is segregated into two groups look like?

CHAPTER OVERVIEW

4: Working with Netdraw to Visualize Graphs

A lot of the work that we do with social networks is primarily descriptive and/or exploratory, rather than confirmatory hypothesis testing. Using some of the tools described in this chapter can be particularly helpful because they may let you see patterns that you might not otherwise have seen. The tools can be used to explore tentative empirical generalizations and provide crude first examinations of hypotheses about patterns that may be present in the data. Some of the tools are also very helpful for dealing with the complexity of social network data, which may involve many actors, many ties, and several types of ties. Hiding, highlighting, and locating parts of the data can be a big help in making sense of the data. In some cases (like ego networks and the evolution of networks over time) hiding and revealing parts of the data are critical to understanding and describing the construction and evolution of the social structures.

4.1: Introduction - A picture is worth...

As we saw in chapter 3, a graph representing the information about the relations among nodes can be an very efficient way of describing a social structure. A good drawing of a graph can immediately suggest some of the most important features of overall network structure. Are all the nodes connected? Are there many or few ties among the actors? Are there sub-groups or local "clusters" of actors that are tied to one another, but not to other groups? Are there some actors with many ties, and some with few?

A good drawing can also help us to better understand how a particular "ego" (node) is "embedded" (connected to) its "neighborhood" (the actors that are connected to ego, and their connections to one another) and to the larger graph (is "ego" an "isolate" a "pendant"?). By looking at "ego" and the "ego network" (i.e. "neighborhood"), we can get a sense of the structural constraints and opportunities that an actor faces; we may be better able to understand the role that an actor plays in a social structure.

There is no single "right way" to represent network data with graphs. There are a few basic rules, and we reviewed these in the previous chapter. Different ways of drawing pictures of network data can emphasize (or obscure) different features of the social structure. It's usually a good idea to play with visualizing a network, to experiment and be creative. There are a number of software tools that are available for drawing graphs, and each has certain strengths and limitations. In this chapter, we will look at some commonly used techniques for visualizing graphs using NetDraw (version 4.14, which is distributed along with UCINET). There are many other packages though, and you might want to explore some of the tools available in Pajek, and Mage (look for software at the web-site of the International Network of Social Network Analysts - INSNA).

Of course, if there are a large number of actors or a large number of relations among them, pictures may not help the situation much; numerical indexes describing the graph may be the only choice. Numerical approaches and graphical approaches can be used in combination, though. For example, we might first calculate the "between-ness centrality" of the nodes in a large network, and then use graphs that include only those actors that have been identified as "important."

4.2: Node Attributes

Differences of kind: We often have information available about some attributes of each the actors in our network. In the Bob, Carol, Ted and Alice example, we noted that two of the actors were male and two female. The scores of the cases (Bob, Carol, Ted, Alice) on the variable "sex" are a nominal dichotomy. It is also pretty common to be able to divide actors in a "multiple-choice" way; that is, we can record an attribute as a nominal polyotomy (for example, if we knew the religious affiliation of each actor, we might record it as "Christian," "Muslim," "Jewish," "Zoroastrian," or whatever).

It is often the case that the structure of a network depends on the attributes of the actors embedded in it. If we are looking at the network of "spouse" ties among Bob, Carol, Ted, and Alice, one would note that ties exist for male-female pairs, but not (in our example) for female-female or male-male pairs. Being able to visualize the locations of different types of actors in a graph can help us see patterns, and to understand the nature of the social processes that generated the tie structure.

Using colors and shapes are useful ways of conveying information about what "type" of actor each node is. Figures 4.1 and 4.2 provide an example. The data here describe the exchange of information among ten organizations that were involved in the local political economy of social welfare services in a Midwestern city (from a study by David Knoke; the data are one of the data sets distributed with UCINET). In Figure 4.1, NetDraw has been used to render a directed graph of the data. This is done by opening *Netdraw>File>Open>UCInet dataset>Network*, and locating the data file. NetDraw produces a basic graph that you can then edit.

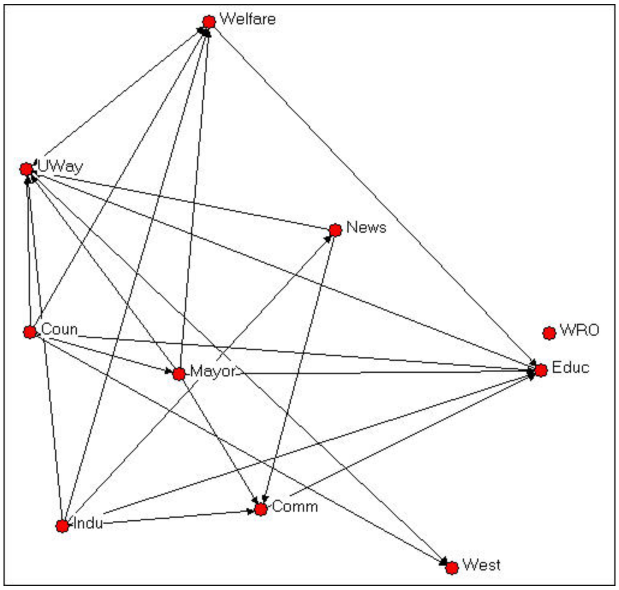


Figure 4.1. *Knoke information exchange network*

Institutional theory might suggest that information exchange among organizations of the same "type" would be more common than information exchange between organizations of different types. Some of the organizations here are governmental (Welfare, Coun, Educ, Mayor, Indu), some are non-governmental (UWay, News, WRO, Comm, West).

In Netdraw, we used the *Transform>mode attribute* editor to assign a score of "1" to each node if it was governmental, and "0" if it was not. We then used *Properties>nodes>color>attribute-based* to select the government attribute, and assign the color red to government organizations, and blue to non-government organizations. You could also create an attribute data file in UCINET using the same nodes as the network data file, and creating one or more columns of attributes. *NetDraw>File>Open>UCInet dataset>Attribute data* can then be used to open the attributes, along with the network, in NetDraw.

Ecological theory of organizations suggests that a division between organizations that are "generalists" (i.e. perform a variety of functions and operate in several different fields) and organizations that are "specialists" (e.g. work only in social welfare) might affect information-sharing patterns.

In Netdraw, we used the *Transform>mode attribute* editor to create a new column to hold information about whether each organization was a "generalist" or a "specialist." We assigned the score of "1" to "generalists" (e.g. the Newspaper, Mayor) and a score of "0" to "specialists (e.g. the Welfare Rights Organization). We then used *Properties>nodes>shape>attribute-based* to assign the shape "square" to generalists and "circle" to specialists. The result of these operations is shown in figure 4.2.

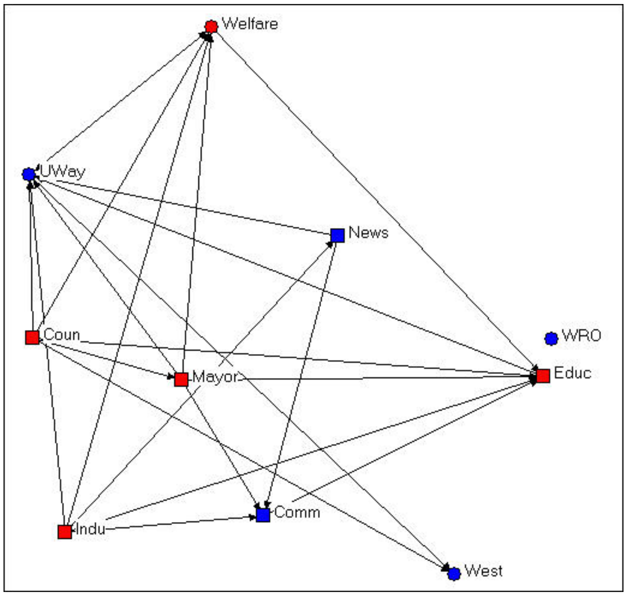


Figure 4.2. *Knoke information exchange with government/non-government and specialist/generalist codes*

A visual inspection of the diagram with the two attributes highlighted by node color and shape is much more informative about the hypotheses of differential rates of connection among red/blue and among circle/square. It doesn't look like this diagram is very supportive of either of our hypotheses.

Identifying types of nodes according to their attributes can be useful to point out characteristics of actors that are based on their position in the graph. Consider the example in the next Figure, 4.3.

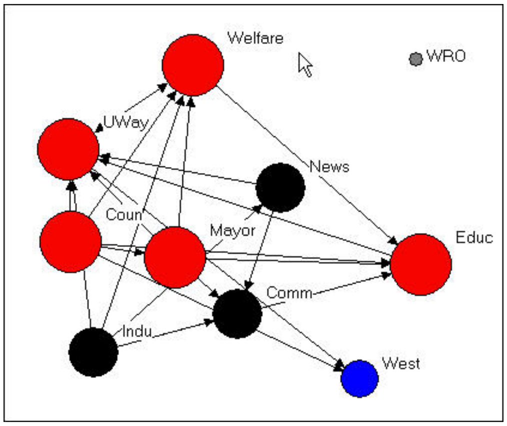


Figure 4.3. *Knoke information exchange with k-cores*

This figure was created by using the *Analysis>K-core* tool that is built into NetDraw. We'll talk about the precise definition of a k-core in a later chapter. But, generally, a k-core is a set of nodes that are more closely connected to one another than they are to nodes in other k-cores. That is, a k-core is one definition of a "group" or "sub-structure" in a graph.

Figure 4.3 shows four sub-groups, which are colored to identify which nodes are members of which group (the "West" group and the "WRO" group each contain only a single node). In addition, the size of the nodes in each K-core are proportional to the size of the K-core. The largest group contains government members (Mayor, County Government, Board of Education), as well as the main public (Welfare) and private (United Way) welfare agencies. A second group, colored in black, groups together the newspaper, chamber of commerce, and industrial development agency. Substantively, this actually makes some sense!

This example shows that color and shape of nodes to represent qualitative differences among actors can be based on classifying actors according to their position in the graph, how they are embedded, rather than on some inherent feature of the actor itself (e.g. governmental or non-governmental). One can use UCINET (or other programs) to identify "types" of actors based on their relations (e.g. where are the cliques?), and then enter this information into the attribute editor of NetDraw (*Transform>node attribute editor>edit>add column*). The groupings that are created by using *Analysis* tools already built-in to NetDraw are automatically added to the node attribute data base).

Differences of amount: Figure 4.3 also uses the size of the nodes (*Properties>nodes>size>attribute-based*) to display an index of the number of nodes in each group. This difference of amount among the nodes is best indicated, visually, by assigning the size of the node to values of some attribute. In example 4.3, NetDraw has done this automatically for an amount that was computed by its *Analysis>K-cores* tool. One can easily compute other variables reflecting differences in amount among actors (e.g. how many "out degrees" or arrows from each actor are there?) using UCINET or other programs. Once these quantities are computed, they can be added to NetDraw (*Transform>node attribute editor>edit>add column*), and then added to the graph (*Properties>nodes>size>attribute-based*).

Differences of amount among the nodes could also reflect an attribute that is inherent to an actor alone. In the welfare organizations example, we might know the annual budget or number of employees of each organization. These measures of organizational "size" might be added to the node attributes, and used to scale the size of the nodes so that we could see whether information sharing patterns differ by organizational size.

Color, shape, and size of the nodes in a graph then can be used to visualize information that we have about the attributes of the actors. These attributes can be based on either "external" information about inherent differences in kind and amount among the actors; or, the attributes can be based on "internal" information about differences in kind and amount that describe how the actor is embedded in the relational network.

4.3: Relation Properties

A graph, as we discussed in the last chapter, is made up of both the actors and the relations among the actors. The relations among the actors (the line segments in a simple graph or the arrows in a directed graph) can also have "attributes." Sometimes it can be very helpful to use color and size to indicate difference of kind and amount among the relations. You can be creative with this idea to explore and display patterns in the connections among the actors in a network. Here are a few ideas of things that you could do.

Suppose that you wanted to highlight certain "types" of relations in the graph. For example, in a sociogram of friendship ties you might want to show how the patterns of ties among persons of the same sex differed from the patterns of ties among persons of different sexes. You might want to show the ties in the graph as differing in color or shape (e.g. dashed or solid line) depending on the type of relation. If you have recorded the two kinds of relations (same sex, different sex) as two relations in a multiplex graph, then you can use NetDraw's*Properties>Lines>Color* from the menu. Then select *Relations*, and choose the color for each of the relations you want to graph (e.g. red for same-sex, blue for different-sex).

Line colors (but not line shapes) can be used to highlight links within actors of the same type or between actors of different types, or both, using NetDraw. First select *Properties>Lines>Node-attribute* from the menus. Then select whether you want to color the ties among actors of the same attribute type ("*within*") or ties among actors of different types ("*between*"), or both. Then use the drop-down menu to select the attribute that you want to graph.

If you have measured each tie with an ordinal or interval level variable (usually reflecting the "strength" of the tie), you can also assign colors to ties based on tie strength (*Properties>Lines>Color>Tie-strength*). But, when you have information on the "value" of the relations, a different method would usually be preferred.

Where the ties among actors have been measured as a value (rather than just present-absent), the magnitude of the tie can be suggested by using thicker lines to represent stronger ties, and thinner lines to represent weaker ties. Recall that sometimes ties are measured as negative-neutral-positive (recorded as -1, 0, +1), as grouped ordinal (5=very strong, 4=strong, 3= moderate, 2=weak, 1=very weak, 0=absent), full-rank order (10=strongest tie of 10, 9=second strongest tie of 10, etc.), or interval (e.g. dollars of trade flowing from Belgium to the United States in 1975).

Since the value of the tie is already recorded in the data set, it is very easy to get NetDraw to visualize it in the graph. From the menus, select *Properties>Lines>Size*. Then, select *Tie-Strength* and indicate which relation you want graphed. You can select the amount of gradation in the line widths (e.g. from 0 to 5 for a grouped ordinal variable with 6 levels; or from 5 to 10 if you want really thick lines).

Using line colors and thickness, you can highlight certain types of ties and varying strength of ties among actors in the network. This can be combined with visual highlighting of attributes of the actors to make a compelling presentation of the features of the graph that you want to emphasize. There is a lot of art to this, and you will have to play and experiment. Node and line attributes can obscure as well as reveal; they can mis-represent, as well as represent. Sometimes, they add to the confusion of an already too-complicated graph.

4.4: Location, location, location

Most graphs of networks are drawn in a two-dimensional "X-Y axis" space (Mage and some other packages allow 3-dimensional rendering and rotation). Where a node or a relation is drawn in the space is essentially arbitrary -- the full information about the network is contained in its list of nodes and relations. The figures below are exactly the same network (Knoke's money flow network) that has been rendered in several different ways.

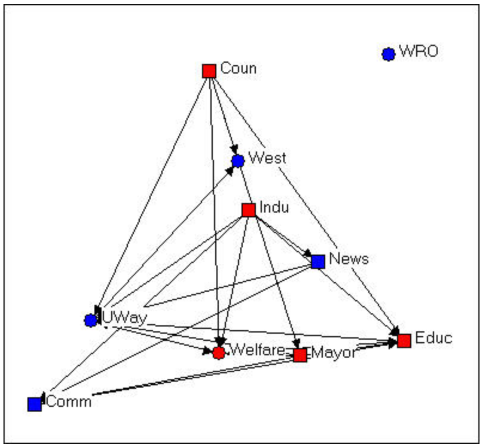


Figure 4.4. *Random configuration of Knoke's money network*

This drawing was created using NetDraw's *Layout>Random* on the graph that we had previously "colored" (blue for non-government, red for government; circles for welfare specialists, squares for generalists). This algorithm locates the nodes randomly in the X-Y space (you can use other tools to change the size of the graphic, rotate it, etc.). Since the X and Y directions don't "mean" anything, the location of the nodes and relations don't provide any particular insight.

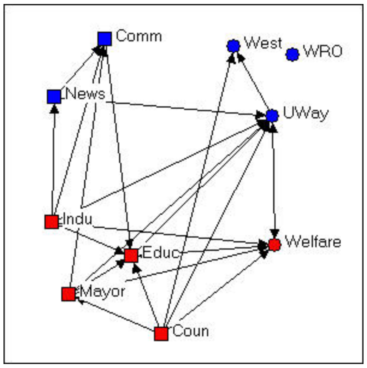


Figure 4.5. *Free-hand grouping by attributes configuration of Knoke's money network*

In Figure 4.5, we've used the "drag and drop" method ("grab" a node with the cursor, and drag it to a new location) to relocate the nodes so that organizations that share the same combinations of attributes are located in different quadrants of the graph. Since we had hypothesized that organizations of like "kind" would have higher rates of connection, this is a useful (but still arbitrary) way to locate the points. If the hypothesis were strongly supported (and its not) most of the arrows would be located within each the four quadrants, and there would be few arrows between quadrants. NetDraw has a built-in tool that allows the user to assign the X and Y dimensions of the graph to scores on attributes (either categorical or continuous): *Layout>Attributes as Coordinates*, and then select attributes to be assigned to X or Y or both. This can be a very useful tool for exploring how patterns of ties differ within and between "partitions" (or types of nodes).

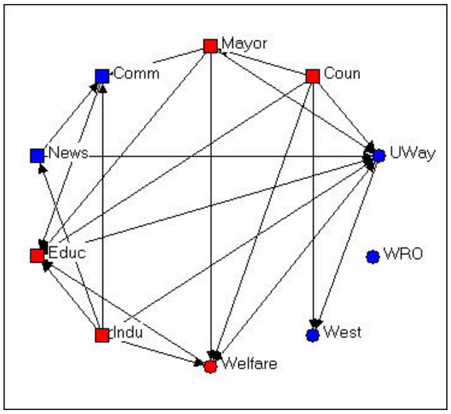


Figure 4.6. *Circle configuration of Knoke's money network*

Figure 4.6 shows the same graph using *Layout>Circle*, and selecting the "generalist-specialist" (i.e. the circle or square node type) as the organizing criterion. Circle graphs are commonly used to visualize which nodes are most highly connected. The nodes are located at equal distances around a circle, and nodes that are highly connected are very easy to quickly locate (e.g. UWay, Educ) because of the density of lines. When nodes sharing the same attribute are located together in a segment of the circle, the density of ties within and between types is also quite apparent.

In each of the different layouts we've discussed so far, the distances between the nodes are arbitrary, and can't be interpreted in any meaningful way as "closeness" of the actors. And, the "directions" X and Y have no meaning -- we could rotate any of the graphs any amount, and it would not change a thing about our interpretation. There are several other commonly used graphic layouts that do try to make the distances and/or directions of locations among the actors somewhat more meaningful.

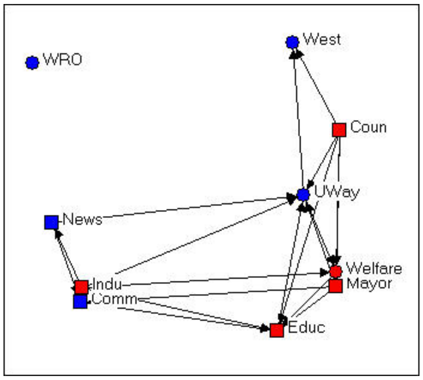


Figure 4.7. *Non-metric multi-dimensional scaling configuration of Knoke's money network*

Figure 4.7 was generated using the *Layout>Graph Theoretic Layout>MDS* tool of NetDraw. MDS stands for (non-metric, in this case) "Multi-Dimensional Scaling." MDS is a family of techniques that is used (in network analysis) to assign locations to nodes in multi-dimensional space (in the case of the drawing, a 2-dimensional space) such that nodes that are "more similar" are closer together. There are many reasonable definitions of what it means for two nodes to be "similar." In this example, two nodes are "similar" to the extent that they have similar shortest paths (geodesic distances) to all other nodes. There are many, many ways of doing MDS, but the default tools chosen in NetDraw can often generate meaningful renderings of graphs that provide insights. NetDraw has several built-in algorithms for generating coordinates based on similarity (metric and non-metric two-dimensional scaling, and principle components analysis).

One very important difference between figure 4.7 and the earlier graphs is that the distances between the nodes is interpretable. The "Welfare" and "Mayor" nodes are very similar in their geodesic distances from other actors. "West" and "Educ" have very different patterns of ties to the other nodes.

The other important difference between figure 4.7 and the earlier graphs is that direction may be meaningful. Notice that there is a cluster of nodes at the left (News, Indu, Comm) that are all pretty much not welfare organizations themselves, while the nodes at the right are (generally) more directly involved in welfare service provision. The upper left-hand quadrant contains mostly "blue" nodes, while the lower right quadrant contains mostly "red" ones -- so one "direction" might be interpreted as "non-government/government."

Let me emphasize that different applications of MDS (and other scaling tools) to different definitions of what it means for nodes to be "similar" can generate wildly different looking graphs. These are (almost always) exploratory techniques, and there is (usually) no single "correct" interpretation. But, a graphic that uses both distance and direction to summarize something about the structure of the network can provide considerable insight when compared to graphs (like figures 4.4, 4.5, and 4.6) that don't.

Consider one last example of rendering the same data, this time using NetDraw's unique built-in algorithm for locating points (*Layout>Graph Theoretic Layout>Spring Embedding*).

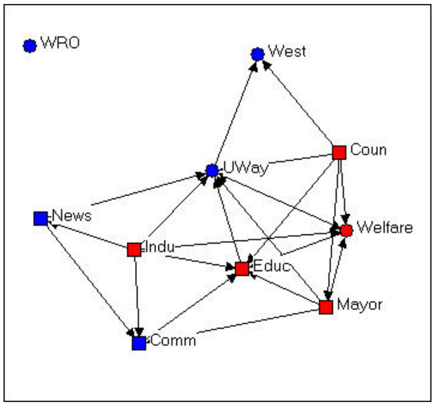


Figure 4.8. *"Spring-embedding" configuration of Knoke's money network*

You might immediately notice that this graph is fairly similar to the MDS solution. The algorithm uses iterative fitting (i.e. start with a random graph, measure "badness" of fit; move something, measure "badness" and if it's better keep going in that direction...) to locate the points in such a way as to put those with smallest path lengths to one another closest in the graph. This approach can often locate points very close together, and make for a graph that is hard to read. In the current example, we've also selected the optional "node repulsion" criterion that creates separation between objects that would otherwise be located very close to one another. We've also used the optional criterion of seeking to make the paths of "equal edge length" so that the distances between adjacent objects are similar.

The result is a graph that preserves many of the features of the dimensional scaling approach (distances are still somewhat interpretable; directions are often interpretable), but is usually easier to read -- particularly if it matters which specific nodes are where (rather than node types of clusters).

There is no one "right way" to use space in a graph. But one can usually do much better than a random configuration -- particularly if one has some prior hypotheses or research questions about the kinds of patterns that would be meaningful.

4.5: Highlighting parts of the network

Large networks (those that contain many actors, many kinds of relations, and/or high densities of ties) can be very difficult to visualize in any useful way -- there is simply too much information. Often, we need to clear away some of the "clutter" to see main patterns more clearly.

One of the most interesting features of social networks -- whether small or large -- is the extent to which we can locate "local sub-structures." We will discuss this topic a good bit more in a later chapter. Highlighting or focusing attention on sub-sets of nodes in a drawing can be a powerful tool for visualizing sub-structures.

In this section, we will briefly outline some approaches to rendering network drawings that can help to simplify complex diagrams and locate interesting sub-graphs (i.e. collections of nodes and their connections).

Clearing away the underbrush

Social structures can be composed of multiple relations. Bob, Carol, Ted, and Alice in our earlier example are a multi-plex structure of people connected by both friendship and by spousal ties. Graphs that display all of the connections among a set of nodes can be very useful for understanding how actors are tied together -- but they can also get so complicated and dense that it is difficult to see any patterns. There are a couple approaches that can help.

One approach is to combine multiple relations into an index. For example, one could combine the information on friendship and spousal ties using an "and" rule: if two nodes have both a friendship and spousal tie, then they have a tie - otherwise they do not (i.e. if they have no tie, or only one type of tie). Alternatively, we could create an index that records a tie when there is either a friendship tie or a spousal tie. If we had measured relations with values, rather than simple presence-absence, multiple relations could be combined by addition, subtraction, multiplication, division, averaging, or other methods. UCINET has tools for these kinds of operations, that are located at: *Transform>matrix operations>within dataset>aggregations*.

The other approach is to simplify the data a bit. NetDraw has some tools that can be of some help.

Rather than examining the information on multiple kinds of ties in one diagram, one can look at them one at a time, or in combination. If the data have been stored as a UCINET or NetDraw data file with multiple relations, then the *Options>View>Relations Box* opens a dialog box that lets you select which relations you want to display. Suppose that we had a data set in which we had recorded the friendship ties among a number of people at intervals over a period of time. By first displaying the first time point, and then adding subsequent time point, we can visualize the evolution of the friendship structure.

It isn't unusual for some of the nodes in a graph of a social network to not be connected to the others at all. Nodes that aren't connected are called "isolates." Some nodes may be connected to the network by a single tie. These nodes sort of "dangle" from the diagram; they are called "pendants." One way of simplifying graphs is to hide isolates and/or pendants to reduce visual clutter. Of course, this does mis-represent the structure, but it may allow us to focus more attention where most of the action is. NetDraw has both button-bar tools and a menu item (*Analysis>Isolates*) to hide these less-connected nodes.

Finding and visualizing local sub-structures

One of the common questions in network analysis is whether a graph displays various kinds of "sub-structures." For example, a "clique" is a sub-structure that is defined as a set of nodes where every element of the set is connected to every other member. A network that has no cliques might be a very different place than a network that has many small cliques, or one that has one clique and many nodes that are not part of the clique. We'll take a closer look at UCINET tools for identifying sub-structures in a later chapter.

NetDraw has built-in a number of tools for identifying sub-structures, and automatically coloring the graph to identify them visually.

Analysis>components locates the parts of graph that are completely disconnected from one another, and colors each set of nodes (i.e. each component). In our Bob-Carol-Ted-Alice example, the entire graph is one component, because all the actors are connected. In the welfare bureaucracies example, there are two components, one composed of only WRO (which does not receive ties from any other organization) and the other composed of the other nine nodes. In NetDraw, executing this command also creates a variable in the database of node attributes -- as do all the other commands discussed here. These attributes can then be used for selecting cases, changing color, shape, and size, etc.

Analysis>Blocks and Cutpoints locates parts of the graph that would become disconnected components if either one node or one relation were removed (the blocks are the resulting components; the cutpoint is the node that would, if removed, create the dis-connect). NetDraw graphs these sub-structures, and saves the information in the node-attribute database.

Analysis>K-cores locates parts of the graph that form sub-groups such that each member of a sub-group is connected to N-K of the other members. That is, groups are the largest structures in which all members are connected to all but some number (K) of other members. A "clique" is a group like this where all members are connected to all other members; "fuzzier" or "looser" groups are created by increasing "K." NetDraw identifies the K-cores that are created by different levels of K, and provides colored graphs and data-base entries.

Analysis>Subgroups>block based. Sorry, but I don't know what this algorithm does! Most likely, it creates sub-structures that would become components with differing amounts of nodes/relations removed.

Analysis>Subgroups>Hierarchical Clustering of Geodesic Distances. The geodesic distance between two nodes is the length of the shortest path between them. A hierarchical clustering of distances produces a tree-like diagram in which the two nodes that are most similar in their profile of distances to all other points are joined into a cluster; the process is then repeated over and over until all nodes are joined. The resulting graphic is one way of understanding which nodes are most similar to one another, and how the nodes may be classified into "types" based on their patterns of connection to other nodes. The graph is colored to represent the clusters, and database information is stored about the cluster memberships at various levels of aggregation. A hierarchical clustering can be very interesting in understanding which groups are more homogeneous (those that group together at early stages in the clustering) than others; moving up the clustering tree diagram, we can see a sort of a "contour map" of the similarity of nodes.

Analysis>Subgroups>Factions (select number). A "faction" is a part of a graph in which the nodes are more tightly connected to one another than they are to members of other "factions." This is quite an intuitively appealing idea of local clustering or sub-structure (though, as you can see, only one such idea). NetDraw asks you how many factions you would like to find (always explore various reasonable possibilities!). The algorithm then forms the number of groups that you desire by seeking to maximize connection within, and minimize connection between the groups. Points are colored, and the information about which nodes fall in which partitions (i.e. which cases are in which factions) is saved to the node attributes database.

Analysis>Subgroups>Newman-Girvan. This is another numerical algorithm that seeks to create clusters of nodes that are closely connected within, and less connected between clusters. The approach is that of "block modeling." Rows and columns are moved to try to create "blocks" where all connections within a block are present, and all connections between blocks are absent. This algorithm will usually produce results similar to the factions algorithm. Importantly, though, the Newman-Girvan algorithm also produces measures of goodness-of-fit of the configuration for two blocks, three blocks, etc. This allows you to get some sense of what division into blocks is optimal for your needs (there isn't one "right" answer).

Ego Networks (neighborhoods)

A very useful way of understanding complicated network graphs is to see how they arise from the local connections of individual actors. The network formed by selecting a node, including all actors that are connected to that node, and all the connections among those other actors is called the "ego network" or (1-step) neighborhood of an actor. Figure 4.9 is an example from the Knoke bureaucracies information network, where we select as our "ego" the board of education.

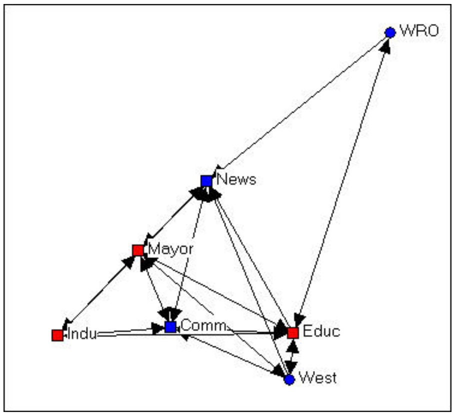


Figure 4.9. *Ego network of Educ in Knoke information network*

We note that the ego-network of the board of education is fairly extensive, and that the density of connection among the actors in the ego-network is fairly high. This is to say say the the board of education is quite "embedded" in a dense local sub-structure.

Next, let's add the ego network of the "West" agency, in figure 4.10.

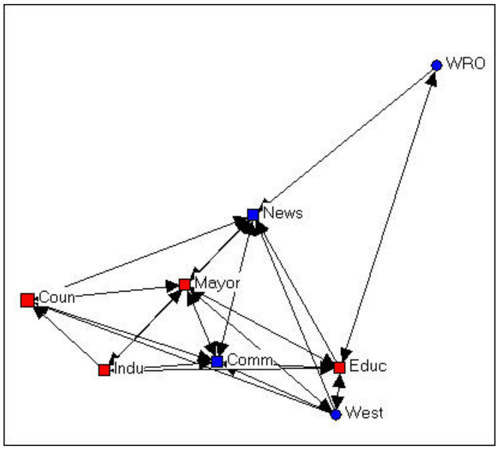


Figure 4.10. *Ego networks of Educ and West in Knoke information network*

The two ego networks combined result in a fully connected structure. We note that one connection between Educ and Coun is mediated by West.

One can often gain considerable insight about complicated networks by "building" them starting with one actor and adding others. Or, one can begin with the whole network, and see what happens as individual's ego networks are removed.

The network of each individual actor may also be of considerable interest. Who's most connected? How dense are the neighborhoods of particular actors?

NetDraw has useful tools for visualizing and working with ego-networks. The *Layout>Egonet* command presents a dialog box that lets you select which ego's networks are to be displayed. You can start with all the actors and delete; or start with focal actors and build up the full network.

4.6: A few hints on data handling with NetDraw

Input

There are several ways to get data into NetDraw. Probably the simplest is to import data from UCINET or Pajek. The *File>Open* command lets you read a UCINET text (DL) file (discussed elsewhere), and existing UCINET dataset, or a Pajek dataset. This menu also is used to access data that have been stored in the native data format of the NetDraw program (.VNA format). Once the data has been imported with the *Open* command, the node and line attribute editors of NetDraw can be used to create a diagram that can be saved with colors, shapes, locations, etc.

A second method is to build a dataset within NetDraw itself. Begin by creating a random network (*File>Random*). This creates an arbitrary network of 20 nodes. You can then use the node attributes editor (*Transform>Node Attribute editor*) and the link editor (*Transform>Link Editor*) to modify the nodes (and add or delete nodes) and their attributes; and to create connections among nodes. This is great for small, simple networks; for more complicated data, it's best to create the basic data set elsewhere and import it.

The third method is to use an external editor to create a NetDraw dataset (a .vna file) directly. This file is a plain ascii text file (if you use a word processor, be sure to save as ascii text). The contents of the file is pretty simple, and is discussed in the brief tutorial to NetDraw. Here is part of the file for the Knoke data, after we have created some of the diagrams we've seen.

```
*Node data
"ID", "General", "Size", "Govt"
"1" "1" "3" "1"
"2" "1" "2" "0"
.
.
"9" "0" "2" "1"
"10" "0" "1" "0"
*Node properties
ID x y color shape size shortlabel labelsize labelcolor active
"1" 51 476 255 2 16 "Coun" 11 0 TRUE
"2" 451 648 16711680 2 11 "Comm" 11 0 TRUE
.
.
"9" 348 54 255 1 11 "Welfare" 11 0 TRUE
"10" 744 801 16711680 1 6 "West" 11 0 TRUE
*Tie data
FROM TO "KNOKI" "KNOKM"
"1" "2" 1 0
"1" "5" 1 1
```

```
.
.
"9" "3" 0 1
"9" "8" 0 1
*Tie properties
FROM TO color size active
"1" "2" 0 1 TRUE
"1" "5" 0 1 TRUE
.
.
"9" "3" 0 1 FALSE
"9" "8" 0 1 FALSE
```

There are four sections of code here (not all are needed, but the *node data and *tie data are, to define the network structure). *Node data lists variables describing the nodes. An ID variable is necessary, the other variables in the example describe attributes of each node. The (optional) *Node properties section lists the variables, and gives values for ID, location on the diagram (X and Y coordinates from the upper left corner), shape, size, color, etc. Usually, one will not create this code; rather you input the data, use NetDraw to create a diagram, and save the result as a file -- and this section (and the *Tie properties) is created for you.

The *Tie data section is necessary to define the connections among the nodes. There is one data line for each relation in the graph. Each data line is described by its origin and destination, and value. Here. since there are two relations, "KNOKI" and "KNOKM" there are two values -- each of which happens, in our example, to be binary (but they could be valued).

The *Tie properties section is probably best created by using NetDraw and saving the resulting file. Each tie is identified by origin and destination, and its color and size are set. Here, certain ties are not to be visible in the drawing (the "active" property is set to "FALSE").

Output

When you are working with NetDraw, it is a good idea to save a copy of your work in the format (.vna, above) that is native to the program (*File>Save Data As>Vna*). This format keeps all of the information about your diagram (what's visible and not, node and line attributes, locations) so that you can re-open the diagram looking exactly as you left it.

You may also want to save datasets created with NetDraw to other program's formats. You won't be able to save all of the information about node and line properties and locations, but you can save the basic network (what are the nodes, which is connected to which) and node attributes. *File>Save Data As>Pajek* lets you save the network, partitions of it (which record attributes), and clusterings in Pajek format. *File>Save Data As>UCINET* lets you save the basic network information for binary or valued networks (UCINET needs to know which) and attributes (which are stored in a separate file in UCINET).

The whole point of making more interesting drawings of graphs, of course, is to be able to use them to illustrate your ideas. There are several possibilities.

Screen capture programs (I used SnagIT) can take pictures of your graphics that can be then saved in any number of formats, and edited further by external graphics editors (perhaps to add titles, annotations, and other highlights).

File>print, of course, does just that.

File>Save Diagram As let's you save your diagram in three of the most common graphics formats (Windows Metafile, bitmap BMP, or JPEG). Once saved, these file formats can be further edited with graphics editing programs to be inserted into web or hard-copy documents.

CHAPTER OVERVIEW

5: Using Matrices to Represent Social Relations

Graphs are very useful ways of presenting information about social networks. However, when there are many actors and/or many kinds of relations, they can become so visually complicated that it is very difficult to see patterns. It is also possible to represent information about social networks in the form of matrices. Representing the information in this way also allows the application of mathematical and computer tools to summarize and find patterns. Social network analysts use matrices in a number of different ways. So, understanding a few basic things about matrices from mathematics is necessary. We'll go over just a few basics here that cover most of what you need to know to understand what social network analysts are doing. For those who want to know more, there are a number of good introductory books on matrix algebra for social scientists.

5.1: What is a matrix?

To start with, a matrix is nothing more than a rectangular arrangement of a set of elements (actually, it's a bit more complicated than that, but we will return to matrices of more than two dimensions in a little bit). Rectangles have sizes that are described by the number of rows of elements and columns of elements that they contain. A "3 by 6" matrix has three rows and six columns; an "I by j" matrix has I rows and j columns. A matrix that has only one row is called a "row vector." A matrix that has only one column is called a "column vector.

Figure 5.1 shows a two-by-four matrix. Figure 5.2 shows a four by two matrix. Just for the moment, ignore the contents of the cells (e.g. 1,1).

Figure 5.1. Example of a "two-by-four" matrix

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4

Figure 5.2. Example of at "four-by-two" matrix

1,1	1,2
2,1	2,2
3,1	3,2
4,1	4,2

The elements (cells) of a matrix are identified by their "addresses." Element 1,1 is the entry in the first row and first column; element 13,2 is in the 13th row and is the second element of that row. The cell addresses have been entered as matrix elements in the two examples above.

Matrices are often represented as arrays of elements surrounded by vertical lines at their left and right, or square brackets at the left and right. In web pages it's easier to use "tables" to represent matrices. Matrices can be given names; these names are usually presented as capital bold-faced letters. Social scientists using matrices to represent social networks often dispense with the mathematical conventions, and simply show their data as an array of labeled rows and columns. The labels are not really part of the matrix, but are simply for clarity of presentation. The matrix in figure 5.3 for example, is a 4 by 4 matrix, with additional labels.

Figure 5.3. Four-by-four matrix with additional row and column labels

	A	B	C	D
A	---	1	0	0
B	1	---	1	0
C	1	1	---	1
D	0	0	1	---

The matrices used in social network analysis are frequently "square." That is, they contain the same number of rows and columns. But "rectangular" matrices are also used, as are row and column vectors. The same conventions apply to all these variations.

Occasionally, social network analysts will use a "3-dimensional" matrix. A three dimensional matrix has rows, columns, and "levels" or "slices." Each "slice" has the same rows and columns as each other slice. UCINET thinks about these more complicated 3-dimensional arrays of data as a collection of two-dimensional matrices.

5.2: The "adjacency" matrix

The most common form of matrix in social network analysis is a very simple square matrix with as many rows and columns as there are actors in our data set. The "elements" or scores in the cells of the matrix record information about the ties between each pair of actors.

The simplest and most common matrix is binary. That is, if a tie is present, a one is entered in a cell; if there is no tie, a zero is entered. This kind of a matrix is the starting point for almost all network analysis, and is called an "adjacency matrix" because it represents who is next to, or adjacent to whom in the "social space" mapped by the relations that we have measured.

An adjacency matrix may be "symmetric" or "asymmetric." Social distance can be either symmetric or asymmetric. If Bob and Carol are "friends" they share a "bonded tie" and the entry in the $X_{i,j}$ cell will be the same as the entry in the $X_{j,i}$ cell.

But social distance can be a funny (non-Euclidean) thing. Bob may feel close to Carol, but Carol may not feel the same way about Bob. In this case, the element showing Bob's relationship to Carol would be scored "1," while the element showing Carol's relation to Bob would be scored "0." That is, in an "asymmetric" matrix, $X_{i,j}$ is not necessarily equal to $X_{j,i}$.

By convention, in a directed (i.e. asymmetric) matrix, the sender of a tie is the row and the target of the tie is the column. Let's look at a simple example. The directed graph of friendship choices among Bob, Carol, Ted, and Alice is shown in figure 5.4.

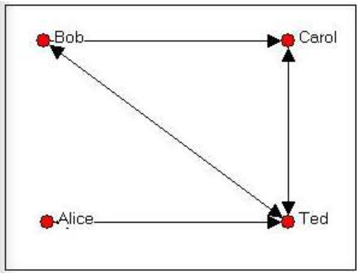


Figure 5.4. *Bob, Carol, Ted, and Alice*

We can since the ties are measured at the nominal level (that is, the data are binary choice data), we can represent the same information in a matrix that looks like:

	Bob	Carol	Ted	Alice
Bob	---	1	1	0
Carol	0	---	1	0
Ted	1	1	---	1
Alice	0	0	1	---

Figure 5.5. *Asymmetric adjacency matrix of the graph shown in Figure 5.4.*

Remember that the rows represent the source of directed ties, and the columns the targets; Bob chooses Carol here, but Carol does not choose Bob. This is an example of an "asymmetric" matrix that represents directed ties (ties that go from a source to a receiver). That is, the element i,j does not necessarily equal the element j,i. If the ties that we were representing in our matrix were "bonded-ties" (for example, ties representing the relation "is a business partner of" or "co-occurrence or co-presence," (e.g. where ties represent a relation like: "serves on the same board of directors as") the matrix would necessarily be symmetric; that is element i,j would be equal to element j,i.

Binary choice data are usually represented with zeros and ones, indicating the presence or absence of each logically possible relationship between pairs of actors.

Signed graphs are represented in matrix form (usually) with -1, 0, and +1 to indicate negative relations, no or neutral relations, and positive relations. "Signed" graphs are actually a specialized version of an ordinal relation.

When ties are measured at the ordinal or interval level, the numeric magnitude of the measured tie is entered as the element of the matrix. As we discussed earlier, other forms of data are possible (multi-category nominal, ordinal with more than three ranks, full-rank order nominal). These other forms, however, are rarely used in sociological studies, and we won't give them very much attention.

In representing social network data as matrices, the question always arises: what do I do with the elements of the matrix where i = j? That is, for example, does Bob regard himself as a close friend of Bob? This part of the matrix is called the *main diagonal*. Sometimes the value of the main diagonal is meaningless, and it is ignored (and left blank or filled with zeros or ones). Sometimes, however, the main diagonal can be very important, and can take on meaningful values. This is particularly true when the rows and columns of our matrix are "super-nodes" or "blocks." More on that in a minute.

It is often convenient to refer to certain parts of a matrix using shorthand terminology. If I take all of the elements of a row (e.g. who Bob chose as friends: ---,1,1,0) I am examining the "*row vector*" for Bob. If I look only at who chose Bob as a friend (the first column, or ---,0,1,0), I am examining the "*column vector*" for Bob. It is sometimes useful to perform certain operations on row or column vectors. For example, if I summed the elements of the column vectors in this example, I would be measuring how "popular" each node was (in terms of how often they were the target of a directed friendship tie). So a "vector" can be an entire matrix (1 x ... or ...x 1), or a part of a larger matrix.

5.3: Matrix permutation, blocks, and images

It is also helpful, sometimes, to rearrange the rows and columns of a matrix so that we can see patterns more clearly. Shifting rows and columns (if you want to rearrange the rows, you must rearrange the columns in the same way, or the matrix won't make sense for most operations) is called "permutation" of the matrix.

Our original data look like Figure 5.6:

	Bob	Carol	Ted	Alice
Bob	---	1	1	0
Carol	0	---	1	0
Ted	1	1	---	1
Alice	0	0	1	---

Figure 5.6. *Asymmetric adjacency matrix*

Let's rearrange (permute) this so that the two males and the two females are adjacent in the matrix. *Matrix permutation* (*Data>Permute*) simply means to change the order of the rows and columns. Since the matrix is symmetric, if I change the position of a row, I must also change the position of the corresponding column. The result is shown in Figure 5.7.

	Bob	Ted	Carol	Alice
Bob	---	1	1	0
Ted	1	---	1	1
Carol	0	1	---	0
Alice	0	1	0	---

Figure 5.7. *Permuted matrix*

None of the elements have had their values changed by this operation or rearranging the rows and columns, we have just shifted things around. We've also highlighted some sections of the matrix. Each colored section is referred to as a *block*. Blocks are formed by passing dividing lines through the matrix (e.g. between Ted and Carol) rows and columns. Passing these dividing lines through the matrix is called *partitioning the matrix*. Here we have partitioned by the actor by their sex. Partitioning is also sometimes called "blocking the matrix," because partitioning produces blocks.

This kind of grouping of cells is often done in network analysis to understand how some sets of actors are "embedded" in social roles or in larger entities. Here, for example, we can see that all occupants of the social role "male" choose each other as friends; no females choose each other as friends, and that males are more likely to choose females (3 out of 4 possibilities are selected) than females are to choose males (only 2 out of 4 possible choices). We have grouped the males together to create a "partition" or "super-node" or "social role" or "block." We often partition social network matrices in this way to identify and test ideas about how actors are "embedded" in social roles or other "contexts."

We might wish to dispense with the individual nodes altogether, and examine only the positions or roles. If we calculate the proportion of all ties within a block that are present, we can create a *block density matrix*. In doing this, we have ignored self-ties in Figure 5.8.

	Male	Female
Male	1.00	0.75
Female	0.50	0.00

Figure 5.8. *Block density matrix*

We may wish to summarize the information still further by using *block image* or *image matrix.* If the density in a block is greater than some amount (we often use the average density for the whole matrix as a cut-off score, in the current example the density is .58), we enter a "1" in a cell of the blocked matrix, and a "0" otherwise. This kind of simplification is called the "image" of the blocked matrix, as in Figure 5.9.

	Male	Female
Male	1	1
Female	0	0

Figure 5.9. *Image matrix of sex blocked data, using overall mean density as the cut-off*

Images of blocked matrices are powerful tools for simplifying the presentation of complex patterns of data. Like any simplifying procedure, good judgment must be used in deciding how to block and what cut-offs to use to create images -- or we may lose important information.

UCINET includes tools that make permuting and blocking matrices rather easy.

Transform>Block allows you to select a matrix to be blocked, a row and/or column partition, and a method for calculating the entries in the resulting blocks.

To use this command, you need to first create separate files that describe the row partition and the column partition. These files are simply vectors (either one row, or one column) that identify which actors are to fall into which partition. For example, if actors 1, 2, and 5 were to form group A, and actors 3 and 4 were to form group B, my column partition data set would read: 1 1 2 2 1. These partitions or blockings are simply regular UCINET data files with one row or one column.

The command asks for a method of summarizing the information within each block. You may take the average of the values in the block (if the data are binary, taking the average is the same thing as calculating the density), sum the values in the block, select the highest value or the lowest value, or select a measure of the amount of variation among the scores in the block -- either the sums of squares or the standard deviation.

The command outputs two new matrices. The "PreImage" data set contains the original scores, but permuted; the "Reduced image dataset" contains a new block matrix containing the block densities.

Transform>Collapse allows you to combine rows and/or columns by specifying (detailed instructions are given on the command window) which elements are to be combined, and how. We might select, for example, to combine columns 1, 2, and 5, and rows 1, 2, and 5 by taking the average of the values (we could also select the maximum, minimum, or sum). The command creates a new matrix that has collapsed the desired rows or columns using the summary operation you selected.

The data menu also gives you some tools for this kind of work:

Data>Permute allows you to re-arrange the rows and/or columns and/or matrices (if your data set contains multiple matrices representing multiple relations, like the Knoke bureaucracies "information" and "money" relations). You simply specify the new order with a list. If I wanted to group rows 1, 2, and 5 to be new rows 1, 2, and 3; and rows 3 and 4 to be new rows 4 and 5, I would enter 1 2 4 5 3.

Data>Sort re-arranges the rows, columns, or both of the matrix according to a criterion you select. If you data are valued (i.e. represent tie strength) you might want to sort the rows and columns in ascending or descending order (this could make sense for binary data, too). If you want a more complicated sort (say "all the 3's first, then all the 1's, then all the 2's) you can use an external UCINET data file to specify this as a vector (i.e. the data set would just be: 3 1 2).

Data>Transpose re-arranges the data in a way that is very commonly used in matrix algebra -- by taking the "transpose." A transpose is, very simply, switching the rows and columns of a matrix for one another.

5.4: Doing mathematical operations on matrices

Representing the ties among actors as matrices can help us to see patterns by performing simple manipulations like summing row vectors or partitioning the matrix into blocks. Social network analysts use a number of other mathematical operations that can be performed on matrices for a variety of purposes (matrix addition and subtraction, transposes, inverses, matrix multiplication, and some other more exotic stuff like determinants and eigenvalues). Without trying to teach you matrix algebra, it is useful to know at least a little bit about some of these mathematical operations, and what they are used for in social network analysis.

UCINET has built-in functions for doing most matrix algebra functions. Look under the *Tools>Matrix Algebra* menu. If you do know some matrix algebra, you will find that this tool lets you do almost anything to matrix data that you may desire. But, you do need to know what you are doing. The help screen for this command shows how to identify the matrix or matrices that are to be manipulated, and the algorithms that can be applied.

Transposing a matrix

This simply means to exchange the rows and columns so that i becomes j, and *vice versa*. If we take the transpose of a directed adjacency matrix and examine its row vectors (you should know all this jargon by now!), we are looking at the sources of ties directed at an actor. The degree of similarity between an adjacency matrix and the transpose of that matrix is one way of summarizing the degree of symmetry in the pattern of relations among actors. That is, the correlation between an adjacency matrix and the transpose of that matrix is a measure of the degree of reciprocity of ties (think about that assertion a bit). Reciprocity of ties can be a very important property of a social structure because it relates to both the balance and to the degree and form of hierarchy in a network. This command is also available as *Data>Transpose*.

Taking the inverse of a matrix

This is a mathematical operation that finds a matrix which, when multiplied by the original matrix, yields a new matrix with ones in the main diagonal and zeros elsewhere (which is called an identity matrix). Without going any further into this, you can think of the inverse of a matrix as being sort of the "opposite of" the original matrix. Matrix inverses are used mostly in calculating other things in social network analysis. They are sometimes interesting to study in themselves, however. It is sort of like looking at black lettering on white paper versus white lettering on black paper: sometimes you see different things. Inverses are calculated with *Tools>Matrix Algebra*.

Matrix addition and matrix subtraction

These are the easiest of matrix mathematical operations. One simply adds together or subtracts each corresponding i,j element of the two (or more) matrices. Of course, the matrices that this is being done to have to have the same numbers of I and j elements (this is called "conformable" to addition and subtraction) - and, the values of i and j have to be in the same order in each matrix.

Matrix addition and subtraction are most often used in network analysis when we are trying to simplify or reduce the complexity of multiplex (multiple relations recorded as separate matrices or slices) data to simpler forms. If I had a symmetric matrix that represented the tie "exchanges money" and another that represented the relation "exchanges goods" I could add the two matrices to indicate the intensity of the exchange relationship. Pairs with a score of zero would have no relationship, those with a "1" would be involved in either barter or commodity exchange, and those with a "2" would have both barter and commodity exchange relations. If I subtracted the "goods" exchange matrix from the "money exchange" matrix, a score of -1 would indicate pairs with a barter relationship; a score of zero would indicate either no relationship or a barter and commodity tie; a score of +1 would indicate pairs with only a commodified exchange relationship. For different research questions, either or both approaches might be useful. *Tools>Matrix Algebra* are one way of doing these sorts of data transformations.

Matrix multiplication and Boolean matrix multiplication

Matrix multiplication is a somewhat unusual operation, but can be very useful for the network analyst. You will have to be a bit patient here. First we need to show you how to do matrix multiplication and a few important results (like what happens when you multiply an adjacency matrix times itself, or raise it to a power). Then, we will try to explain why this is useful.

To multiply two matrices, they must be "conformable" to multiplication. This means that the number of rows in the first matrix must equal the number of columns in the second. Usually network analysis uses adjacency matrices, which are square, and hence, conformable for multiplication. Multiplying a matrix by itself (i.e. raising it to a power) and multiplying a square matrix by its transpose are obviously "conformable." Unlike regular multiplication of individual numbers X*Y is not the same thing as Y*X in matrix multiplication -- the order matters!

To multiply two matrices, begin in the upper left hand corner of the first matrix, and multiply every cell in the first row of the first matrix by the values in each cell of the first column of the second matrix, and sum the results. Proceed through each cell in each row in the first matrix, multiplying by the column in the second. To perform a Boolean matrix multiplication, proceed in the same fashion, but enter a zero in the cell if the multiplication product is zero, and one if it is not zero. An example helps. Suppose we wanted to multiply the two matrices in figure 5.10.

0	1
2	3
4	5

times

6	7	8
9	10	11

Figure 5.10. *Two matrices to be multiplied.*

The result is shown in Figure 5.11.

(0*6)+(1*9)	(0*7)+(1*10)	(0*8)+(1*11)
(2*6)+(3*9)	(2*7)+(3*10)	(2*8)+(3*11)
(4*6)+(5*9)	(4*7)+(5*10)	(4*8)+(5*11)

Figure 5.11. *Result of matrix multiplication.*

The mathematical operation in itself doesn't interest us here (any number of programs can perform matrix multiplication). But, the operation is useful when applied to an adjacency matrix. Consider our four friends again, in figure 5.12.

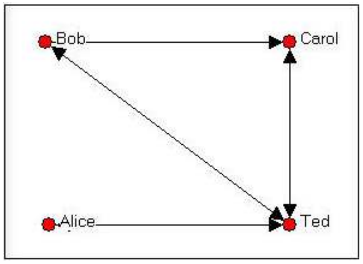


Figure 5.12. *Directed graph of friendship relations among Bob, Carol, Ted, and Alice*

The adjacency matrix for the four actors B, C, T, and A (in that order) is shown as Figure 5.13.

---	1	1	0
0	---	1	0
1	1	---	1
0	0	1	---

Figure 5.13. *Adjacency matrix for graph in Figure 5.12.*

Another way of thinking about this matrix is to notice that it tells us whether there is a path from each actor to each actor. A one represents the presence of a path, a zero represents the lack of a path. The adjacency matrix is exactly what its name suggests -- it tells us which actors are adjacent, or have a direct path from one to the other.

Now suppose that we multiply this adjacency matrix times itself (i.e. raise the matrix to the 2nd power, or square it). We will treat "self-ties" as zeros, which, effectively, ignores them. The calculation of the matrix squared is shown as Figure 5.14.

(0*0)+(1*0)+(1*1)+(0*0)	(0*1)+(1*0)+(1*1)+(0*0)	(0*1)+(1*1)+(1*0)+(0*1)	(0*0)+(1*0)+(1*1)+(0*0)
(0*0)+(0*0)+(1*1)+(0*0)	(0*1)+(0*0)+(1*1)+(0*0)	(0*1)+(0*1)+(1*0)+(0*1)	(0*0)+(0*0)+(1*1)+(0*0)
(1*0)+(1*0)+(0*1)+(1*0)	(1*1)+(1*0)+(0*1)+(1*0)	(1*1)+(1*1)+(0*0)+(1*1)	(1*0)+(1*0)+(0*1)+(1*0)
(0*0)+(0*0)+(1*1)+(0*0)	(0*1)+(0*0)+(1*1)+(0*0)	(0*1)+(0*1)+(1*0)+(0*1)	(0*0)+(0*0)+(1*1)+(0*0)

equals:

1	1	1	1

1	1	0	1
0	1	3	0
1	1	0	1

Figure 5.14. *Squaring matrix 5.13.*

This matrix (i.e. the adjacency matrix squared) counts the number of pathways between two nodes that are of length two. Stop for a minute and verify this assertion (go back to the graph and find the paths). For example, note that actor "B" is connected to each of the other actors by a pathway of length two; and that there is no more than one such pathway to any other actor. Actor T is connected to himself by pathways of length two, three times. This is because actor T has reciprocal ties with each of the other three actors. There is no pathway of length two from T to B (although there is a pathway of length one).

So, the adjacency matrix tells us how many paths of length one are there from each actor to each other actor. The adjacency matrix squared tells us how many pathways of length two are there from each actor to each other actor. It is true (but we won't show it to you) that the adjacency matrix cubed counts the number of pathways of length three from each actor to each other actor. And so on...

If we calculated the Boolean product, rather than the simple matrix product, the adjacency matrix squared would tell us whether there was a path of length two between two actors (not how many such paths there were). If we took the Boolean squared matrix and multiplied it by the adjacency matrix using Boolean multiplication, the result would tell us which actors were connected by one or more pathways of length three. And so on...

Now, finally: why should you care?

Some of the most fundamental properties of a social network have to do with how connected the actors are to one another. Networks that have few or weak connections, or where some actors are connected only by pathways of great length may display low solidarity, a tendency to fall apart, slow response to stimuli, and the like. Networks that have more and stronger connections with shorter paths among actors may be more robust and more able to respond quickly and effectively. Measuring the number and lengths of pathways among the actors in a network allow us to index these important tendencies of whole networks.

Individual actor's positions in networks are also usefully described by the numbers and lengths of pathways that they have to other actors. Actors who have many pathways to other actors may be more influential with regard to them. Actors who have short pathways to more other actors may me more influential or central figures. So, the number and lengths of pathways in a network are very important to understanding both individual's constraints and opportunities, and for understanding the behavior and potentials of the network as a whole.

There are many measures of individual position and overall network structure that are based on whether there are pathways of given lengths between actors, the length of the shortest pathway between two actors, and the numbers of pathways between actors. Indeed, most of the basic measures of networks, measures of centrality and power, and measures of network groupings and substructures are based on looking at the numbers and lengths of pathways among actors.

For most analyses, you won't have to manipulate matrices -- UCINET and other programs have already built algorithms that have the compute do these operations. Most of the computational work in network analysis is done with matrix mathematics though, so in order to understand what is going on, it's useful to understand the basics.

5.E: Using matrices to represent social relations (Exercises)

Review Questions

1. A matrix is "3 by 2." How many columns does it have? How many rows?

2. Adjacency matrices are "square" matrices. Why?

3. There is a "1" in cell 3,2 of an adjacency matrix representing a sociogram. What does this tell us?

4. What does it mean to "permute" a matrix, and to "block" it?

Application Questions

1. Think of the readings from the first part of the course. Did any studies present matrices? If they did, what kinds of matrices were they (that is, what is the technical description of the kind of graph or matrix). Pick one article, and show what the data would look like, if represented in matrix form.

2. Think of some small group of which you are a member (maybe a club, or a set of friends, or people living in the same apartment complex, etc.). What kinds of relations among them might tell us something about the social structures in this population? Try preparing a matrix to represent one of the kinds of relations you chose. Can you extend this matrix to also describe a second kind of relation? (e.g. one might start with "who likes whom?" and add "who spends a lot of time with whom?").

3. Using the matrices you created in the previous question, does it make sense to leave the diagonal "blank," or not, in your case? Try permuting your matrix, and blocking it.

4. Can you make an adjacency matrix to represent the "star" network? what about the "line" and "circle." Look at the ones and zeros in these matrices -- sometimes we can recognize the presence of certain kinds of social relations by these "digital" representations. What does a strict hierarchy look like? What does a population that is segregated into two groups look like?

5.S: Using Matrices to Represent Social Relations (Summary)

Matrices are collections of elements into rows and columns. They are often used in network analysis to represent the adjacency of each actor to each other actor in a network. An adjacency matrix is a square actor-by-actor (i=j) matrix where the presence of pair wise ties are recorded as elements. The main diagonal, or "self-tie" of an adjacency matrix is often ignored in network analysis.

Sociograms, or graphs of networks can be represented in matrix form, and mathematical operations can then be performed to summarize the information in the graph. Vector operations, blocking and partitioning, and matrix mathematics (inverses, transposes, addition, subtraction, multiplication and Boolean multiplication), are mathematical operations that are sometimes helpful to let us see certain things about the patterns of ties in social networks.

Social network data are often multiplex (i.e. there are multiple kinds of ties among the actors). Such data are represented as a series of matrices of the same dimension with the actors in the same position in each matrix. Many of the same tools that we can use for working with a single matrix (matrix addition and correlation, blocking, etc.) Are helpful for trying to summarize and see the patterns in multiplex data.

Once a pattern of social relations or ties among a set of actors has been represented in a formal way (graphs or matrices), we can define some important ideas about social structure in quite precise ways using mathematics for the definitions. In the remainder of the book, we will look at how social network analysts have formally translated some of the core concepts that social scientists use to describe social structures.

CHAPTER OVERVIEW

6: Working with Network Data

This chapter is about the kinds of "data structures" that network analysts work with most frequently, and some of the most common kinds of transformations and manipulations of these structures.

6.1: Manipulating Network Data Structures

Data Structures

Most everyone reading this is very familiar with the kind of "data structure" that is used in many statistical studies. The rectangular array of data that we are used to seeing in SPSS, SAS, Excel, and other programs is a "structure" that is defined by its rows (which represent cases) and columns (which represent variables). An example is shown as Table 6.1.

Table 6.1: Rectangular data array.

ID	Sex	Age	Married
Bob	M	42	1
Carol	F	44	1
Ted	M	39	0
Alice	F	27	0

Earlier, we emphasized that the social network perspective leads us to focus our attention on the relations between actors, more than on the attributes of actors. This approach often results in data that have a different "structure" in which both rows and columns refer to the same actors, and the cells report information on one variable that describes variation (in the case of the example below, simple presence or absence of a tie) in the relations between each pair of actors. An example is given as Table 6.2.

Table 6.2. Square data structure for social network data

	Bob	Carol	Ted	Alice
Bob	---	1	0	0
Carol	0	---	1	0
Ted	1	1	---	1
Alice	0	0	1	---

A "data structure" is simply the way in which information is recorded. These two examples are both two-dimensional (rows and columns). It is possible for a data structure or data object to have more than two dimensions. For example, if we wanted to also record information about the network relations of who is married to whom, we would usually create another table of actors by actors (that is, the row and column indexes would be the same), and record the presence or absence of marital ties. If we "stacked" the two tables together, would would have a 4 by 4 by 2 "data structure". Counts of the rows, columns, and matrices (or "slices") do not include the labeling or indexing information (i.e. it's not 5 x 5 x3).

Social Network Analysis Data Structures

Network analysts work with a variety of data structures. In this chapter, we'll look at tools for creating and manipulating the most common types.

One major "type" of data structure is the actor-by-actor matrix (like the friendship data above). This kind of structure is, by definition a "two-dimensional" and "square" (the number of rows and columns are equal) structure. The information in each cell provides information about the relation between a particular pair of actors.

The two-dimensional actor-by-actor matrix is very often expanded into a "third dimension" by adding "slices" that represent additional kinds of relations among the actors. For example, we might have an actor-by-actor matrix of Bob, Carol, Ted, and Alice that records the degree of "liking" directed from each to each. In addition, we might add a second "slice" that records the presence or absence of a kinship relation between each pair. These kinds of 3-dimensional network data structures are "multiplex". That is, they represent multiple relations among the same set of actors. Some of the special issues in working with multiplex data are discussed in Chapter 15.

The other major "type" of data structure that network analysts use looks a lot like the "rectangular data array" from conventional statistical work. The data structure consists of rows (representing actors) by columns (representing attributes of each actor - what would be called "variables" in statistics). Such an array might record just one attribute, in which case the data structure would be a "column vector". Or, such an array might record a number of attributes of each actor. Network analysts think of this kind of "rectangular" array of actors by attributes simply as a collection of vectors.

The "rectangular" data structure (called an "attribute" data set) is used in a number of ways in network analysis.

- It can record attributes of each actor that we know from other sources (e.g. gender, age, etc.).
- It can record attributes of each actor that arise from their position in the network itself (e.g. the "between-ness centrality" score of each actor).
- It can record what part or sub-part of a network an actor falls in. For example, a column in an "attribute" data structure might consist of the letters "A", "B", and "C" to indicate which of three "factors" each actor was a member of. This is called "partition".
- It can be used to tell UCINET how the actors in a matrix are to be re-arranged, or "permuted"

The "rectangular" data structure can also be used to record information about the relationships between two types of nodes (called bi-partite data). This use is so common and so important that it has a special name - an "incidence" or an "affiliation" matrix. For example, the rows might be indexed by actors (e.g. Bob, Carol...), but the columns might be the organizations that employ the actors (IBM, Sun, Microsoft...). Entries in the cells indicate the presence or strength of the relation between an actor and an employer.

Incidence or affiliation data is particularly important in many social network analyses because it is "multi-level". Actors may be tied together because they are present in the same place, time, or category (that is, they are in the same "incident", or are "affiliated" with the same structure). But such data also show how "incidents" are tied together by the "co-presence" of actors. Incidence data involving two kinds of actors (bi-partite data) are very important in network analysis because they are often our best window into questions of "agency and structure" or "macro-micro linkages".

In this chapter we will describe some of the most common kinds of manipulations that social network analysts use in creating data structures, and changing their structures to meet the needs of particular research questions. Even though this chapter is going to be a bit long, it hardly covers all the possibilities. Different questions require different data structures. The examples and tools here will get you started.

6.2: Making UCINET Datasets

UCINET datasets are stored in a special (Pascal) format, but can be created and manipulated using both UCINET's and other software tools (text editors and spreadsheets). Each UCINET dataset consists of two separate files that contain header information (e.g. myfile.##h) and the data lines (e.g. myfile.##d). Because of this somewhat unusual way of storing data, it is best to create data sets with the internal spreadsheet editor or DL language tools, or to import text or spreadsheet files and save the results as UCINET files.

There are several ways of creating data files that UCINET can read.

The spreadsheet editor. UCINET has a built-in spreadsheet editor that can be used to enter case and variable labels and data values (*data>Spreadsheets>matrix*). This editor allows you to specify the number of rows and column, and has the nice feature of being able to specify that a data set is symmetric. If we are recording a data set where ties among actors are not directed, this feature saves half the data entry. There are also tools to fill the matrix with zeros (a common starting point for many data sets that have sparse connections among actors), permuting rows, symmetrizing and dichotomizing (see discussions in the sections below), and copying row labels to the column labels (if the data are symmetric, you need only enter the labels once).

The UCINET spreadsheet editor can import and export Excel spreadsheets, so you can use tools in both programs to full advantage. To import Excel to UCINET, be sure to save your spreadsheet as version 4 or earlier; the multi-sheet format of more recent Excel versions isn't supported in UCINET.

If you have a fairly small dataset, the UCINET spreadsheet editor is a good choice for making single matrix datasets, which are automatically saved as UCINET files that can be used by other parts of the program.

Importing (and Exporting). Data sets can be moved from a number of other program's data file formats into UCINET's. The *Data>Import>*... menu item supports import from NetDraw (VNA format), Pajek, Krackplot, and Negopy. It also supports importing raw ASCII text files, and files saved as Excel spreadsheets (version 4 or earlier). So, if you started with a NetDraw drawing, for example, and saved the results as VNA, you may import this into UCINET for calculating network measures. I'm more comfortable with Excel than with UCINET's editor, so I usually make data sets in Excel, and import them.

When UCINET imports a file, it will produce a window with your results. Check to make sure they are correct! When the import is performed, UCINET automatically saves the data files in UCINET format in the default directory.

It's often a good idea to set up a new directory for each project, and to set the default to this new directory using the file-cabinet icon on the toolbar, or *File>Change default* folder.

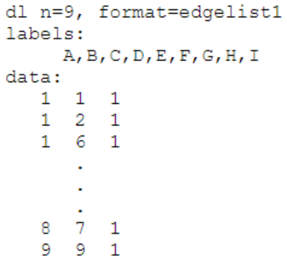
UCINET datasets can also be exported for use in other programs. *Data>Export>*... will produce Excel, raw ASCII text, Pajek, Mage, Metis, and Krackplot files.

The DL Language: If you've been looking at the UCINET Data menu as you read the preceding discussion, you may have noted that the program imports and exports "DL" files. DL (for "data language") is a very powerful and (fairly) simple language that allows the creation of quite complex and large UCINET data sets with minimal data entry.

DL language files are plain ASCII text files that can be created with any editor (be sure to store the results as plain text). A quite complete reference guide is provided in UCINET (*Help>Help Topics>DL*).

The DL language can be a bit picky, and it does take a little effort to figure out how to do exactly what you want to do with it. But, there are a number of circumstances where it is well worth the effort -- when compared to using a spreadsheet. Particularly, if your data set consists of multiple matrices, and if the data are fairly sparse, or if the data set has many rows and columns; then the DL file is the right way to go.

We won't explore the language in any detail here -- the help file is quite good. Figure 6.3 shows an example of a DL file that illustrates a few of the features.



```
dl n=9, format=edgelist1
labels:
     A,B,C,D,E,F,G,H,I
data:
   1  1  1
   1  2  1
   1  6  1
      .
      .
      .
   8  7  1
   9  9  1
```

Figure 6.3: *Example DL language file.*

The file begins with "dl" to indicate file type, and specification of the dimension of the data structure (the language allows specification of number of rows, columns, and matrices). Labels for the nodes are given in the "labels:" paragraph. The data are given in a "data:" paragraph.

The interesting thing in this example is the use of the *format=edgelist1* command. This tells UCINET to read the data lines in a way that is very efficient. The *edgelist1* format is a set of rows, each one of which identifies two nodes and the value of the connection between them. In the resulting data set, all entries are zero, except those that have been specified. So, among our nine actors, there is a tie from actor 1 to actor 1, a tie from actor 1 to actor 2, a tie from actor 1 to actor 6, etc. Here, the matrix is binary -- the value of each tie (the third entry on each line) is 1.

Another very useful *format=* method is *nodelist1*. In this format, each line of data consists of the name (or number) of an origin node, followed by all of the nodes to which it has a connection (this particular format is for zero/one data on the presence or absence of a connection). This approach then requires only one line of data for each actor. For example, a line in the *data:* section that reads: 3 5 6 19 24 would indicate that actor number 3 had a binary directed tie to actors 5, 6, 19, and 24.

These, and other methods available in DL allow the entry of very large and complex data sets with the greatest efficiency and minimum typing. If you are facing a problem with many cases, connections, or kinds of connections, invest a little time in DL.

6.3: Transforming Data Values

It is not at all unusual for the analyst to want to change the values that describe the relations between actors, or the values that describe the attributes of actors. Suppose the attribute "gender" had been entered into a data set using the values "1" and "2", and we wanted to change the attribute to be "Female" coded as "0" and "1". Or, suppose that we had recorded the strength of ties between companies by counting the number of members of boards of directors that they had in common. But we then decide that we are only interested in whether there are members in common or not. We need to change the coded values to be equal to "0" if there are no board members in common, and "1" if there are any (regardless of how many).

Just like statistical packages, UCINET has built-in tools that do some of the most common data transformations.

Transform>Recode is a very flexible and general purpose tool for recoding values in any UCINET data structure. Its dialog box has two tabs: "*Files*" and "*Recode*".

In the *Files* tab, you can browse for an input dataset, select which matrices in the set to recode (if there is more than one), which rows and columns to recode (this is good if you are working on a collection of attribute vectors, for example, and only want to recode selected ones), whether to recode the values on the diagonal, and the name of the output dataset.

In the *Recode* tab, you specify what you want done by creating rules. For example, if I wanted to recode all values 1, 2, and 3 to be zero; and any values of 4, 5, and 6 to be one, I would create two rules. "Values from 1 to 3 are recorded as 0." "Values from 4 to 6 are recoded as 1." The rules are created by using simple built-in tools.

Almost any transformation in a dataset of any complexity can be done with this tool. But, often there are simpler ways to do certain tasks.

Transform>Reverse recodes the values of selected rows, columns, and matrices so that the highest value is now the lowest, the lowest is now the highest, and all other values are linearly transformed. For example, the vector 1 3 3 5 6 would become the vector 6 4 4 2 1.

If we've coded a relationship as "strength of tie" but want our results to be about "weakness of tie" a "reverse" transform would be helpful.

Transform>Dichotomize is a tool that is useful for turning valued data into binary data. That is, if we have measured the strength of ties among actors (e.g. on a scale from 0 = no tie to 5 = strong tie), the "dichotomize" can be used to turn this into data that represent only the absence or presence of a tie (e.g. zero or one).

Why would one ever want to do this? To convert an ordinal or interval measure of relation strength into simple presence/absence may throw away considerable amounts of information. Many of the tools of social network analysis were developed for use with binary data only, and give misleading results (or none at all!) when applied to valued data. Many of the tools in UCINET that are designed for binary data will arbitrarily dichotomize interval or ordinal data in ways that might not be appropriate for your problem.

So, if your data are valued, but the tool you want to use requires binary data, you can turn your data into zero-one form by selecting a cut-off value (you will also have to select a "cut-off operator" and decide what to do with the diagonal).

Suppose, for example, I'd measured tie strength on a scale from 0 to 3. I'd like to have the values 0 and 1 treated as "0" and the values 2 and 3 treated as "1". I would select "greater than" as the cut-off operator, and select a cut-off value of "2". The result would be a binary matrix of zeros (when the original scores were 0 or 1) and ones (when the original scores were 2 or 3).

This tool can be particularly helpful when examining the values of many network measures. For example, the shortest distance between two actors ("geodesic distance") might be computed and saved in a file. We might then want to look at a map or image of the data at various levels of distance -- first, only display actors who are adjacent (distance = 1), then actors who are one or two steps apart, etc. The "dichotomize" tool could be used to create the necessary matrices.

Transform>Diagonal lets you modify the values of the ties of actors with themselves, or the "main diagonal" of square network data sets. The dialog box for this tool allows you to specify either a single value that is to be entered for all the cells on the diagonal; or, a list of (comma separated) values for each of the diagonal cells (from actor one through the last listed actor).

For many network analyses, the values on the main diagonal are not very meaningful, and you may wish to set them all to zero or to one -- which are pretty common approaches. Many of the tools for calculating network measures in UCINET will automatically ignore the main diagonal, or ask you whether to include it or not.

On some occasions, though, you may wish to be sure that ties of an actor with themselves are treated as present (e.g.set diagonal values to 1), or treated as absent (e.g. set diagonal values to zero).

Transform>Symmetrize is a tool that is used to turn "directed" or "asymmetric" network data into "un-directed" or "symmetric" data.

Many of the measures of network properties computed by UCINET are defined only for symmetric data (see the help screens for information about this). If you ask to calculate a measure that is defined for only symmetric data, but your data are not symmetric, UCINET either will refuse to calculate a measure, or will symmetrize the data for you.

But, there are a number of ways to symmetrize data, and you need to be sure that you choose an approach that makes sense for your particular problem. The choices that are available in the *Transform>Symmetrize* tool are:

>Maximum looks at each cell in the upper part of the matrix and the corresponding cell in the lower part of the matrix (e.g. cell 2,5 is compared to cell 5,2), and enters the larger of the values found into both cells (e.g. 2,5 and 5,2 will now have the same output value). For example, suppose that we felt that the strength of the tie between actor A and actor B was best represented as being the strongest of the ties between them (either A's tie to B, or B's tie to A, whichever was strongest).

>Minimum characterizes the strength of the symmetric tie between A and B as being the weaker of the ties AB or BA. This corresponds to the "weakest link", and is a pretty common choice.

>Average characterizes the strength of the symmetric tie between A and B as the simple average of the ties AB and BA. Honestly, I have trouble thinking of a case where this approach makes a lot of sense for social relations.

>Sum characterizes the strength of the symmetric tie between A and B as the sum of AB and BA. This does make some sense -- that all the tie strength be included, regardless of direction.

>Difference characterizes the strength of the symmetric tie between A and B as |AB - BA|. So, relationships that are completely reciprocal end up with a value of zero; those that are completely asymmetric end up with a value equal to the stronger relation.

>Product characterizes the strength of the symmetric relation between A and B as the product of AB and BA. If reciprocity is necessary for us to regard a relationship as being "strong" then either "sum" or "product" might be a logical approach to symmetrizing.

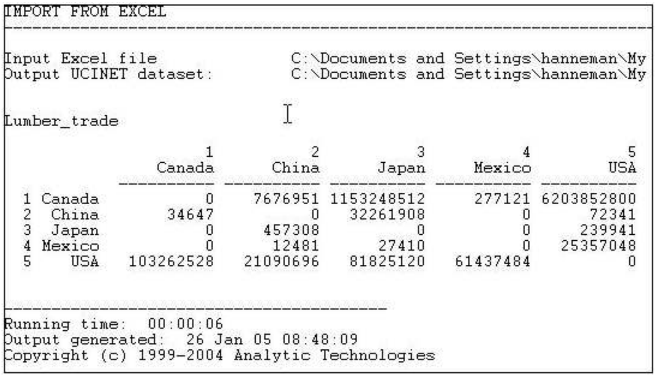
>Division characterizes the strength of the symmetric relation between A and B as AB/BA. This approach "penalizes" relations that are equally reciprocated, and "rewards" relations that are strong in one direction, but not the other.

>Lower Half or *>Upper Half* uses the values in one half of the matrix for the other half. For example, the value of BA is set equal to whatever AB is. This transformation, though it may seem odd at first, is quite helpful. If we are interested in focusing on the network properties of "senders" we would choose to set the lower half equal to the upper half (i.e. select Upper Half). If we were interested in the structure of tie receiving, we would set the upper half equal to the lower.

>Upper > Lower or *>Upper < Lower* (and similar functions available in the dialog box) compare the values in cell AB and BA, and return one or the other based on the test function. If, for example, we had selected Upper > Lower and AB = 3 and BA = 5, the function would select the value "5", because the upper value (AB) was not greater than the lower value (BA).

Transform>Normalize provides a number of tools for rescaling the scores in rows, in columns, or in both dimensions of a matrix of valued relations. A simple example might be helpful.

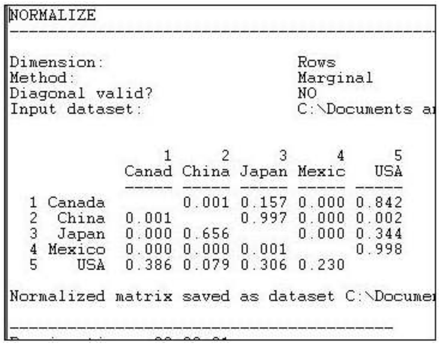
Figure 6.4 shows some data (from the United Nations Commodity Trade database) on trade flows, valued in dollars, of softwood lumber among 5 major Pacific Rim nations at c. 2000.



```
IMPORT FROM EXCEL
--------------------------------------------------------------------------------

Input Excel file               C:\Documents and Settings\hanneman\My
Output UCINET dataset:         C:\Documents and Settings\hanneman\My


Lumber_trade

                      1          2          3          4          5
                 Canada      China      Japan     Mexico        USA
             ---------- ---------- ---------- ---------- ----------
  1 Canada            0    7676951 1153248512     277121 6203852800
  2  China        34647          0   32261908          0      72341
  3  Japan            0     457308          0          0     239941
  4 Mexico            0      12481      27410          0   25357048
  5    USA    103262528   21090696   81825120   61437484          0


----------------------------------------
Running time:  00:00:06
Output generated:  26 Jan 05 08:48:09
Copyright (c) 1999-2004 Analytic Technologies
```

Figure 6.4: *Value of softwood lumber exports among five nations.*

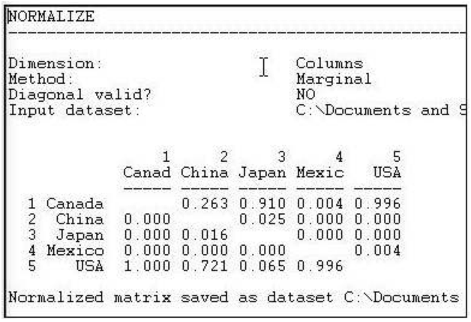
Suppose we were interested in exploring the structure of export partner dependence -- the disadvantage that a nation might face in establishing high prices when it has few alternative places to sell its products. For this purpose, we might choose to "normalize" the data by expressing it as row percentages.That is, what proportion of Canada's exports went to China, Japan, etc. Using the row normalization routine, we produce Figure 6.5.



```
NORMALIZE
--------------------------------------------

Dimension:                    Rows
Method:                       Marginal
Diagonal valid?               NO
Input dataset:                C:\Documents a


                1     2     3     4     5
            Canad China Japan Mexic   USA
            ----- ----- ----- ----- -----
  1 Canada        0.001 0.157 0.000 0.842
  2  China  0.001       0.997 0.000 0.002
  3  Japan  0.000 0.656       0.000 0.344
  4 Mexico  0.000 0.000 0.001       0.998
  5    USA  0.386 0.079 0.306 0.230

Normalized matrix saved as dataset C:\Docume

----------------------------------------
```

Figure 6.5: *Row (sending or export) normalized lumber trade data.*

Graphing the original trade-flow data would answer our question, but graphing the row normalized data gives us a much clearer picture of export dependency. If we were interested in import partner trading concentration, we might normalize the data by columns, producing Figure 6.6.



```
NORMALIZE
--------------------------------------------

Dimension:                    Columns
Method:                       Marginal
Diagonal valid?               NO
Input dataset:                C:\Documents and S


                1     2     3     4     5
            Canad China Japan Mexic   USA
            ----- ----- ----- ----- -----
  1 Canada        0.263 0.910 0.004 0.996
  2  China  0.000       0.025 0.000 0.000
  3  Japan  0.000 0.016       0.000 0.000
  4 Mexico  0.000 0.000 0.000       0.004
  5    USA  1.000 0.721 0.065 0.996

Normalized matrix saved as dataset C:\Documents
```

Figure 6.6: *Column (receiving or import) normalized lumber trade data.*

We see, for example, that all of Canada's imports are from the USA, and that virtually all of the USA's imports are from Canada.

The *>Transform>Normalize* tool provides a number of ways of re-scaling the data that are frequently used.

Normalization may be applied to either rows or columns (as in our examples, above), or it may be applied to the entire matrix (for example, rescaling all trade flows as percentages of the total amount of trade flow in the whole matrix). Normalization may also be applied to both rows and columns, iteratively. For example, if we wanted an "average" number to put in each cell of the trade flow matrix, so that both the rows and the columns summed to 100%, we could apply the iterative row and column approach. This is sometimes used when we want to give "equal weight" to each node, and to take into account both outflow (row) and inflow (column) structure.

There are a number of alternative, commonly used, approaches to how to rescale the data. Our examples use the "marginal" total (row or column) and set the sum of the entries in each row (or column) to equal 1.0. Alternatively, we might want to express each entry relative to the mean score (e.g.divide each element of the row by the average of the elements in a row). Alternatively, one might rescale by dividing by the standard deviation, or both mean and standard deviation (i.e. express the elements as Z scores). UCINET supports all of these as built-in functions. In addition, scores can be normalized by Euclidean norm, or by expressing each element as a percentage of the maximum element in a row.

Rescaling transforms like these can be very, very helpful in highlighting structural features of the data. But, obviously different normalizing approaches highlight very different features. Try thinking through how what applying each of the available transformations would tell you for some data that describe a relation that you are interested in. Some of the transformations will be completely useless; some may give you some new ideas.

6.4: File Handling Tools

Because UCINET data files are stored in a somewhat unusual dual-file format, it is usually most convenient to do basic file-handling tasks within UCINET. The program has basic file handling tools within it. If you use file handling commands outside UCINET (e.g. using Windows), you need to remember to deal with both files for each data set.

File utilities:

File>Copy UCINET Dataset

File>Rename UCINET Dataset

File>Delete UCINET Dataset

These commands do exactly what usual operating system commands do, but manage both component files with a single command.

Viewing the contents of files:

Data>Browse is a tool for examining parts of a dataset. You select the dataset, and specify which rows, columns, and labels you would like to see. This can be very useful if the dataset you're working with is a large one, and your interest is in examining only a portion of it.

Data>Display also allows you to modify how you see a data file. You may set field width, numbers of decimals to display, whether to show zeros or not; in addition, you can select which rows and/or columns to display (the row and column numbers are specified as comma delimited lists, and can use "AND" and "OR"). If the data have been grouped into "blocks", and the block memberships have been stored as UCINET datasets, these may be used to present the data with members of blocks adjacent to one another.

Data>Describe provides basic information about a file (number of rows, columns, matrices). It also shows labels, and allows you to import row and column labels from an external text file (just prepare an ASCII text file with the labels in rows, or comma delimited). You can also use this little utility to add a longer descriptive title to a UCINET dataset. This is often a good idea if you are working with a number of related datasets with similar names.

6.5: Selecting Subsets of the Data

As we work on understanding the structure of a social network, there are occasions when we may wish to focus our attention on only a portion of the actors. Sometimes it's just a matter of clearing away "underbrush" of nodes that aren't "important". Sometimes it's a matter of selecting sets of actors for separate consideration.

UCINET has a number of built-in tools that can be useful for creating new datasets from existing datasets, that include only portions of the actors.

Data>Extract is a general-purpose tool that allows you to either "keep" or to "delete" rows, columns, or matrices for output to a new dataset. You may select the rows, columns, or relations (matrices) to keep by listing them in external data files, or by choosing the names of the rows, columns, or matrices from drop-down lists.

Data>Extract main component retains all the nodes and relations among nodes that are part of the largest component of a graph. In English: the information about the actor and connections among the actors who are part of the largest set of actors who are all connected is retained. If a graph contains several components (e.g. if there are some "isolates" or there are subgroups who have no connection to the largest group) only the largest will be retained. Many analyses require that all the nodes be connected. But, not all real networks actually are. So, you may wish to extract the largest component and analyze it.

Data>Subgraphs from partitions is a (somewhat more complicated) tool that lets you divide the cases into groups (partitions), and output separate data files for each group. The first step (after you've decided which cases fall in which partition), is to create an external data file that lists partition membership. Suppose I wanted to put nodes 1, 3, and 5 in one value of a partition (i.e. in one group) and cases 2, 4, and 6 in another. I'd create a data file that looked like: 1, 2, 1, 2, 1, 2. This says, put the first node in partition one, put the second node in partition two, put the third node in partition one, etc. This filename is supplied to the *>Subgraphs from partitions* dialog. You may also limit the process by electing to output only certain partitions (list them in the dialog window), and/or to create new datasets for a partition value only if there are more than some number (which you specify) of cases.

Many network analysis algorithms generate information on partition membership (and save partition membership information as files you can plug in to this utility). You might also want to impose your own partitions to identify people in one community, people of a particular gender, etc.

Data>Remove isolates creates a new dataset that contains all cases that are not isolated. An "isolate" is a case that has no connections at all to any other actors. Sometimes, when we collect information by doing a census of all the actors of a given type, or in a given location, some are "isolated". While this is usually an interesting social fact, we may wish to focus our attention on the community of actors who are connected (though not necessarily forming a single "component").

Data>Remove pendants creates a new dataset that contains all cases that are not "pendants". A "pendant" is a case that is connected to the graph by only one tie; cases like these will "dangle" off of more central cases that are more heavily connected. In looking at large graphs with many actors, we may wish to limit out attention to nodes that are connected to at least two other actors -- so as to focus attention on the "core" of the network. Removing isolates and pendants can help to clear some of the "clutter".

Data>Egonet is a tool that lets us extract particular actors and those in their immediate "neighborhood" as separate datasets. As we will see later on, the "ego-network" of a single actor, or of some selection of actors (all men, all cases with high between-ness, etc.) is often the focus of investigation more than the structure of the whole network.

An "ego-network" is the set of actors who are connected to a focal actor, along with the relations between ego and the alters, and any relations among the alters. The structure of ego networks (whether they are dense or thin, and whether they contain "structural holes") are often critical variables in understanding and predicting the behavior of "ego".

The *Data>Egonet* tool lets you list the "egos" or "focal nodes" you'd like to extract by using an external file list or by selecting their labels from a drop-down list. The dialog asks whether you want to include ego, or only to retain information on ego's neighbors; the most common, and default, choice is to include ego as well as ego's neighbors.

Data>Unpack is a tool for creating a new dataset that contains a subset of matrices from a larger dataset. For example, if we had stored information of both "liking" and "spouse" relation in a single dataset, we can use this utility to create separate data files for one or both relations. The relations to be "unpacked" are selected from a drop-down box.

Data>Join is a tool that can be used to combine separate sets of data into a new dataset. Often we collect attribute information about actors in several different settings (e.g. several classrooms in a school) and store these as separate files. Or, we may have multiple files that contain information about different attributes of actors (for example, one file might be things we know from outside sources like age, sex, etc.; another file might contain information on which partition of a graph each actor falls into). We might want to combine all the attribute information into a single file. Or, we might have information about different relations among the same set of actors, that have been stored as separate data files (as in the "liking" and "spouse" relations example).

Using *Data>Join>Rows* will combine two or more matrices (stored as separate files) into a single matrix that has rows for all nodes in each of the files. If I had separate files that listed the age of students in each of two classrooms, and I wanted to create a single file with all the students, the "rows" approach would be used.

Using *Data>Join>Columns* will combine two or more matrices (stored as separate files) into a single matrix that has the same number of rows as each of the input files, but appends the columns. If I had information on age and sex for actors A, B, and C in one file and information on centrality and degree for actors A, B, and C in another, I could do a column join to produce a file that listed age, sex, centrality, and degree for actors A, B, and C.

Using *Data>Join>Matrices* will combine information on multiple relations among the same sets of actors into a single file. Each input file has the same actors by actors array, but for different relations. The output file combines the multiple files into a three-dimensional array of actor by actor by relation.

6.6: Making New Kinds of Graphs from Existing Graphs

Turning Attributes into Relations

At the beginning of this chapter we looked at the "data structures" most commonly used in network analysis. One was the node-by-node square matrix, to record the relations between pairs of actors; and its more general "multiplex" form to record multiple relations among the same set of actors. The other was the rectangular matrix. This "actor by attribute" matrix is most commonly used to record information about the variable properties of each node.

Network analysis often finds it useful to see actor attributes as actually indicating the presence, absence, or strength of "relations" among actors. Suppose two persons have the same gender. To the non-network analyst, this may represent a statistical regularity that describes the frequencies of scores on a variable. A network analyst, though, might interpret the same data a bit differently. A network analyst might, instead, say "these two persons share the relation of having the same gender".

Both interpretations are, of course, entirely reasonable. One emphasizes the attributes of individuals (here are two persons, each is a woman); one emphasizes the relation between them (here are two persons who are related by sharing the same social role).

It's often the case that network researchers, whoa re pre-disposed to see the world in "relational" terms, want to turn "attribute" data in "relational" data for their analyses.

Data>Attribute is a tool that creates an actor-by-actor relational matrix from the scores on a single attribute vector. Suppose that we had an attribute vector stored in a UCINET file (other vectors could also be in the file, but this algorithm operates on a single vector), that measured whether each of 100 large donors had given funds in support of (+1) or in opposition to (-1) a given ballot proposition. Those who made no contribution are coded zero.

We might like to create a new matrix that identifies pairs of actors who shared support or shared opposition to the ballot initiative, or who took opposite positions. That is, for each pair of actors, the matrix element is "1" if the actors jointly supported or jointly opposed the proposition, "-1" if one supported and the other opposed, and zero otherwise (if either or both made no contribution).

Using the *Data>Attribute* tool, we can form a new square (actor-by-actor) matrix from the scores of actors on one attribute in a number of ways. The *Exact Matches* choice will produce a "1" when two actors have exactly the same score on the attribute, and zero otherwise. The *Difference* choice will create a new matrix where the elements are the differences between the attribute scores of each pair of actors (alternatively, the *Absolute Difference*, or *Squared Difference* choices will yield positively valued measures of the distance between the attribute scores of each pair of actors). The *Sum* choice yields a score for each pair that is equal to the sum of their attribute scores. In our current example, the *Product* choice (that is, multiply the score of actor i times the score of actor j, and enter the result) would yield a score of "1" if two actors shared either support or opposition, "-1" if they took opposed stands on the issue, or "0" if either did not take a position.

The *Data>Attribute* tool can be very useful for conceptually turning attributes into relations, so that their association with other relations can be studied.

Data>Affiliations extends the idea of turning attributes into relations to the case where we want to consider to multiple attributes. Probably the most common situations of this type are where the multiple "attributes" we have measured are "repeated measures" of some sort. Davis, for example, measured the presence of a number of persons (rows) at a number of parties (attributes or columns). From these data, we might be interested in the similarity of all pairs of actors (how many times where they co-present at the same event?), or how similar were the parties (how much of the attendance of each pair of parties were the same people?).

The example of donors to political campaigns can be seen in the same way. We might collect information on whether political donors (rows) had given funds against or for a number of different ballot propositions (columns). From this rectangular matrix, we might be interested in forming a square actor-by-actor matrix (how often to each pair of actors donate to the same campaigns?); we might be interested in forming a square campaign-by-campaign matrix (how similar are the campaigns in terms of their constituencies?).

The *Data>Affiliations* algorithm begins with a rectangular (actor-by-affiliate) matrix, and asks you to select whether the new matrix is to be formed by rows (i.e. actor-by-actor) or columns (i.e. attribute-by-attribute).

There are different ways in which we could form the entries of the new matrix. UCINET provides two methods: *Cross-Products* or *Minimums*. These approaches produce the same result for binary data, but different results for valued data.

Let's look at the binary case first. Consider two actors "A" and "B" who have made contributions (or not) to each of 5 political campaigns, as in Figure 6.7.

	Campaign 1	Campaign 2	Campaign 3	Campaign 4	Campaign 5
A	0	0	1	1	1
B	0	1	1	0	1

Figure 6.7: *Donations of two donors to five political campaigns (binary data)*

The *Cross-Products* method multiplies each of A's scores by the corresponding score for B, and then sums across the columns (if we were creating a campaign-by-campaign matrix, the logic is exactly the same, but would operate by multiplying columns, and summing across rows). Here, this results in: $(0*0)+(0*1)+(1*1)+(1*0)+(1*1)=2$. That is, actors A and B have two instances where they both supported a campaign.

The *Minimums* method examines the entries of A and B for Campaign 1, and selects the lowest score (zero). It then does this for the other campaigns (resulting in 0, 1, 0, 1) and sums. With binary data, the results will be the same by either method.

With valued data, the methods do not produce the same results; they get at rather different ideas.

Suppose that we had measured whether A and B supported (+1) , took no position on (0), or opposed (-1) each of the five campaigns. This is the simplest possible "valued" data, but the ideas hold for valued scales with wider ranges, and with all positive values, as well. Now, our data might look like those in Figure 6.8.

	Campaign 1	Campaign 2	Campaign 3	Campaign 4	Campaign 5
A	-1	0	1	-1	1
B	-1	1	1	0	-1

Figure 6.8: *Donations of two donors for or against five political campaigns (valued data)*

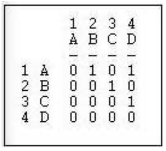
Both A and B took the same position on two issues (both opposed on one, both supporting another). On two campaigns (2, 4), one took no stand. On issue number 5, the two actors took opposite positions.

The *Cross-Products* method yields: $(-1*-1)+(0*1)+(1*1)+(-1*0)+(1*-1)$. That is: 1 + 0 + 1 + 0 + -1, or 1. The two actors have a "net" agreement of 1 (they took the same position on two issues, but opposed positions on one issue).

The *Minimums* method yields: -1 + 0 + 1 - 1 - 1 or -2. In this example, this is difficult to interpret, but can be seen as the net number of times either member of the pair opposed an issue. The minimums method produces results that are easier to interpret when all values are positive. Suppose we re-coded the data to be: 0 = opposed, 1 = neutral, and 2 = favor. The minimums method would then produce 0 + 1 + 2 + 0 + 0 = 3. This might bee seen as the extent to which the pair of actors jointly supported the five campaigns.

Turning Relations into Attributes

Suppose that we had a simple directed relation, represented as a matrix as in Figure 6.9.



```
          1 2 3 4
          A B C D
          - - - -
  1 A     0 1 0 1
  2 B     0 0 1 0
  3 C     0 0 0 1
  4 D     0 0 0 0
```

Figure 6.9: *Linegraph example matrix*

This is easier to see as a graph, as in Figure 6.10.

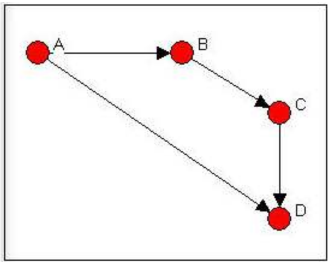
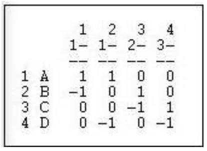


Figure 6.10: *Linegraph example graph*

Now suppose that we are really interested in describing and thinking about the relations, and the relations among the relations - rather than the actors, and the relations among them. That sounds odd, I realize. Let me put it a different way. We can think about the graph in Figure 6.5 as composed of four relations (A to B, B to C, C to D, and A to C). These relations are connected by having actors in common (e.g. the A to B and the B to C relations have the actor B in common). That is, we can think about relations as being "adjacent" when they share actors, just as we can think about actors being adjacent when they share relations.

Transform>Incidence is an algorithm that changes the way we look at a directed graph from "actors connected by relations" to "relations connected by actors". This is sometimes just a semantic trick. But, sometimes it's more than that - our theory of the social structure may actually be one about which relations are connected, not which actors are connected. If we apply the *Transform>Incidence* algorithm to the data in Figures 6.4 and 6.5, we get the result in Figure 6.11.



```
         1  2  3  4
        1- 1- 2- 3-
        -- -- -- --
1 A      1  1  0  0
2 B     -1  0  1  0
3 C      0  0 -1  1
4 D      0 -1  0 -1
```

Figure 6.11: *Incidence matrix of Figure 6.10*

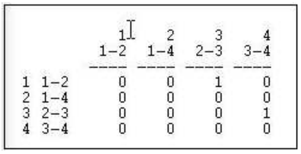
Each row is an actor. Each column is now a relation (the relations are numbered 1 through 4). A positive entry indicates that an actor is the source of a directed relation. For example, actor A is the origin of the relation "1" that connects A to B, and is a source of the relation "2" that connects actor A to actor D. A negative entry indicates that an actor is the "sink" or recipient of a directed relation. For example, actor C is the recipient in relation "3" (which connects actor B to actor C), and the source of relation "4" (which connects actor C to actor D).

The "incidence" matrix here then shows how actors are connected to relationships. By examining the rows, we can characterize how much, and in what ways actors are embedded in relations. One actor may have few entries - a near isolate; another may have many negative and few positive entries - a "taker" rather than a "giver". By examining the columns, we get a description of which actors are connected, in which way, by each of the relations in the graph.

Focusing on the Relations, Instead of the Actors

Turning an actor-by-actor adjacency matrix into an actor-by-relation incidence graph takes us part of the way toward focusing on relations rather than actors. We can go further.

Transform>Linegraph converts an actor-by-actor matrix (like Figure 6.4) into a full relation-by-relation matrix. Figure 6.12 shows the results of applying it to the example data.



```
             1     2     3     4
           1-2   1-4   2-3   3-4
          ----  ----  ----  ----
1 1-2        0     0     1     0
2 1-4        0     0     0     0
3 2-3        0     0     0     1
4 3-4        0     0     0     0
```

Figure 6.12: *Linegraph matrix*

We again have a square matrix. This time, though, it describes which relations in the graph are "adjacent to" which other relations. Two relations are adjacent if they share an actor. For example, relation "1" (the tie between actors 1 and 2, or A and B) is adjacent to the relation "3" (the tie between actors 2 and 3, or B and C). Note that the "adjacency" here is directional - relation 1 is a source of relation 3. We could also apply this approach to symmetric or simple graphs to describe which relations are simply adjacent in an un-directed way.

A quick glance at the linegraph matrix is suggestive. It is very sparse in this example - most of the relations are not sources of other relations. The maximum degree of each entry is 1 - no relation is the source of multiple relations. While there may be a key or central actor (A), it's not so clear that there is a single central relation.

To be entirely honest, most social network analysts do (much of the time) think about actors connected to actors by relations, rather than relations connecting actors, or relations connecting relations. But changing our point of view to put the relations first, and the actors second is, in many ways, a more distinctively "sociological" way of looking at networks. Transforming actor-by-actor data into relation-by-relation data can yield some interesting insights about social structures.

6.S: Working with Network Data (Summary)

In this chapter we've covered a number of rather diverse but related topics. We've described some of the basic "nuts and bolts" tools for entering and transforming network data. The "bigger picture" is to think about network data (and any other, for that matter) as having "structure". Once you begin to see data in this way, you can begin to better imagine the creative possibilities: for example, treating actor-by-attribute data as actor-by-actor, or treating it as attribute-by-attribute. Different research problems may call for quite different ways of looking at, and transforming, the same data structures. We've hardly covered every possibility here, but we have looked at some of the most frequently used tricks.

CHAPTER OVERVIEW

7: Connection

In this chapter we will examine some of the most obvious and least complex ideas of formal network analysis methods. Despite the simplicity of the ideas and definitions, there are good theoretical reasons (and some empirical evidence) to believe that these basic properties of social networks have very important consequences. For both individuals and for structures, one main question is connections. Typically, some actors have lots of connections, others have fewer. Some networks are well-connected or "cohesive", others are not. The extent to which individuals are connected to others, and the extent to which the network as a whole is integrated are two sides of the same coin.

7.1: Networks and Actors

The social network perspective emphasizes multiple levels of analysis. Differences among actors are traced to the constraints and opportunities that arise from how they are embedded in networks; the structure and behavior of networks grounded in, and enacted by local interactions among actors. As we examine some of the basic concepts and definitions of network analysis in this and the next several chapters, this duality of individual and structure will be highlighted again and again.

In this chapter we will examine some of the most obvious and least complex ideas of formal network analysis methods. Despite the simplicity of the ideas and definitions, there are good theoretical reasons (and some empirical evidence) to believe that these basic properties of social networks have very important consequences. For both individuals and for structures, one main question is connections. Typically, some actors have lots of connections, others have fewer. Some networks are well-connected or "cohesive", others are not. The extent to which individuals are connected to others, and the extent to which the network as a whole is integrated are two sides of the same coin.

Differences among individuals in how connected they are can be extremely consequential for understanding their attributes and behavior. More connections often mean that individuals are exposed to more, and more diverse, information. Highly connected individuals may be more influential, and may be more influenced by others. Differences among whole populations in how connected they are can be quite consequential as well. Disease and rumors spread more quickly where there are high rates of connection. But, so too does useful information. More connected populations may be better able to mobilize their resources, and may be better able to bring multiple and diverse perspectives to bear to solve problems. In between the individual and the whole population, there is another level of analysis - that of "composition". Some populations may be composed of individuals who are all pretty much alike in the extent to which they are connected. Other populations may display sharp differences, with a small elite of central and highly connected persons, and larger masses of persons with fewer connections. Differences in connections can tell us a good bit about the stratification order of social groups. A great deal of recent work by Duncan Watts, Doug White and many others outside of the social sciences is focusing on the consequences of variation in the degree of connection of actors.

Because most individuals are not usually connected directly to most other individuals in a population, it can be quite important to go beyond simply examining the immediate connections of actors, and the overall density of direct connections in populations. The second major (but closely related) set of approaches that we will examine in this chapter have to do with the idea of the distance between actors (or, conversely, how close they are to one another). Some actors may be able to reach most other members of the population with little effort: they tell their friends, who tell their friends, and "everyone" knows. Other actors may have difficulty being heard. They may tell people, but the people they tell are not well connected, and the message doesn't go far. Thinking about it the other way around, if all of my friends have one another as friends, my network is fairly limited - even though I may have quite a few friends. But, if my friends have many non-overlapping connections, the range of my connection is expanded. If individuals differ in their closeness to other actors, then the possibility of stratification along this dimension arises. Indeed, one major difference among "social classes" is not so much in the number of connections that actors have, but in whether these connections overlap and "constrain" or extend outward and provide "opportunity". Populations as a whole, then, can also differ in how close actors are to other actors, on the average. Such differences may help us to understand diffusion, homogeneity, solidarity, and other differences in macro properties of social groups.

Social network methods have a vocabulary for describing connectedness and distance that might, at first, seem rather formal and abstract. This is not surprising, as many of the ideas are taken directly from the mathematical theory of graphs. But it is worth the effort to deal with the jargon. The precision and rigor of the definitions allow us to communicate more clearly about important properties of social structures - and often lead to insights that we would not have had if we used less formal approaches.

An Example: Knoke's Information Exchange

The basic properties of networks are easier to learn and understand by example. Studying an example also shows sociologically meaningful applications of the formalisms. In this chapter, we will look at a single directed binary network that describes the flow of information among 10 formal organizations concerned with social welfare issues in one mid-western U.S. city (Knoke and Burke). Of course, network data come in many forms (undirected, multiple ties, valued ties, etc.) and one example can't capture all of the possibilities. Still, it can be rather surprising how much information can be "squeezed out" of a single binary matrix by using basic graph concepts.

For small networks, it is often useful to examine graphs. Figure 7.1 shows the di-graph (directed graph) for the Knoke information exchange data:

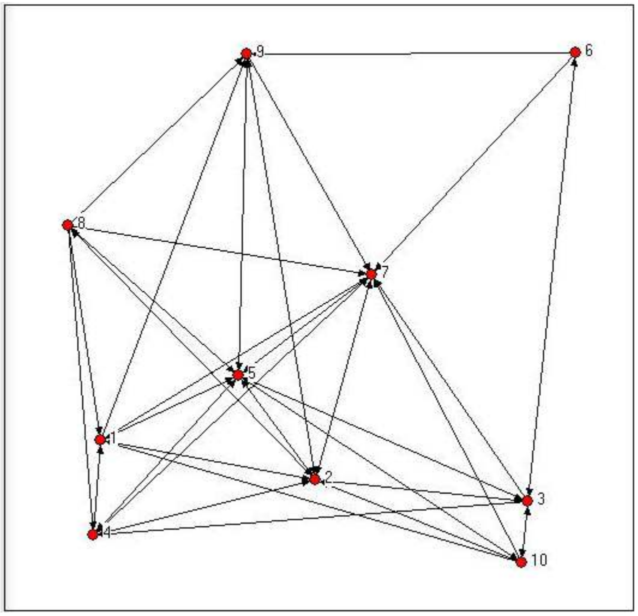
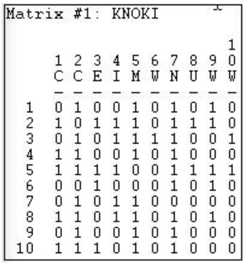


Figure 7.1: *Knoke information exchange directed graph*

Your trained eye should immediately perceive a number of things in looking at the graph. There is a limited number of actors here (ten, actually), and all of them are "connected". But, clearly not every possible connection is present, and there are "structural holes" (or at least "thin spots" in the fabric). There appear to be some differences among the actors in how connected they are (compare actor number 7, a newspaper, to actor number 6, a welfare rights advocacy organization). If you look closely, you can see that some actors' connections are likely to be reciprocated (that is, if A shares information with B, B also shares information with A); some other actors (e.g. 6 and 10), are more likely to be senders than receivers of information. As a result of the variation in how connected individuals are, and whether the ties are reciprocated, some actors may be at quite some "distance" from other actors. There appear to be groups of actors who differ in this regard (2, 5, and 7 seem to be in the center of the action; 6, 9, and 10 seem to be more peripheral).

A careful look at the graph can be very useful in getting an intuitive grasp of the important features of a social network. With larger populations or more connections, however, graphs may not be much help. Looking at a graph can give a good intuitive sense of what is going on, but our descriptions of what we see are rather imprecise (the previous paragraph is an example of this). To get more precise, and to use computers to apply algorithms to calculate mathematical measures of graph properties, it is necessary to work with the adjacency matrix instead of the graph. The Knoke data graphed above are shown as an asymmetric adjacency matrix in Figure 7.2.



```
Matrix #1: KNOKI

                                1
        1 2 3 4 5 6 7 8 9 0
        C C E I M W N U W W
        - - - - - - - - - -
   1    0 1 0 0 1 0 1 0 1 0
   2    1 0 1 1 1 0 1 1 1 0
   3    0 1 0 1 1 1 1 0 0 1
   4    1 1 0 0 1 0 1 0 0 0
   5    1 1 1 1 0 0 1 1 1 1
   6    0 0 1 0 0 0 1 0 1 0
   7    0 1 0 1 1 0 0 0 0 0
   8    1 1 0 1 1 0 1 0 1 0
   9    0 1 0 0 1 0 1 0 0 0
  10    1 1 1 0 1 0 1 0 0 0
```

Figure 7.2: *Knoke information exchange adjacency matrix*

Using *Data>Display*, we can look at the network in matrix form. There are ten rows and columns, the data are binary, and the matrix is asymmetric. As we mentioned in the chapter on using matrices to represent networks, the row is treated as the source of information and the columns as the receiver. By doing some very simple operations on this matrix it is possible to develop systematic and useful index numbers, or measures, of some of the network properties that our eye discerns in the graph.

7.2: Connection

Since networks are defined by their actors and the connections among them, it is useful to begin our description of networks by examining these very simple properties. Focusing first on the network as a whole, one might be interested in the number of actors, the number of connections that are possible, and the number of connections that are actually present. Differences in the size of networks, and how connected the actors are tell us two things about human populations that are critical. Small groups differ from large groups in many important ways - indeed, population size is one of the most critical variables in all sociological analyses. Differences in how connected the actors in a population are may be a key indicator of the "cohesion", "solidarity", "moral density", and "complexity" of the social organization of a population.

Individuals, as well as whole networks, differ in these basic demographic features. Individual actors may have many or few ties. Individuals may be "sources" of ties, "sinks" (actors that receive ties, but don't send them), or both. These kinds of very basic differences among actors' immediate connections may be critical in explaining how they view the world, and how the world views them. The number and kinds of ties that actors have are a basis for similarity or dissimilarity to other actors - and hence to possible differentiation and stratification. The number and kinds of ties that actors have are keys to determining how much their embeddedness in the network constrains their behavior, and the range of opportunities, influence, and power that they have.

Basic Demographics

Network size. The size of a network is often very important. Imagine a group of 12 students in a seminar. It would not be difficult for each of the students to know each of the others fairly well, and build up exchange relationships (e.g. sharing reading notes). Now imagine a large lecture class of 300 students. It would be extremely difficult for any student to know all of the others, and it would be virtually impossible for there to be a single network for exchanging reading notes. Size is critical for the structure of social relations because of the limited resources and capacities that each actor has for building and maintaining ties. Our example network has ten actors. Usually the size of a network is indexed simply by counting the number of nodes.

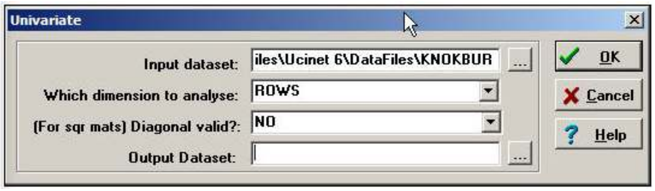
In any network there are $(k*k-1)$ unique ordered pairs of actors (that is AB is different from BA, and leaving aside self-ties), where k is the number of actors. You may wish to verify this for yourself with some small networks. So, in our network of 10 actors, with directed data, there are 90 logically possible relationships. If we had undirected, or symmetric, ties, the number would be 45, since the relationship AB would be the same as BA. The number of logically possible relationships then grows exponentially as the number of actors increases linearly. It follows from this that the range of logically possible social structures increases (or, by one definition, "complexity" increases) exponentially with size.

Actor degree. The number of actors places an upper limit on the number of connections that each individual can have $(k-1)$. For networks of any size, though, few - if any - actors approach this limit. It can be quite useful to examine the distribution of actor degree. The distribution of how connected individual actors are can tell use a good bit about the social structure.

Since the data in our example are asymmetric (that is, directed ties), we can distinguish between ties being sent and ties being received. Looking at the density for each row and for each column can tell us a good bit about the way in which actors are embedded in the overall density.

Tools>Univariate Stats provides quick summaries of the distribution of actor's ties.

Let's first examine these statistics for the rows, or out-degree of actors.



Univariate

Input dataset: iles\Ucinet 6\DataFiles\KNOKBUR

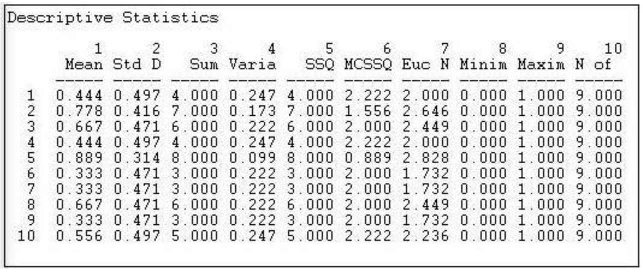
Which dimension to analyse: ROWS

(For sqr mats) Diagonal valid?: NO

Output Dataset:

Figure 7.3: *Dialog for Tools>Univariate Stats*

Produces this result:



```
Descriptive Statistics

          1      2      3      4      5      6      7      8      9     10
       Mean  Std D    Sum  Varia    SSQ  MCSSQ  Euc N  Minim  Maxim  N of
      ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1    0.444  0.497  4.000  0.247  4.000  2.222  2.000  0.000  1.000  9.000
  2    0.778  0.416  7.000  0.173  7.000  1.556  2.646  0.000  1.000  9.000
  3    0.667  0.471  6.000  0.222  6.000  2.000  2.449  0.000  1.000  9.000
  4    0.444  0.497  4.000  0.247  4.000  2.222  2.000  0.000  1.000  9.000
  5    0.889  0.314  8.000  0.099  8.000  0.889  2.828  0.000  1.000  9.000
  6    0.333  0.471  3.000  0.222  3.000  2.000  1.732  0.000  1.000  9.000
  7    0.333  0.471  3.000  0.222  3.000  2.000  1.732  0.000  1.000  9.000
  8    0.667  0.471  6.000  0.222  6.000  2.000  2.449  0.000  1.000  9.000
  9    0.333  0.471  3.000  0.222  3.000  2.000  1.732  0.000  1.000  9.000
 10    0.556  0.497  5.000  0.247  5.000  2.222  2.236  0.000  1.000  9.000
```

Figure 7.4: *Out-degree statistics for Knoke information exchange*

Statistics on the rows tell us about the role that each actor plays as a "source" of ties (in a directed graph). The sum of the connections from the actor to others (e.g. actor #1 sends information to four others) is called the *out-degree* of the point (for symmetric data, of course, each node simply has *degree*, as we cannot distinguish *in-degree* from *out-degree*). The degree of points is important because it tells us how many connections an actor has. With out-degree, it is usually a measure of how influential the actor may be.

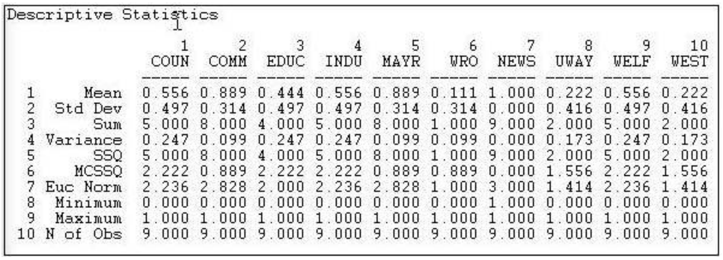
We can see that actor #5 sends ties to all but one of the remaining actors; actors #6, #7, and #9 send information to only three other actors. Actors #2, #3, #5, and #8 are similar in being sources of information for large portions of the network; actors #1, #6, #7 and #9 are similar in not being sources of information. We might predict that the first set of organizations will have specialized divisions for public relations, the latter set might not. Actors in the first set have a higher potential to be influential; actors in the latter set have lower potential to be influential; actors in "the middle" will be influential if they are connected to the "right" other actors, otherwise, they might have very little influence. So, there is variation in the roles that these organizations play as sources of information. We can norm this information (so we can compare it to other networks of different sizes, by expressing the out-degree of each point as a proportion of the number of elements in the row; that is, calculating the mean). Actor #10, for example, sends ties to 56% of the remaining actors. This is a figure we can compare across networks of different sizes.

Another way of thinking about each actor as a source of information is to look at the row-wise variance or standard deviation. We note that actors with very few out-ties, or very many out-ties, have less variability than those with medium levels of ties. This tells us something: those actors with ties to almost everyone else, or with ties to almost no-one else are more "predictable" in their behavior toward any given other actor than those with intermediate numbers of ties. In a sense, actors with many ties (at the center of a network) and actors at the periphery of a network (few ties) have patterns of behavior that are more constrained and predictable. Actors with only some ties can vary more in their behavior, depending on to whom they are connected.

If we were examining a valued relation instead of a binary one, the meaning of the "sum", "mean", and "standard deviation" of actors' out-degree would differ. If the values of the relations are all positive and reflect the strength or probability of a tie between nodes, these statistics would have the easy interpretations as the sum of the strengths, the average strength, and variation in strength.

It's useful to examine the statistics for in-degree, as well (look at the data column-wise). Now, we are looking at the actors as "sinks" or receivers of information. The sum of each column in the adjacency matrix is the *in-degree of the point*. That is, how many other actors send information or ties to the one we are focusing on. Actors that receive information from many sources may be prestigious (other actors want to be known by the actor, so they send information). Actors that receive information from many sources may also be more powerful - to the extent that "knowledge is power". But, actors that receive a lot of information could also suffer from "information overload" or "noise and interference" due to contradictory messages from different sources.

Here are the results of *Tools>Univariate Stats* when we select "column" instead of "row".



```
Descriptive Statistics

                      1      2      3      4      5      6      7      8      9     10
                   COUN   COMM   EDUC   INDU   MAYR    WRO   NEWS   UWAY   WELF   WEST
                 ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1       Mean    0.556  0.889  0.444  0.556  0.889  0.111  1.000  0.222  0.556  0.222
  2    Std Dev    0.497  0.314  0.497  0.497  0.314  0.314  0.000  0.416  0.497  0.416
  3        Sum    5.000  8.000  4.000  5.000  8.000  1.000  9.000  2.000  5.000  2.000
  4   Variance    0.247  0.099  0.247  0.247  0.099  0.099  0.000  0.173  0.247  0.173
  5        SSQ    5.000  8.000  4.000  5.000  8.000  1.000  9.000  2.000  5.000  2.000
  6      MCSSQ    2.222  0.889  2.222  2.222  0.889  0.889  0.000  1.556  2.222  1.556
  7   Euc Norm    2.236  2.828  2.000  2.236  2.828  1.000  3.000  1.414  2.236  1.414
  8    Minimum    0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000
  9    Maximum    1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
 10   N of Obs    9.000  9.000  9.000  9.000  9.000  9.000  9.000  9.000  9.000  9.000
```

Figure 7.5: *In-degree statistics for Knoke information exchange*

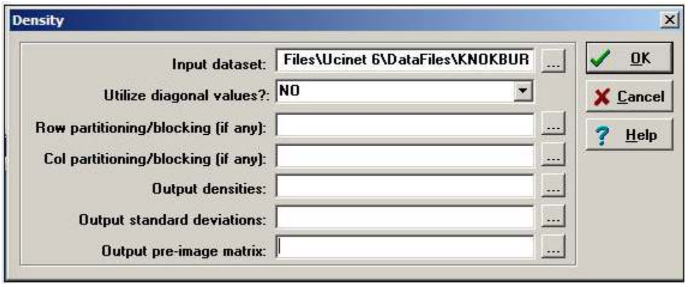
Looking at the means, we see that there is a lot of variation in information receiving - more than for information sending. We see that actors #2, #5, and #7 are very high. #2 and #5 are also high in sending information - so perhaps they act as "communicators" and "facilitators" in the system. Actor #7 receives a lot of information, but does not send a lot. Actor #7, as it turns out, is an "information sink" - it collects facts, but it does not create them (at least we hope so, since actor #7 is a newspaper). Actors #6, #8, and #10 appear to be "out of the loop" - that is, they do not receive information from many sources directly. Actor #6 also does not send much information - so #6 appears to be something of an "isolate". Actors #8 and #10 send relatively more information than they receive. One might suggest that they are "outsiders" who are attempting to be influential, but may be "clueless".

We can learn a great deal about a network overall, and about the structural constraints on individual actors, and even start forming some hypotheses about social roles and behavioral tendencies, just by looking at the simple adjacencies and calculating a few very basic statistics. Before discussing the slightly more complex idea of distance, there are a couple other aspects of "connectedness" that are sometimes of interest.

Density

The density of a binary network is simply the proportion of all possible ties that are actually present. For a valued network, density is defined as the sum of the ties divided by the number of possible ties (i.e. the ratio of all tie strength that is actually present to the number of possible ties). The density of a network may give us insights into such phenomena as the speed at which information diffuses among the nodes, and the extent to which actors have high levels of social capital and/or social constraint.

Network>Cohesion>Density is a quite powerful tool for calculating densities. Its dialog is shown in Figure 7.6.



Density

Input dataset: Files\Ucinet 6\DataFiles\KNOKBUR

Utilize diagonal values?: NO

Row partitioning/blocking (if any):

Col partitioning/blocking (if any):

Output densities:

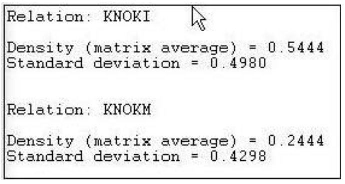
Output standard deviations:

Output pre-image matrix:

Figure 7.6: *Dialog for Network>Cohesion>Density*

To obtain densities for a matrix (as we are doing in this example), we simply need a dataset. Usually self-ties are ignored in computing density (but there are circumstances where you might want to include them). The *Network>Cohesion>Density* algorithm also can be used to calculate the densities within partitions or blocks by specifying the file name of an attribute data set that contains the node name and partition number. That is, the density tool can be used to calculate within and between block densities for data that are grouped. One might, for example, partition the Knoke data into "public" and "private" organizations, and examine the density of information exchange within and between types.

For our current purposes, we won't block or partition the data. Here's the result of the dialog above.



```
Relation: KNOKI

Density (matrix average) = 0.5444
Standard deviation = 0.4980


Relation: KNOKM

Density (matrix average) = 0.2444
Standard deviation = 0.4298
```

Figure 7.7: *Density of Knoke information network*

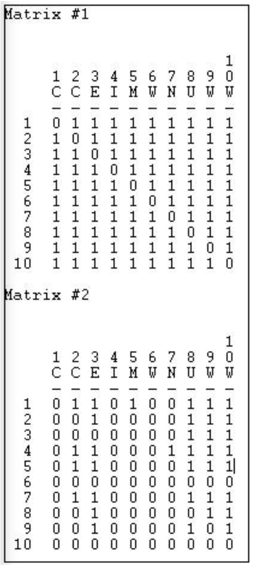
Since the Knoke data set contains two matrices, separate reports for each relation (KNOKI and KNOKM) are produced.

The density of the information exchange matrix is 0.5444. That is, 54% of all the possible ties are present. The standard deviation of the entries in the matrix is also given. For binary data, the standard deviation is largely irrelevant - as the standard deviation of a binary variable is a function of its mean.

Reachability

An actor is "reachable" by another if there exists any set of connections by which we can trace from the source to the target actor, regardless of how many others fall between them. If the data are asymmetric or directed, it is possible that actor A can reach actor B, but that actor B cannot reach actor A. With symmetric or undirected data, of course, each pair of actors either are or are not reachable to one another. If some actors in a network cannot reach others, there is the potential of a division of the network. Or, it may indicate that the population we are studying is really composed of more than one sub-populations.

In the Knoke information exchange dataset, it turns out that all actors are reachable by all others. This is something that you can verify by eye. See if you can find any pair of actors in the diagram such that you cannot trace from the first to the second along arrows all headed in the same direction (don't waste a lot of time on this, there is no such pair!). For the Knoke "M" relation, it turns out that not all actors can "reach" all other actors. Here's the output of *Network>Cohesion>Reachability* from UCINET.



```
Matrix #1


                                   1
        1  2  3  4  5  6  7  8  9  0
        C  C  E  I  M  W  N  U  W  W
        -  -  -  -  -  -  -  -  -  -
   1    0  1  1  1  1  1  1  1  1  1
   2    1  0  1  1  1  1  1  1  1  1
   3    1  1  0  1  1  1  1  1  1  1
   4    1  1  1  0  1  1  1  1  1  1
   5    1  1  1  1  0  1  1  1  1  1
   6    1  1  1  1  1  0  1  1  1  1
   7    1  1  1  1  1  1  0  1  1  1
   8    1  1  1  1  1  1  1  0  1  1
   9    1  1  1  1  1  1  1  1  0  1
  10    1  1  1  1  1  1  1  1  1  0

Matrix #2


                                   1
        1  2  3  4  5  6  7  8  9  0
        C  C  E  I  M  W  N  U  W  W
        -  -  -  -  -  -  -  -  -  -
   1    0  1  1  0  1  0  0  1  1  1
   2    0  0  1  0  0  0  0  1  1  1
   3    0  0  0  0  0  0  0  1  1  1
   4    0  1  1  0  0  0  1  1  1  1
   5    0  1  1  0  0  0  0  1  1  1
   6    0  0  0  0  0  0  0  0  0  0
   7    0  1  1  0  0  0  0  1  1  1
   8    0  0  1  0  0  0  0  0  1  1
   9    0  0  1  0  0  0  0  1  0  1
  10    0  0  0  0  0  0  0  0  0  0
```

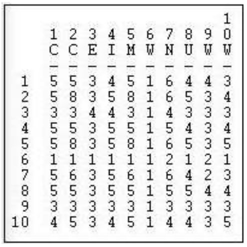
Figure 7.8: *Reachability of Knoke "I" and "M" relations*

So, there exists a directed "path" from each organization to each other actor for the flow of information, but not for the flow of money. Sometimes "what goes around comes around", and sometimes it doesn't!

Connectivity

Adjacency tells us whether there is a direct connection from one actor to another (or between two actors for undirected data). Reachability tells us whether two actors are connected or not by way of either a direct or an indirect pathway of any length.

Network>Cohesion>Point Connectivity calculates the number of nodes that would have to be removed in order for one actor to no longer be able to reach another. If there are many different pathways that connect two actors, they have high "connectivity" in the sense that there are multiple ways for a signal to reach from one to the other. Figure 7.9 shows the point connectivity for the flow of information among the 10 Knoke organizations.



```
                                   1
        1  2  3  4  5  6  7  8  9  0
        C  C  E  I  M  W  N  U  W  W
        -  -  -  -  -  -  -  -  -  -
   1    5  5  3  4  5  1  6  4  4  3
   2    5  8  3  5  8  1  6  5  3  4
   3    3  3  4  4  3  1  4  3  3  3
   4    5  5  3  5  5  1  5  4  3  4
   5    5  8  3  5  8  1  6  5  3  5
   6    1  1  1  1  1  1  2  1  2  1
   7    5  6  3  5  6  1  6  4  2  3
   8    5  5  3  5  5  1  5  5  4  4
   9    3  3  3  3  3  1  3  3  3  3
  10    4  5  3  4  5  1  4  4  3  5
```

Figure 7.9: *Point connectivity of Knoke information exchange*

The result again demonstrates the tenuousness of organization 6's connection as both a source (row) or receiver (column) of information. To get its message to most other actors, organization 6 has alternative; should a single organization refuse to pass along information, organization 6 would receive none at all! Point connectivity can be a useful measure to get at notions of dependency and vulnerability.

7.3: Distance

The properties of the network that we have examined so far primarily deal with adjacencies - the direct connections from one actor to the next. But the way that people are embedded in networks is more complex than this. Two persons, call them A and B, might each have five friends. But suppose that none of person A's friends have any friends except A. Person B's five friends, in contrast, each have five friends. The information available to B and B's potential for influence is far greater than A's. That is, sometimes being a "friend of a friend" may be quite consequential.

To capture this aspect of how individuals are embedded in networks, one main approach is to examine the distance that an actor is from others. If two actors are adjacent, the distance between them is one (that is, it takes one step for a signal to go from the source to the receiver). If A tells B, and B tells C (and A does not tell C), then actors A and C are at a distance of two. How many actors are at various distances from each actor can be important for understanding the differences among actors in the constraints and opportunities they have as a result of their position. Sometimes we are also interested in how many ways there are to connect between two actors, at a given distance. That is, can actor A reach actor B in more than one way? Sometimes multiple connections may indicate a stronger connection between two actors than a single connection.

The distances among actors in a network may be an important macro-characteristic of the network as a whole. Where distances are great, it may take a long time for information to diffuse across a population. It may also be that some actors are quite unaware of, and influenced by others - even if they are technically reachable, the costs may be too high to conduct exchanges. The variability across the actors in the distances that they have from other actors may be a basis for differentiation and even stratification. Those actors who are closer to more others may be able to exert more power than those who are more distant. We will have a good deal more to say about this aspect of variability in actor distances in the next chapter.

For the moment, we need to learn a bit of jargon that is used to describe the distances between actors: *walks, paths, semi-paths,* etc. Using these basic definitions, we can then develop some more powerful ways of describing various aspects of the distances among actors in a network.

Walks etc.

To describe the distances between actors in a network with precision, we need some terminology. And, as it turns out, whether we are talking about a simple graph or a directed graph makes a good bit of difference. If A and B are adjacent in a simple graph, they have a distance of one. In a direct graph, however, A can be adjacent to B while B is not adjacent to A - the distance from A to B is one, but there is no distance from B to A. Because of this difference, we need slightly different terms to describe distances between actors in graphs and digraphs.

Simple graphs: The most general form of connection between two actors in a graph is called a *walk*. A walk is a sequence of actors and relations that begins and ends with actors. A *closed walk* is one where the beginning and end point of the walk are the same actor. Walks are unrestricted. A walk can involve the same actor or the same relation multiple times. A *cycle* is a specially restricted walk that is often used in algorithms examining the neighborhoods (the points adjacent) of actors. A cycle is a closed walk of 3 or more actors, all of whom are distinct, except for the origin/destination actor. The length of a walk is simply the number of relations contained in it. For example, consider this graph in Figure 7.10.

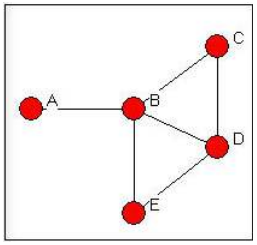


Figure 7.10: *Walks in a simple graph*

There are many walks in a graph (actually, an infinite number if we are willing to include walks of any length - though, usually, we restrict our attention to fairly small lengths). To illustrate just a few, begin at actor A and go to actor C. There is one walk of length 2 (A,B,C). There is one walk of length three (A,B,D,C). There are several walks of length four (A,B,E,D,C; A,B,D,B,C; A,B,E,B,C). Because these are unrestricted, the same actors and relations can be used more than once in a given walk. There are no cycles beginning and ending with A. There are some beginning and ending with actor B (B,D,C,B; B,E,D,B; B,C,D,E,B).

It is usually more useful to restrict our notion of what constitutes a connection somewhat. One possibility is to restrict the count only walks that do not re-use relations. A *trail* between two actors is any walk that includes a given relation no more than once (the same other actors, however, can be part of a trail multiple times. The length of a trail is the number of relations in it. All trails are walks, but not all walks are trails. If the trail begins and ends with the same actor, it is called a *closed trail*. In our example above, there are a number of trails from A to C. Excluded are tracings like A,B,D,B,C (which is a walk, but is not a trail because the relation BD is used more than once).

Perhaps the most useful definition of a connection between two actors (or between an actor and themselves) is a *path*. A path is a walk in which each other actor and each other relation in the graph may be used at most one time. The single exception to this is a *closed path*, which begins and ends with the same actor. All paths are trails and walks, but all walks and all trails are not paths. In our example, there are a limited number of paths connecting A and C: A,B,C; A,B,D,C; A,B,E,D,C.

Directed graphs: Walks, trails, and paths can also be defined for directed graphs. But there are two flavors of each, depending on whether we want to take direction into account or not. *Semi-walks, semi-trails,* and *semi-paths* are the same as for undirected data. In defining these distances, the directionality of connections is simply ignored (that is, arcs - or directed ties are treated as though they were edges - undirected ties). As always, the length of these distances is the number of relations in the walk, trail, or path.

If we do want to pay attention to the directionality of the connections we can define *walks*, *trails*, and *paths* in the same way as before, but with the restriction that we may not "change direction" as we move across relations from actor to actor. Consider the directed graph in Figure 7.11.

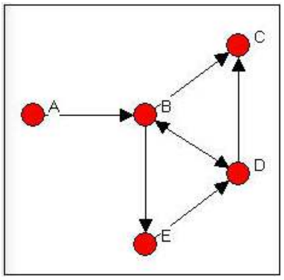


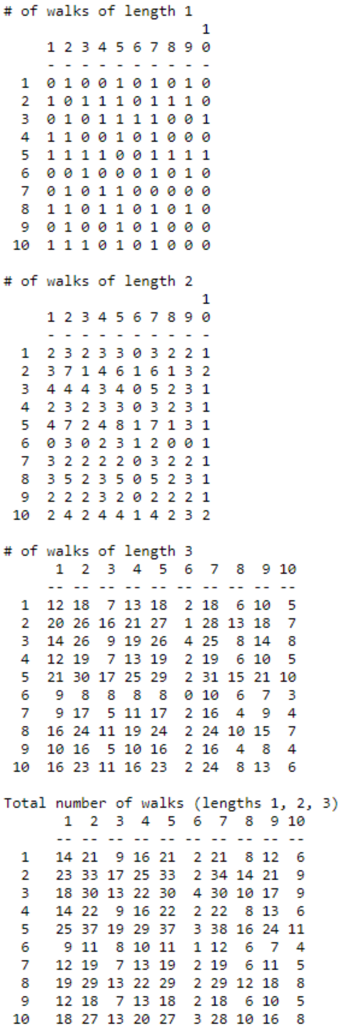
Figure 7.11: *Walks in a directed graph*

In this directed graph, there are a number of walks from A to C. However, there are no walks from C (or anywhere else) to A. Some of these walks from A to C are also trails (e.g. A,B,E,D,B,C). There are, however, only three paths from A to C. One path is length 2 (A,B,C); one is length three (A,B,D,C); one is length four (A,B,E,D,C).

The various kinds of connections (walks, trails, paths) provide use with a number of different ways of thinking about the distances between actors. The main reason that social network analysts are concerned with these distances is that they provide a way of thinking about the strength of ties or relations. Actors that are connected at short lengths or distances may have stronger ties. Their connection may also be less subject to disruption, and hence more stable and reliable.

The numbers of walks of a given length between all pairs of actors can be found by raising the matrix to that power. A convenient method for accomplishing this is to use *Tools>Matrix Algebra*, and to specify an expression like *out=prod(X1,X1)*. This produces the square of the matrix X1, and stores it as the dataset "out". A more detailed discussion of this idea can be found in the earlier chapter on representing networks as matrices. This matrix could then be added to X1 to show the number of walks between any two actors of length two or less.

Let's look briefly at the distances between pairs of actors in the Knoke data on directed information flows. Counts of the numbers of paths of various lengths are shown in Figure 7.12.



```
# of walks of length 1
                       1
     1 2 3 4 5 6 7 8 9 0
     - - - - - - - - - -
  1  0 1 0 0 1 0 1 0 1 0
  2  1 0 1 1 1 0 1 1 1 0
  3  0 1 0 1 1 1 1 0 0 1
  4  1 1 0 0 1 0 1 0 0 0
  5  1 1 1 1 0 0 1 1 1 1
  6  0 0 1 0 0 0 1 0 1 0
  7  0 1 0 1 1 0 0 0 0 0
  8  1 1 0 1 1 0 1 0 1 0
  9  0 1 0 0 1 0 1 0 0 0
 10  1 1 1 0 1 0 1 0 0 0

# of walks of length 2
                       1
     1 2 3 4 5 6 7 8 9 0
     - - - - - - - - - -
  1  2 3 2 3 3 0 3 2 2 1
  2  3 7 1 4 6 1 6 1 3 2
  3  4 4 4 3 4 0 5 2 3 1
  4  2 3 2 3 3 0 3 2 3 1
  5  4 7 2 4 8 1 7 1 3 1
  6  0 3 0 2 3 1 2 0 0 1
  7  3 2 2 2 2 0 3 2 2 1
  8  3 5 2 3 5 0 5 2 3 1
  9  2 2 2 3 2 0 2 2 2 1
 10  2 4 2 4 4 1 4 2 3 2

# of walks of length 3
      1  2  3  4  5  6  7  8  9 10
     -- -- -- -- -- -- -- -- -- --
  1  12 18  7 13 18  2 18  6 10  5
  2  20 26 16 21 27  1 28 13 18  7
  3  14 26  9 19 26  4 25  8 14  8
  4  12 19  7 13 19  2 19  6 10  5
  5  21 30 17 25 29  2 31 15 21 10
  6   9  8  8  8  8  0 10  6  7  3
  7   9 17  5 11 17  2 16  4  9  4
  8  16 24 11 19 24  2 24 10 15  7
  9  10 16  5 10 16  2 16  4  8  4
 10  16 23 11 16 23  2 24  8 13  6

Total number of walks (lengths 1, 2, 3)
       1  2  3  4  5  6  7  8  9 10
      -- -- -- -- -- -- -- -- -- --
  1   14 21  9 16 21  2 21  8 12  6
  2   23 33 17 25 33  2 34 14 21  9
  3   18 30 13 22 30  4 30 10 17  9
  4   14 22  9 16 22  2 22  8 13  6
  5   25 37 19 29 37  3 38 16 24 11
  6    9 11  8 10 11  1 12  6  7  4
  7   12 19  7 13 19  2 19  6 11  5
  8   19 29 13 22 29  2 29 12 18  8
  9   12 18  7 13 18  2 18  6 10  5
 10   18 27 13 20 27  3 28 10 16  8
```

Figure 7.12: *Numbers of walks in Knoke information network*

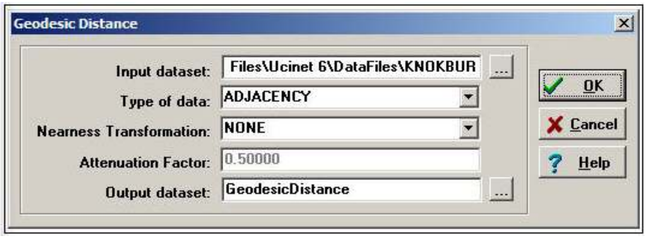
The inventory of the total connections among actors is primarily useful for getting a sense of how "close" each pair is, and for getting a sense of how closely coupled the entire system is. Here, we can see that using only connections of two steps (e.g. "A friend of a friend"), there is a great deal of connection in the graph overall; we also see that there are sharp differences among actors in their degree of connectedness, and who they are connected to. These differences can be used to understand how information moves in the network, which actors are likely to be influential on one another, and a number of other important properties.

Geodesic Distance, Eccentricity, and Diameter

One particular definition of the distance between actors in a network is used by most algorithms to define more complex properties of individual's positions and the structure of the network as a whole. This quantity is the *geodesic distance*. For both directed and undirected data, the geodesic distance is the number of relations in the shortest possible walk from one actor to another (or, from an actor to themselves, if we care, which we usually do not).

The geodesic distance is widely used in network analysis. There may be connections between two actors in a network. If we consider how the relation between two actors may provide each with opportunity and constraint, it may well be the case that not all of these ties matter. For example, suppose that I am trying to send a message to Sue: Since I know her email address, I can send it directly (a path of length 1). I also know Donna, and I know that Donna has Sue's email address. I could send my message for Sue to Donna, and ask her to forward it. This would be a path of length two. Confronted with this choice, I am likely to choose the geodesic path (i.e. directly to Sue) because it is less trouble and faster, and because it does not depend on Donna. That is, the geodesic path (or paths, as there can be more than one) is often the "optimal" or most "efficient" connection between two actors. Many algorithms in network analysis assume that actors will use the geodesic path when alternatives are available.

Using UCINET, we can easily locate the lengths of the geodesic paths in our directed data on information exchanges. Here is the dialog box for *Network>Cohesion>Distance*.



Geodesic Distance

Input dataset: Files\Ucinet 6\DataFiles\KNOKBUR

Type of data: ADJACENCY

Nearness Transformation: NONE

Attenuation Factor: 0.50000

Output dataset: GeodesicDistance

Figure 7.13: *Network>Cohesion>Distance dialog*

The Knoke information exchange data are binary (organization A sends information to organization B, or it doesn't). That is, the pattern is summarized by an adjacency matrix. For binary data, the geodesic distance between two actors is the count of the number of links in the shortest path between them.

It is also possible to define the distance between two actors where the links are valued. That is, where we have a measure of the strength of ties, the opportunity costs of ties, or the probability of a tie. *Network>Cohesion>Distance* can calculate distance (and nearness) for valued data, as well (select the appropriate "type of data").

Where we have measures of the strengths of ties (e.g. the dollar volume of trade between two nations), the "distance" between two actors is defined as the strength of the weakest path between them. If A sends 6 units to B, and B sends 4 units to C, the "strength" of the path from A to C (assuming A to B to C is the shortest path) is 4.

Where we have a measure of the probability that a link will be used, the "distance" between two actors is defined as the product along the pathway - as in path analysis in statistics.

The *Nearness Transformation* and *Attenuation Factor* parts of the dialog allow the rescaling ot distances into nearnesses. For many analyses, we may be interested in thinking about the connections among actors in terms of how close or similar they are, rather than how distant. There are a number of ways that this may be done.

The *multiplicative* nearness transformation divides the distance by the largest possible distance between two actors. For example, if we had 7 nodes, the maximum possible distance for adjacency data would be 6. This method gives a measure of the distance as a percentage of the theoretical maximum for a given graph.

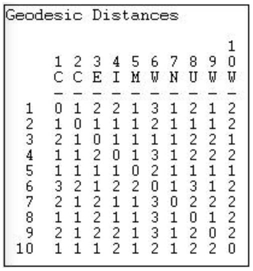
The *additive* nearness transformation subtracts the actual distance between two actors from the number of nodes. It is similar to the multiplicative scaling, but yields a value as the nearness measure, rather than a proportion.

The *linear* nearness transformation rescales distance by reversing the scale (i.e. the closest becomes the most distant, the most distant becomes the nearest) and re-scoring to make the scale range from zero (closest pair of nodes) to one (most distant pair of nodes).

The *exponential decay* method turns distance into nearness by weighting the links in the pathway with decreasing values as they fall farther away from ego. With an *attenuation factor* of 0.5, for example, a path from A to B to C would result in a distance of 1.5.

The *frequency decay* method is defined as 1 minus the proportion of other actors who are as close or closer to the target as ego is. The idea (Ron Burt's) is that if there are many other actors closer to the target you are trying to reach than yourself, you are effectively "more distant".

In our example, we are using simple directed adjacencies, and the results (Figure 7.14) are quite straightforward.



```
Geodesic Distances

                                     1
        1  2  3  4  5  6  7  8  9  0
        C  C  E  I  M  W  N  U  W  W
        -  -  -  -  -  -  -  -  -  -
   1    0  1  2  2  1  3  1  2  1  2
   2    1  0  1  1  1  2  1  1  1  2
   3    2  1  0  1  1  1  1  2  2  1
   4    1  1  2  0  1  3  1  2  2  2
   5    1  1  1  1  0  2  1  1  1  1
   6    3  2  1  2  2  0  1  3  1  2
   7    2  1  2  1  1  3  0  2  2  2
   8    1  1  2  1  1  3  1  0  1  2
   9    2  1  2  2  1  3  1  2  0  2
  10    1  1  1  2  1  2  1  2  2  0
```

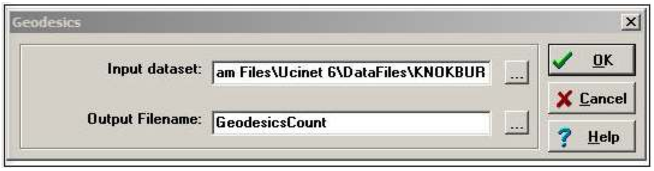
Figure 7.14: *Geodesic distances for Knoke information exchange*

Because the network is moderately dense, the geodesic distances are generally small. This suggests that information may travel pretty quickly in this network. Also note that there is a geodesic distance for each x,y and y,x pair - that is, the graph is fully connected, and all actors are "reachable" from all others (that is, there exists a path of some length from each actor to each other actor). When a network is not fully connected, we cannot exactly define the geodesic distances among all pairs. The standard approach in such cases is to treat the geodesic distance between unconnected actors as a length greater than that of any real distance in the data. For each actor, we could calculate the mean and standard deviation of their geodesic distances to describe their closeness to all other actors. For each actor, that actor's largest geodesic distance is called the *eccentricity* - a measure of how far an actor is from the furthest other.

Because the current network is fully connected, a message that starts anywhere will eventually reach everyone. Although the computer has not calculated it, we might want to calculate the mean (or median) geodesic distance, and the standard deviation in geodesic distances for the matrix, and for each actor row-wise and column-wise. This would tell us how far each actor is from each other as a source of information for the other; and how far each actor is from each other who may be trying to influence them. It also tells us which actors' behavior (in this case, whether they've heard something or not) is most predictable and least predictable.

In looking at the whole network, we see that it is connected, and that the average geodesic distance among actors is quite small. This suggests a system in which information is likely to reach everyone, and to do so fairly quickly. To get another notion of the size of a network, we might think about its diameter. The *diameter of a network* is the largest possible geodesic distance in the (connected) network. In the current case, no actor is more than three steps from any other - a very "compact" network. The diameter of a network tells us how "big" it is, in one sense (that is, how many steps are necessary to get from one side of it to the other). The diameter is also a useful quantity in that it can be used to set an upper bound on the lengths of connections that we study. Many researchers limit their explorations of the connections among actors to involve connections that are no longer than the diameter of the network.

Sometimes the redundancy of connection is an important feature of a network structure. If there are many efficient paths connecting two actors, the odds are improved that a signal will get from one to the other. One index of this is a count of the number of geodesic paths between each pair of actors. Of course, if two actors are adjacent, there can only be one such path. The number of geodesic paths can be calculated with *Network>Cohesion>No. of Geodesics*, as in Figure 7.15.

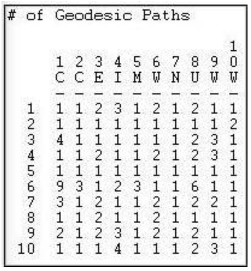


Input dataset: am Files\Ucinet 6\DataFiles\KNOKBUR

Output Filename: GeodesicsCount

Figure 7.15: *Dialog for Network>Cohesion>No. of Geodesics*

The results are shown in Figure 7.16.



```
# of Geodesic Paths

                              1
       1  2  3  4  5  6  7  8  9  0
       C  C  E  I  M  W  N  U  W  W
       -  -  -  -  -  -  -  -  -  -
  1    1  1  2  3  1  2  1  2  1  1
  2    1  1  1  1  1  1  1  1  1  2
  3    4  1  1  1  1  1  1  2  3  1
  4    1  1  2  1  1  2  1  2  3  1
  5    1  1  1  1  1  1  1  1  1  1
  6    9  3  1  2  3  1  1  6  1  1
  7    3  1  2  1  1  2  1  2  2  1
  8    1  1  2  1  1  2  1  1  1  1
  9    2  1  2  3  1  2  1  2  1  1
 10    1  1  1  4  1  1  1  2  3  1
```

Figure 7.16: *Number of geodesic paths for Knoke information exchange*

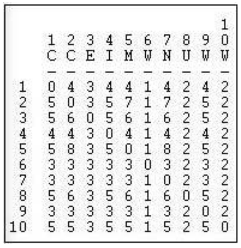
We see that most of the geodesic connections among these actors are not only short distance, but that there are very often multiple shortest paths from x to y. This suggests a couple things: information flow is not likely to break down, because there are multiple paths; and, it will be difficult for any individual to be a powerful "broker" in this structure because most actors have alternative efficient ways of connection to other actors that can bypass any given actor.

Flow

The use of geodesic paths to examine properties of the distances between individuals and for the whole network often makes a great deal of sense. But, there may be other cases where the distance between two actors, and the connectedness of the graph as a whole is best though of as involving all connections - not just the most efficient ones. If I start a rumor, for example, it will pass through a network by all pathways - not just the most efficient ones. How much credence another person gives my rumor may depend on how many times they hear it from different sources - and not how soon they hear it. For uses of distance like this, we need to take into account all of the connections among actors.

Several approaches have been developed for counting the amount of connection between pairs of actors that take into account all connections between them. These measures have been used for a number of different purposes, and these differences are reflected in the algorithms used to calculate them.

Network>Cohesion>Maximum Flow. One notion of how totally connected two actors are (called maximum flow by UCINET) asks how many different actors in the neighborhood of a source lead to pathways to a target. If I need to get a message to you, and there is only one other person to whom I can send this for retransmission, my connection is weak -even if the person I send it to may have many ways of reaching you. If, on the other hand, there are four people to whom I can send my message, each of whom has one or more ways of retransmitting my message to you, then my connection is stronger. The "flow" approach suggests that the strength of my tie to you is no stronger than the weakest link in the chain of connections, where weakness means a lack of alternatives. This approach to connection between actors is closely connected to the notion of betweenness that we will examine a bit later. It is also logically close to the idea that the number of pathways, not their length, may be important in connecting people. For our directed information flow data, the results of UCINET's count of maximum flow are shown in Figure 7.17.



```
                              1
       1  2  3  4  5  6  7  8  9  0
       C  C  E  I  M  W  N  U  W  W
       -  -  -  -  -  -  -  -  -  -
  1    0  4  3  4  4  1  4  2  4  2
  2    5  0  3  5  7  1  7  2  5  2
  3    5  6  0  5  6  1  6  2  5  2
  4    4  4  3  0  4  1  4  2  4  2
  5    5  8  3  5  0  1  8  2  5  2
  6    3  3  3  3  3  0  3  2  3  2
  7    3  3  3  3  3  1  0  2  3  2
  8    5  6  3  5  6  1  6  0  5  2
  9    3  3  3  3  3  1  3  2  0  2
 10    5  5  3  5  5  1  5  2  5  0
```

Figure 7.17: *Maximum flow for Knoke information network*

You should verify for yourself that, for example, there are four intermediaries, or alternative routes, in flows from actor 1 to actor 2, but five such points in the flow from actor 2 to actor 1. The higher the number of flows from one actor to another, the greater the likelihood that communication will occur, and the less "vulnerable" the connection. Note that actors 6, 7, and 9 are relatively disadvantaged. In particular, actor 6 has only one way of obtaining information from all other actors (the column vector of flows to actor 6).

7.E: Connection (Exercises)

Review Questions

1. Explain the differences among the "three levels of analysis" of graphs (individual, aggregate, whole).
2. How is the size of a network measured? Why is population size so important in sociological analysis?
3. You have a network of 5 actors. Assuming no self-ties, what is the potential number of directed ties? What is the potential number of undirected ties?
4. How is density measured? Why is density important in sociological analysis?
5. What is the "degree of a point"? Why might it be important, sociologically, if some actors have high degree and other actors have lower degree? What is the difference between "in-degree" and "out-degree"?
6. If actor "A" is reachable from actor "B" does that necessarily mean that actor "B" is reachable from actor "A"? Why or why not?
7. For pairs of actors with directed relations, there are four possible configurations of ties. Can you show these? Which configurations are "balanced"? For a triad with undirected relations, how many possible configurations of ties are there? Which ones are balanced or transitive?
8. What are the differences among walks, trails, and paths? Why are "paths" the most commonly used approach to inter-actor distances in sociological analysis?
9. What is the "geodesic"distance between two actors? Many social network measures assume that the geodesic path is the most important path between actors - why is this a plausible assumption?
10. I have two populations of ten actors each, one has a network diameter of 3, the other has a network diameter of 6. Can you explain this statement to someone who doesn't know social network analysis? Can you explain why this difference in diameter might be important in understanding differences between the two populations?
11. How do "weighted flow" approaches to social distance differ from "geodesic" approaches to social distance?
12. Why might it matter if two actors have more than one geodesic or other path between them?

Application Questions

1. Think of the readings from the first part of the course. Which studies used the ideas of connectedness and density? Which studies used the ideas of distance? What specific approaches did they use to measure these concepts?
2. Draw the graphs of a "star", a "circle", a "line", and a "hierarchy". Describe the size, potential, and density of each graph. Examine the degrees of points in each graph - are there differences among actors? Do these differences tell us something about the "social roles" of the actors? Create a matrix for each graph that shows the geodesic distances between each pair of actors. Are there differences between the graphs in whether actors are connected by multiple geodesic distances?
3. Think about a small group of people that you know well (maybe your family, neighbors, a study group, etc.). Who helps whom in this group? What is the density of the ties? Are ties reciprocated? Are triads transitive?
4. Chrysler Corporation has called on you to be a consultant. Their research division is taking too long to generate new models of cars, often the work of the "stylists" doesn't fit well with the work of the "manufacturing engineers" (the people who figure out how to actually build the car). Chrysler's research division is organized as a classical hierarchy bureaucracy with two branches (stylists, manufacturing) coordinated through group managers and a division manager. Analyze the reasons why performance is poor. Suggest some alternative ways of organizing that might improve performance, and explain why they will help.

7.S: Connection (Summary)

There is a great deal of information about both individuals and the population in a single adjacency matrix. In this chapter you have learned a lot of terminology for describing the connections and distances between actors, and for whole populations.

One focus in basic network analysis is on the immediate neighborhood of each actor: the dyads and triads in which they are involved. The degree of an actor, and the in-degree and out-degree (if the data are directed) tell us about the extent to which an actor may be constrained by, or constrain others. The extent to which an actor can reach others in the network may be useful in describing an actor's opportunity structure. We have also seen that it is possible to describe "types" of actors who may form groups or strata on the basis of their places in opportunity structure - e.g. "isolates", "sources", etc.

Most of the time and effort of most social actors is spent in very local contexts - interacting in dyads and triads. In looking at the connections of actors, we have suggested that the degree of "reciprocity" and "balance" and "transitivity" in relations can be regarded as important indictors of the stability and institutionalization (that is, the extent to which relations are taken for granted and are norm governed) of actor's positions in social networks.

The local connections of actors are important for understanding the social behavior of the whole population, as well as for understanding each individual. The size of the network, its density, whether all actors are reachable by all others (i.e. is the whole population connected, or are there multiple components?), whether ties tend to be reciprocal or transitive, and all the other properties that we examined for individual connections are meaningful in describing the whole population. Both the typical levels of characteristics (e.g. the mean degree of points), and the amount of diversity in characteristics (e.g. the variance in the degree of points) may be important in explaining macro behavior. Populations with high density respond differently to challenges from the environment than those with low density; populations with greater diversity in individual densities may be more likely to develop stable social differentiation and stratification.

In this chapter we also examined some properties of individual's embeddedness and of whole networks that look at the broader, rather than the local, neighborhoods of actors. A set of specialized terminology was introduced to describe the distances between pairs of actors: walks, trails, and paths. We noted that there are some important differences between undirected and directed data in applying these ideas of distance.

One of the most common and important approaches to indexing the distances between actors is the geodesic. The geodesic is useful for describing the minimum distance between actors. The geodesic distances between pairs of actors is the most commonly used measure of closeness. The average geodesic distance for an actor to all others, the variation in these distances, and the number of geodesic distances to other actors may all describe important similarities and differences between actors in how, and how closely, they are connected to their entire population.

The geodesic distance, however, examines only a single connection between a pair of actors (or, in some cases several, if there are multiple geodesics connecting them). Sometimes the sum of all connections between actors, rather than the shortest connection may be relevant. We have examined approaches to measuring the vulnerability of the connection between actors by looking at the number of geodesic connections between pairs of actors, and the total number of pathways between pairs of actors.

We have seen that there is a great deal of information available in fairly simple examinations of an adjacency matrix. Life, of course, can get more complicated. We could have multiple layers, or multiplex data; we could have data that gave information on the strength of ties, rather than simple presence or absence. Nonetheless, the methods that we've used here will usually give you a pretty good grasp of what is going on in more complicated data.

Now that you have a pretty good grasp of the basics of connection and distance, you are ready to use these ideas to build some concepts and methods for describing somewhat more complicated aspects of the network structures of populations. In the next two chapters, we will focus on ways of examining the local neighborhoods of actors. In Chapter 8, we will look at methods for summarizing the entire graph in terms of the kinds of connections that individuals have to their neighbors. In Chapter 9, we'll examine actors' local neighborhoods from their own individual perspective.

CHAPTER OVERVIEW

8: Embedding

Topic hierarchy

8.1: Introduction

In the previous chapter we looked at some tools for examining the ways that individuals are connected, and the distances between them. In this chapter we will look at the same issue of connection. This time, though, our focus is the social structure, rather than the individual. That is, we will adopt a more "macro" perspective that focuses on the structures within which individual actors are embedded.

The "top down" perspective we'll follow in this chapter seeks to understand and describe whole populations by the "texture" of the relations that constrain its individual members. Imagine one society in which extended kin groups live in separate villages at considerable distances from one another. Most "texture" of the society will be one in which individuals have strong ties to relatively small numbers of others in local "clusters". Compare this to a society where a large portion of the population lives in a single large city. Here, the "texture" of social relations is quite different - individuals may be embedded in smaller nuclear families of mating relations, but have diverse ties to neighbors, friends, coworkers, and others.

Social network analysts have developed a number of tools for conceptualizing and indexing the variations in the kinds of structures that characterize populations. In this chapter, we'll examine a few of these tools.

The smallest social structure in which an individual can be embedded is a dyad (that is, a pair of actors). For binary ties (present or absent), there are two possibilities for each pair in the population - either they have a tie, or they don't. We can characterize the whole population in terms of the prevalence of these dyadic "structures". This is what the density measure does.

If we are considering a directed relation (A might like B, but B might not like A), there are three kinds of dyands (no tie, one likes the other but not vice versa, or both like the other). The extent to which a population is characterized by "reciprocated" ties (those where each directs a tie to the other) may tell us about the degree of cohesion, trust, and social capital that is present.

The smallest social structure that has the true character of a "society" is the triad - any "triple" {A, B, C} of actors. Such a structure "embeds" dyadic relations in a structure where "other" is present along with "ego" and "alter". The analysis of triads, and the prevalence of different types of triads in populations has been a staple of sociometry and social network analysis. In (directed) triads, we can see the emergence of tendencies toward equilibrium and consistency - institutionalization - of social structures (balance and transitivity). Triads are also the simplest structures in which we can see the emergence of hierarchy.

Most of the time, most people interact with a fairly small set of others, many of whom know one another. The extent of local "clustering" in populations can be quite informative about the texture of everyday life. Actors are also embedded in "categorical social units" or "sub-populations" defined either by shared attributes or shared membership. The extent to which these sub-populations are open or closed - the extent to which most individuals have most of their ties within the boundaries of these groups - may be a telling dimension of social structure.

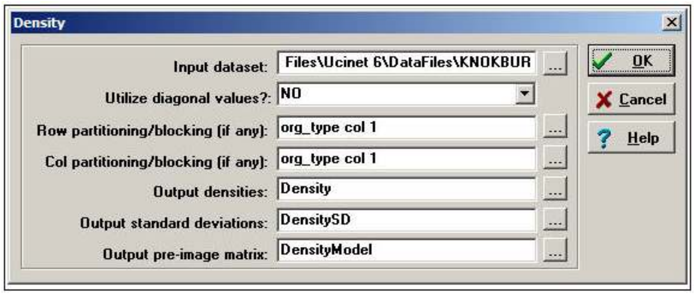
There are many approaches to characterizing the extent and form of "embedding" of actors in populations. There is no one "right" way of indexing the degree of embedding in a population that will be effective for all analytic purposes. There are, however, some very interesting and often useful approaches that you may wish to explore.

8.2: Density

If we are comparing two populations, and we note that there are many actors in one that are not connected to any other ("isolates"), and in the other population most actors are embedded in at least one dyad - we would likely conclude that social life is very different in the two populations.

Measuring the density of a network gives use a ready index of the degree of dyadic connection in a population. For binary data, density is simply the ratio of the number of adjacencies that are present divided by the number of pairs - what proportion of all possible dyadic connections are actually present. If we have measured the ties among actors with values (strengths, closeness, probabilities, etc.) density is usually defined as the sum of the values of all ties divided by the number of possible ties. That is, with valued data, density is usually defined as the average strength of ties across all possible (not all actual) ties. Where the data are symmetric or undirected, density is calculated relative to the number of unique pairs $((n*n-1)/2)$; where the data are directed, density is calculated across the total number of pairs.

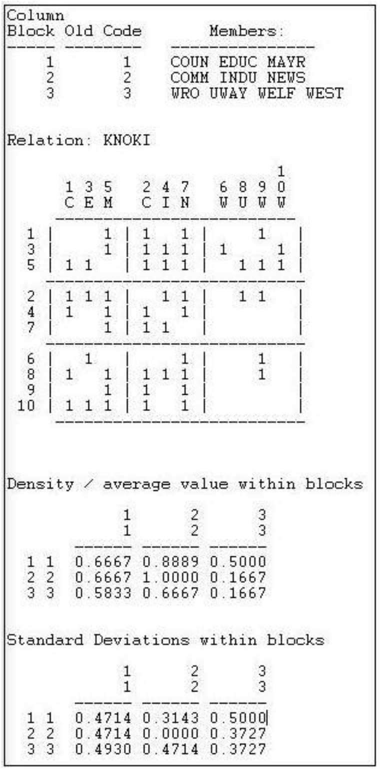
Network>Cohesion>Density is a useful tool for calculating the density of whole populations, or of partitions. A typical dialog is shown in Figure 8.1.



Input dataset: Files\Ucinet 6\DataFiles\KNOKBUR
Utilize diagonal values?: NO
Row partitioning/blocking (if any): org_type col 1
Col partitioning/blocking (if any): org_type col 1
Output densities: Density
Output standard deviations: DensitySD
Output pre-image matrix: DensityModel

Figure 8.1: *Dialog of Network>Cohesion>Density*

In this dialog, we are again examining the Knoke information tie network. We have used an attribute or partition to divide the cases into three sub-populations (governmental agencies, non-governmental generalist, and welfare specialists) so that we can see the amount of connection within and between groups. This is done by creating a separate attribute data file (or a column in such a file), with the same row labels, and scores for each case on the "partitioning" variable. Partitioning is not necessary to calculate density. The results of the analysis are shown in Figure 8.2.



```
Column
Block Old Code      Members:
------ ---------    ---------------
    1       1    COUN EDUC MAYR
    2       2    COMM INDU NEWS
    3       3    WRO UWAY WELF WEST

Relation: KNOKI

                            1
      1 3 5   2 4 7   6 8 9 0
      C E M   C I N   W U W W
     --------------------------
  1 |     1 | 1   1 |     1   |
  3 |     1 | 1 1 1 | 1     1 |
  5 | 1 1   | 1 1 1 |   1 1 1 |
    ---------------------------
  2 | 1 1 1 |   1 1 |   1 1   |
  4 | 1   1 | 1   1 |         |
  7 |     1 | 1 1   |         |
    ---------------------------
  6 |   1   |     1 |     1   |
  8 | 1   1 | 1 1 1 |     1   |
  9 |     1 | 1   1 |         |
 10 | 1 1 1 | 1   1 |         |
     --------------------------


Density / average value within blocks

            1      2      3
            1      2      3
       ------ ------ ------
  1 1  0.6667 0.8889 0.5000
  2 2  0.6667 1.0000 0.1667
  3 3  0.5833 0.6667 0.1667

Standard Deviations within blocks

            1      2      3
            1      2      3
       ------ ------ ------
  1 1  0.4714 0.3143 0.5000
  2 2  0.4714 0.0000 0.3727
  3 3  0.4930 0.4714 0.3727
```

Figure 8.2: *Density of three sub-populations in Knoke information network*

After providing a map of the partitioning, a blocked (partitioned) matrix is provided showing the values of the connections between each pair of actors. Next, the within-block densities are presented. The density in the 1,1 block is 0.6667. That is, of the six possible directed ties among actors 1, 3, and 5, four are actually present (we have ignored the diagonal - which is the most common approach). We can see that the three sub-populations appear to have some differences. Governmental agencies (block 1) have quite dense in and out ties to one another, and to the other populations; non-governmental generalists (block 2) have out-ties among themselves and with block 1, and have high densities of in-ties with all three sub-populations. The welfare specialists have high density of information sending to the other two blocks (but not within their block), and receive more input from governmental than from non-governmental organizations.

The extent to which these simple characterizations of blocks characterize all the individuals within those blocks - essentially the validity of the blocking - can be assessed by looking at the standard deviations within the partitions. The standard deviations measure the lack of homogeneity within the partition, or the extent to which the actors vary.

A social structure in which individuals were highly clustered would display a pattern of high densities on the diagonal, and low densities elsewhere.

8.3: Reciprocity

With symmetric dyadic data, two actors are either connected, or they are not. Density tells us pretty much all there is to know.

With directed data, there are four possible dyadic relationships: A and B are not connected, A sends to B, B sends to A, or A and B send to each other. A common interest in looking at directed dyadic relationships is the extent to which ties are reciprocated. Some theorists fell that there is an equilibrium tendency toward dyadic relationships to be either null or reciprocated, and that asymmetric ties may be unstable. A network that has a predominance of null or reciprocated ties over asymmetric connections may be a more "equal" or "stable" network than one with a predominance of asymmetric connections (which might be more of a hierarchy).

There are (at least) two different approaches to indexing the degree of reciprocity in a population. Consider the very simple network shown in Figure 8.3. Actors A and B have reciprocated ties, actors B and C have a non-reciprocated tie, and actors A and C have no tie.

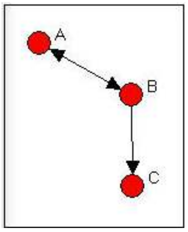
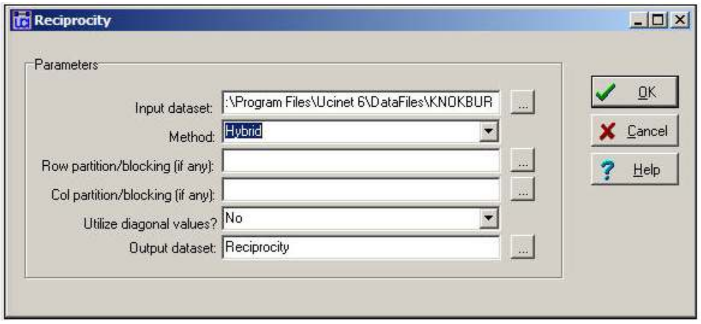


Figure 8.3: *Definitions of reciprocity*

What is the prevalence of reciprocity in this network? One approach is to focus on the dyads, and ask what proportion of pairs have a reciprocated tie between them? This would yield one such tie for three possible pairs (AB, AC, BC), or a reciprocity rate of 0.333. More commonly, analysts are concerned with the ratio of the number of pairs with a reciprocated tie relative to the number of pairs with any tie. In large populations, usually most actors have no direct ties to most other actors, and it may be more sensible to focus on the degree of reciprocity among pairs that have any ties. In our simple example, this would yield one reciprocated pair divided by two tied pairs, or a reciprocity rate of 0.500. The method just described is called the dyad method in *Network>Cohesion>Reciprocity*.

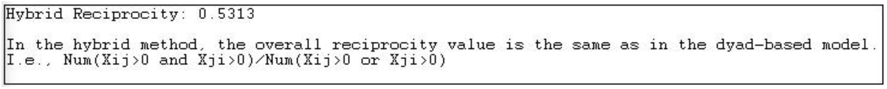
Rather than focusing on actors, we could focus on relations. We could ask, what percentage of all possible ties (or "arcs" of the direct graph) are parts of reciprocated structures? Here, two such ties (A to B and B to A) are a reciprocated structure among the six possible ties (AB, BA, AC, CA, BC, CB) or a reciprocity of 0.333. Analysts usually focus, instead, on the number of ties that are involved in reciprocal relations relative to the total number of actual ties (not possible ties). Here, this definition would give us 2/3 or 0.667. This approach is called the arc method in *Network>Cohesion>Reciprocity*. Here's a typical dialog for using this tool.



Input dataset: :\Program Files\Ucinet 6\DataFiles\KNOKBUR
Method: Hybrid
Row partition/blocking (if any):
Col partition/blocking (if any):
Utilize diagonal values? No
Output dataset: Reciprocity

Figure 8.4: *Dialog for Network>Cohesion>Reciprocity*

We've specified the "hybrid" method (the default) which is the same as the dyad approach. Note that it is possible to block or partition the data by some pre-defined attribute (like in the density example previously) to examine the degree of reciprocity within and between sub-populations. Figure 8.5 shows the results for the Knoke information network.



```
Hybrid Reciprocity: 0.5313

In the hybrid method, the overall reciprocity value is the same as in the dyad-based model.
I.e., Num(Xij>0 and Xji>0)/Num(Xij>0 or Xji>0)
```

Figure 8.5: *Reciprocity in the Knoke information network*

We see that, of all pairs of actors that have any connection, 53% of the pairs have a reciprocated connection. This is neither "high" nor "low" in itself, but does seem to suggest a considerable degree of institutionalized horizontal connection within this organizational population.

The alternative method of "arc" reciprocity (not shown here) yields a result of 0.6939. That is, of all the relations in the graph, 69% are parts of reciprocated ties.

8.4: Transitivity

Small group theorists argue that many of the most interesting and basic questions of social structure arise with regard to triads. Triads allow for a much wider range of possible sets of relations.

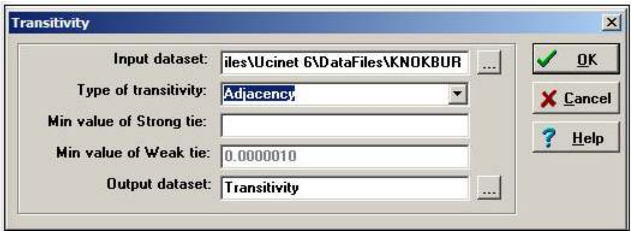
With undirected data, there are four possible types of triadic relations (no ties, one tie, two ties, or all three ties). Counts of the relative prevalence of these four types of relations across all possible triples (that is, a "triad census") can give a good sense of the extent to which a population is characterized by "isolation", "couples only", "structural holes" (i.e. where one actor is connected to two others, who are not connected to each other), or "clusters". UCINET does not have a routine for conducting triad censuses (see Pajek, which does).

With directed data, there are actually 16 possible types of relations among 3 actors, including relationships that exhibit hierarchy, equality, and the formation of exclusive groups (e.g. where two actors connect, and exclude the third). Thus, small group researchers suggest all of the really fundamental forms of social relationships can be observed in triads. Because of this interest, we may wish to conduct a "triad census" for each actor, and for the network as a whole (again, see Pajek).

In particular, we may be interested in the proportion of triads that are "transitive" (that is, display a type of balance where, if A directs a tie to B, and B directs a tie to C, then A also directs a tie to C). Such transitive or balanced triads are argued by some theorists to be the "equilibrium" or natural state toward which triadic relationships tend (not all theorists would agree!).

Of the 16 possible types of directed triads, six involve zero, one, or two relations - and can't display transitivity because there are not enough ties to do so. One type with 3 relations (AB, BC, CB) does not have any ordered triples (AB, BC) and hence can't display transitivity. In three more types of triads, there are ordered triples (AB, BC) but the relation between A and C is not transitive. The remaining types of triads display varying degrees of transitivity.

UCINET does not have extensive algorithms for examining full triad censuses and building more complex models based on them (e.g. balance, clusterability, ranked clusters). A more extended treatment of this approach, with supporting software is available from Pajek. Nonetheless, the *Network>Cohesion>Transitivity* algorithms in UCINET offer some interesting and flexible approaches to characterizing the transitivity of triads in populations. A typical dialog is shown in Figure 8.6.



Transitivity

Input dataset: iles\Ucinet 6\DataFiles\KNOKBUR

Type of transitivity: Adjacency

Min value of Strong tie:

Min value of Weak tie: 0.0000010

Output dataset: Transitivity

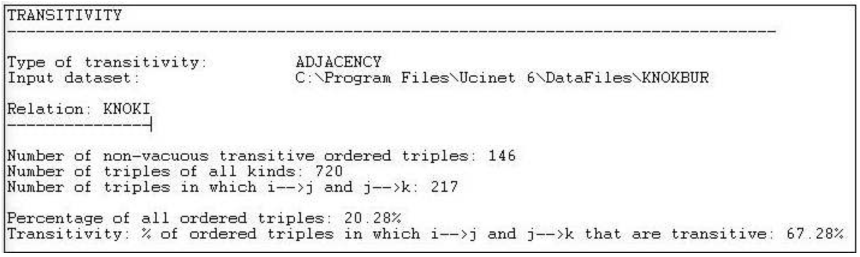
Figure 8.6: *Dialog of Network>Cohesion>Transitivity*

The Knoke information network is a binary, directed graph. For data of this type, the default definition of transitivity (i.e "*Adjacency*") is a reasonable approach. This means that we will count the number of times that, if we see AB and BC, we also see AC.

Network>Cohesion>Transitivity also provides some alternative definitions of what it means for a triad to be transitive which are useful for valued data.

A *strong* transitivity is one in which there are connections AB, BC, and AC, and the connection AC is stronger than the *Min value of Strong tie*. A *weak* transitivity is one in which there are connections AB, BC, and AC, and the value of AC is less than the threshold for a strong tie, but greater than the threshold *Min value of Weak tie*.

Two other methods are also available. A *Euclidean* transitivity is defined as a case where AB, BC, and AC are present, and AC has a value less than the sum of AB + BC. A *Stochastic* transitivity is defined as the case where AB, BC, and AC are present, and AC is less the the product AB * BC.



```
TRANSITIVITY
------------------------------------------------------------------------------------

Type of transitivity:         ADJACENCY
Input dataset:                C:\Program Files\Ucinet 6\DataFiles\KNOKBUR

Relation: KNOKI
---------------

Number of non-vacuous transitive ordered triples: 146
Number of triples of all kinds: 720
Number of triples in which i-->j and j-->k: 217

Percentage of all ordered triples: 20.28%
Transitivity: % of ordered triples in which i-->j and j-->k that are transitive: 67.28%
```

Figure 8.7: *Transitivity results for Knoke information network*

After performing a census of all possible triads, *Network>Cohesion>Transitivity* reports that it finds 146 transitive (directed) triples. That is, there are 146 cases where, if AB and BC are present, then AC is also present. There are a number of different ways in which we could try to norm this count so that it becomes more meaningful. One approach is to divide the number of transitive triads by the total number of triads of all kinds (720). This shows that 20.28% of all triads are transitive. Perhaps more meaningful is to norm the number of transitive triads by the number of cases where a single link could complete the triad. That is, norm the number of {AB, BC, AC} triads by the number of {AB, BC, anything} triads. Seen in this way, about 2/3 of all relations that could easily be transitive, actually are.

8.5: Clustering

Watts (1999) and many others have noted that in large, real-world networks (of all kinds of things) there is often a structural pattern that seems somewhat paradoxical.

On one hand, in many large networks (like, for example, the Internet) the average geodesic distance between any two nodes is relatively short. The "6-degrees" of distance phenomenon is an example of this. So, most of the nodes in even very large networks may be fairly close to one another. The average distance between pairs of actors in large empirical networks are often much shorter than in random graphs of the same size.

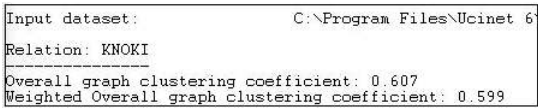
On the other hand, most actors live in local neighborhoods where most others are also connected to one another. That is, in most large networks, a very large proportion of the total number of ties are highly "clustered" into local neighborhoods. That is, the density in local neighborhoods of large graphs tend to be much higher than we would expect for a random graph of the same size.

Most of the people we know may also know one another - seeming to locate us in a very narrow social world. Yet, at the same time, we can be at quite short distances to vast numbers of people that we don't know at all. The "small world" phenomenon - a combination of short average path lengths over the entire graph, coupled with a strong degree of "clique-like" local neighborhoods - seems to have evolved independently in many large networks.

We've already discussed one part of this phenomenon. The average geodesic distance between all actors in a graph gets at the idea of how close actors are together. The other part of the phenomenon is the tendency towards dense local neighborhoods, or what is now thought of as "clustering".

One common way of measuring the extent to which a graph displays clustering is to examine the local neighborhood of an actor (that is, all the actors who are directly connected to ego), and to calculate the density in this neighborhood (leaving out ego). After doing this for all actors in the whole network, we can characterize the degree of clustering as an average of all the neighborhoods.

Figure 8.8 shows the output of *Network>Cohesion>Clustering Coefficient* as applied to the Knoke information network.

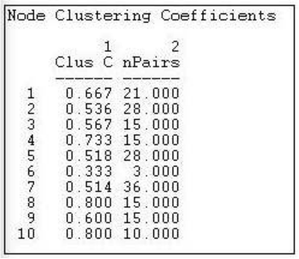


```
Input dataset:                    C:\Program Files\Ucinet 6

Relation: KNOKI
---------------
Overall graph clustering coefficient: 0.607
Weighted Overall graph clustering coefficient: 0.599
```

Figure 8.8: *Network>Cohesion>Clustering Coefficient of Knoke information network*

Two alternative measures are presented. The "overall" graph clustering coefficient is simply the average of the densities of the neighborhoods of all of the actors. The "weighted" version gives weight to the neighborhood densities proportional to their size; that is, actors with larger neighborhoods get more weight in computing the average density. Since larger graphs are generally (but not necessarily) less dense than smaller ones, the weighted average neighborhood density (or clustering coefficient) is usually less than the unweighted version. In our example, we see that all of the actors are surrounded by local neighborhoods that are fairly dense - our organizations can be seen as embedded in dense local neighborhoods to a fairly high degree. Lest we over-interpret, we must remember that the overall density of the entire graph in this population is rather high (0.54). So, the density of local neighborhoods is not really much higher than the density of the whole graph. In assessing the degree of clustering, it is usually wise to compare the cluster coefficient to the overall density.

We can also examine the densities of the neighborhoods of each actor, as is shown in Figure 8.9.



```
Node Clustering Coefficients

          1      2
     Clus C nPairs
     ------ ------
 1    0.667 21.000
 2    0.536 28.000
 3    0.567 15.000
 4    0.733 15.000
 5    0.518 28.000
 6    0.333  3.000
 7    0.514 36.000
 8    0.800 15.000
 9    0.600 15.000
10    0.800 10.000
```

Figure 8.9: *Node level clustering coefficients for Knoke information network*

The sizes of each actor's neighborhood is reflected in the number of pairs of actors in it. Actor 6, for example, has three neighbors, and hence three possible ties. Of these, only one is present - so actor 6 is not highly clustered. Actor 8, on the other hand, is in a slightly larger neighborhood (6 neighbors, and hence 15 pairs of neighbors), but 80% of all the possible ties among these neighbors are present. Actors 8 and 10 are embedded in highly clustered neighbors.

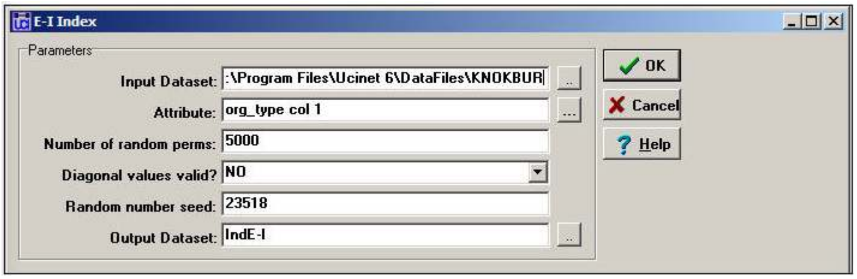
8.6: Group-External and Group-Internal Ties

Actors may be embedded in macro-structures, as well as in dyads, triads, and neighborhoods. Some macro-structures are social agents (like voluntary and formal organizations); some macro-structures are categorical units (like gender and ethnic groups). To understand the "texture" of the "social fabric" we might want to index the extent to which these macro-structures "cluster" the interaction patterns of individuals who fall within them.

Krackhardt and Stern (1988) developed a very simple and useful measure of the group embedding based on comparing the numbers of ties within groups and between groups. The E-I (external-internal) index takes the number of ties of group members to outsides, subtracts the number of ties to other group members, and divides by the total number of ties. The resulting index ranges from -1 (all ties are internal to the group) to +1 (all ties are external to the group). Since this measure is concerned with any connection between members, the directions of ties are ignored (i.e. either an out-tie or an in-tie constitutes a tie between two actors).

The E-I index can be applied at three levels: the entire population, each group, and each individual. That is, the network as a whole (all the groups) can be characterized in terms of the bounded-ness and closure of its sub-populations. We can also examine variation across the groups in their degree of closure; and, each individual can be seen as more or less embedded in their group.

Here's a sample of the dialog with *Network>Cohesion>E-I Index* in which we examine the Knoke information network that has been partitioned according to the attribute of organizational type (group 1 = governmental generalists, group 2 = non-governmental generalists, group 3 = welfare specialists).



E-I Index

Parameters

Input Dataset: :\Program Files\Ucinet 6\DataFiles\KNOKBUR

Attribute: org_type col 1

Number of random perms: 5000

Diagonal values valid? NO

Random number seed: 23518

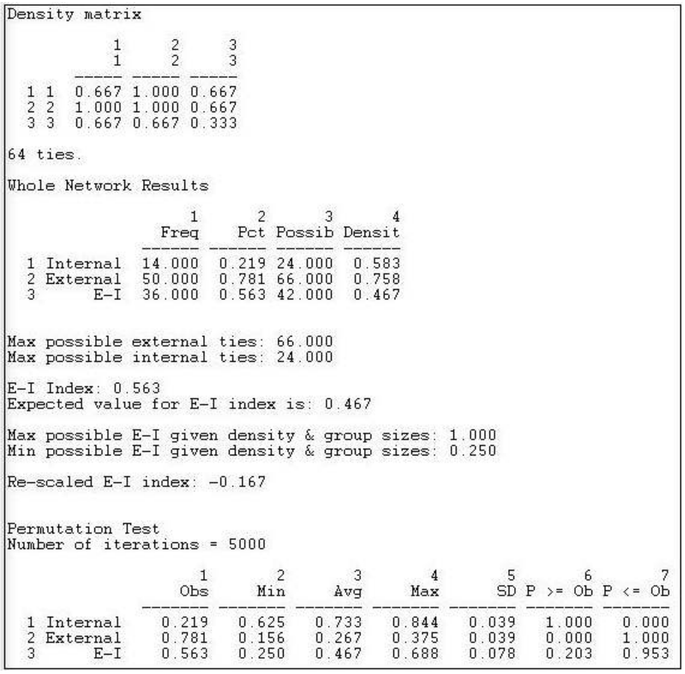
Output Dataset: IndE-I

Figure 8.10: *Dialog of Network>Cohesion>E-I Index*

The range of possible values of the E-I index is restricted by the number of groups, relative group sizes, and total number of ties in a graph. Often this range restriction is quite severe, so it is important to re-scale the coefficient to range between the maximum possible degree of "external-ness" (+1) and the maximum possible degree of "internal-ness" (-1). As Blau and others have noted, the relative sizes of sub-populations have dramatic consequences for the degree of internal and external contacts, even when individuals may choose contacts at random.

To assess whether a given E-I index value is significantly different than what would be expected by random mixing (i.e. no preference for within or without group ties by group members), a permutation test is performed by *Network>Cohesion>E-I Index*. A large number of trials are run in which the block of groups is maintained, and the overall density of ties is maintained, but the actual ties are randomly distributed. From a large number of trials (the default is 5000), a sampling distribution of the numbers of internal and external ties - under the assumption that ties are randomly distributed - can be calculated. This sampling distribution can then be used to assess the frequency with which the observed result would occur by sampling from a population in which ties are randomly distributed.

Let's look first at the results for the graph as a whole, in Figure 8.11.



```
Density matrix

             1      2      3
             1      2      3
         ------ ------ ------
  1 1    0.667  1.000  0.667
  2 2    1.000  1.000  0.667
  3 3    0.667  0.667  0.333

64 ties.

Whole Network Results

                       1       2       3       4
                    Freq     Pct  Possib  Densit
                 ------- ------- ------- -------
  1 Internal     14.000   0.219  24.000   0.583
  2 External     50.000   0.781  66.000   0.758
  3      E-I     36.000   0.563  42.000   0.467


Max possible external ties: 66.000
Max possible internal ties: 24.000

E-I Index: 0.563
Expected value for E-I index is: 0.467

Max possible E-I given density & group sizes: 1.000
Min possible E-I given density & group sizes: 0.250

Re-scaled E-I index: -0.167


Permutation Test
Number of iterations = 5000

                        1        2        3        4        5        6        7
                      Obs      Min      Avg      Max       SD P >= Ob P <= Ob
                 -------- -------- -------- -------- -------- -------- --------
  1 Internal        0.219    0.625    0.733    0.844    0.039    1.000    0.000
  2 External        0.781    0.156    0.267    0.375    0.039    0.000    1.000
  3      E-I        0.563    0.250    0.467    0.688    0.078    0.203    0.953
```

Figure 8.11: *E-I index output for the Knoke information network - whole network*

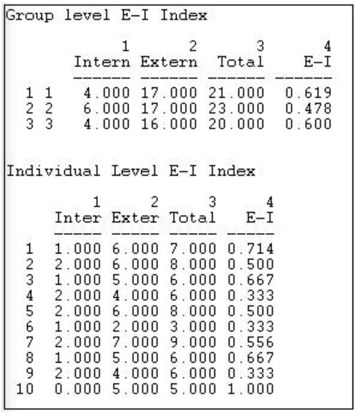
The observed block densities are presented first. Since any tie (in or out) is regarded as a tie, the densities in this example are quite high. The densities off the main diagonal (out-group ties) appear to be slightly more prevalent than the densities on the main diagonal (in-group ties).

Next, we see the numbers of internal ties (14, or 22%) and external ties (50, or 78%) that yield a raw (not re-scaled) E-I index of +0.563. That is, a preponderance of external over internal ties for the graph as a whole. Also shown are the maximum possible numbers of internal and external ties given the group sizes and density. Note that, due to these constraints, the result of a preponderance of external ties is not unexpected - under a random distribution, the E-I index would be expected to have a value of +0.467, which is not very much different from the observed value.

We see that, given the group sizes and density of the graph, the maximum possible value of the index (+1.0) and its minimum value (+0.25) are both positive. If we re-scale the observed value of the E-I index (+0.563) to fall into this range, we obtain a re-scaled index value of -0.167. This suggests that, given the demographic constraints and overall density, there is a very modest tendency toward group closure.

The last portion of the results gives the values of the permutation-based sampling distribution. Most important here is the standard deviation of the sampling distribution of the index, or its standard error (0.078). This suggests that the value of the raw index is expected to vary by this much from trial to trial (on average) just by chance. Given this result, we can compare the observed value in our sample (+0.563) to the expected value (+0.467) relative to the standard error. The observed difference of about 0.10 could occur fairly frequently just by sampling variability (p = 0.203). Most analysts would not reject the null hypothesis that the deviation from randomness was not "significant". That is, we cannot be confident that the observed mild bias toward group closure is not random variation.

The E-I index can also be calculated for each group and for each individual. These index numbers describe the tendencies toward group closure of each of the groups, and the propensity of each individual to have ties within their group. Figure 8.12 displays the results.



```
Group level E-I Index

                 1       2       3       4
            Intern  Extern   Total     E-I
            ------  ------  ------  ------
  1 1        4.000  17.000  21.000   0.619
  2 2        6.000  17.000  23.000   0.478
  3 3        4.000  16.000  20.000   0.600


Individual Level E-I Index

             1     2     3     4
         Inter Exter Total   E-I
         ----- ----- ----- -----
  1      1.000 6.000 7.000 0.714
  2      2.000 6.000 8.000 0.500
  3      1.000 5.000 6.000 0.667
  4      2.000 4.000 6.000 0.333
  5      2.000 6.000 8.000 0.500
  6      1.000 2.000 3.000 0.333
  7      2.000 7.000 9.000 0.556
  8      1.000 5.000 6.000 0.667
  9      2.000 4.000 6.000 0.333
 10      0.000 5.000 5.000 1.000
```

Figure 8.12: *E-I index output for the Knoke information network - groups and individuals*

The first panel of Figure 8.12 shows the raw counts of ties within and without each of the three types of organizations, and the E-I index for each group. Governmental generalists (group 2) appear to be somewhat more likely to have out-group ties than either of the other sub-populations. The relatively small difference, though, should be treated with considerable caution given the sampling variability (we cannot directly apply the standard error estimate for the whole graph to the results for sub-populations or individuals, but they are suggestive). We should also note that the E-I results for groups and individuals are in "raw" form, and not "re-scaled".

There is considerable variability across individuals in their propensity to in-group ties, as can be seen in the last panel of the results. Several actors (4, 6, 9) tend toward closure - having a preponderance of ties within their own group; a couple others (10, 1) tend toward a preponderance of ties outside their groups.

8.7: Krackhardt's Graph Theoretical Dimensions of Hierarchy

Embedding of actors in dyads, triads, neighborhoods, clusters, and groups are all ways in which the social structure of a population may display "texture". All of these forms of embedding structures speak to the issue of the "horizontal differentiation" of the population - separate, but not necessarily ranked or unequal groupings.

A very common form of embedding of actors in structures, though, does involve unequal rankings. Hierarchies, in which individuals or sub-populations are not only differentiated, but also ranked, are extremely common in social life. The degree of hierarchy in a population speaks to the issue of "vertical differentiation".

While we all have an intuitive sense of what it means for a structure to be a hierarchy, most would agree that structures can be "more or less" hierarchical. It is necessary to be quite precise about the meaning of the term if we are going to build indexes to measure the degree of hierarchy.

Krackhardt (1994) provided an elegant definition of the meaning of hierarchy, and developed measures of each of the four component dimensions of the concept that he identified. Krackhardt defines a pure, "ideal typical" hierarchy as an "out-tree" graph. An out-tree graph is a directed graph in which all points are connected, and all but one node (the "boss") has an in-degree of one. This means that all actors in the graph (except the ultimate "boss") have a single superior node. The simplest "hierarchy" is a directed line graph A to B to C to D... More complex hierarchies may have wider, and varying "spans of control" (out-degrees of points).

This very simple definition of the pure type of hierarchy can be deconstructed into four individually necessary and jointly sufficient conditions. Krackhardt develops index numbers to assess the extent to which each of the four dimensions deviates from the pure ideal type of an out-tree, and hence develops four measures of the extent to which a given structure resembles the ideal typical hierarchy.

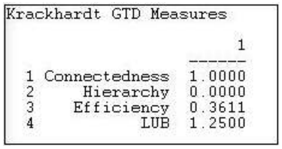
1) Connectedness: To be a pure out-tree, a graph must be connected into a single component - all actors are embedded in the same structure. We can measure the extent to which this is not true by looking at the ratio of the number of pairs in the directed graph that are reachable relative to the number of ordered pairs. That is, what proportion of actors cannot be reached by other actors? Where a graph has multiple components - multiple un-connected sub-populations - the proportion not reachable can be high. If all the actors are connected in the same component, if there is a "unitary" structure, the graph is more hierarchical.

2) Hierarchy: To be a pure out-tree, there can be no reciprocated ties. Reciprocal relations between two actors imply equal status, and this denies pure hierarchy. We can assess the degree of deviation from pure hierarchy by counting the number of pairs that have reciprocated ties relative to the number of pairs where there is any tie; that is, what proportion of all tied pairs have reciprocated ties.

3) Efficiency: To be a pure out-tree each node must have an in-degree of one. That is, each actor (except the ultimate boss) has a single boss. This aspect of the ideal type is termed "efficiency" because structures with multiple bosses have unnecessary redundant communication of orders from superiors to subordinates. The amount of deviation from this aspect of the pure out-tree can be measured by counting the difference between the actual number of links (minus 1, since the ultimate boss has no boss) and the maximum possible number of links. The bigger the difference, the greater the inefficiency. This dimension then measures the extent to which actors have a "single boss".

4) Least upper bound (LUB): To be a pure out-tree, each pair of actors (except pairs formed between the ultimate boss and others) must have an actor that directs ties to both - that is, command must be unified. The deviation of a graph from this condition can be measured by counting the numbers of pairs of actors that do not have a common boss relative to the number of pairs that could (which depends on the number of actors and the span of control of the ultimate boss).

The *Network>Network Properties>Krackhardt GTD* algorithms calculate indexes of each of the four dimensions, where higher scores indicate greater hierarchy. Figure 8.13 shows the results for the Knoke information network.



```
Krackhardt GTD Measures

                        1
                  -------
 1 Connectedness  1.0000
 2     Hierarchy  0.0000
 3    Efficiency  0.3611
 4           LUB  1.2500
```

Figure 8.13: *Output of Network>Network Properties>Krackhardt GTD for Knoke information network*

The information network does form a single component, as there is at least one actor that can reach all others. So, the first dimension of pure hierarchy - that all the actors be embedded in a single structure - is satisfied. The ties in the information exchange network, though, are very likely to be reciprocal (at least insofar as they can be, given the limitations of the density). There are a number of nodes that receive information from multiple others, so the network is not "efficient". The least upper bound measure (the extent to which all actors have a boss in common) reports a value of 1.25, which would appear to be out of range and, frankly, is a puzzle.

8.8: Summary

This chapter and the next are concerned with the ways in which networks display "structure" or deviation from random connection. In the current chapter, we've approached the same issue of structure from the "top-down" by looking at patterns of macro-structure in which individuals are embedded in non-random ways. Individuals are embedded (usually simultaneously) in dyads, triads, face-to-face local groups of neighbors, and larger organizational and categorical social structures. The tools in the current chapter provide some ways of examining the "texture" of the structuring of the whole population.

In the next chapter, we will focus on the same issue of connection and structure from the "bottom-up". That is, we'll look at structure from the point of view of the individual "ego".

Taken together, the approaches in Chapters 8 and 9 illustrate, again, the "duality" of social structure in which individuals make social structure, but do so within a matrix of constraints and opportunities imposed by larger patterns.

CHAPTER OVERVIEW

9: Ego Networks

In the previous chapter we looked at the idea of the amount of "embedding" in whole networks - loosely. The main theme was to understand and index the extent and nature of the pattern of "constraint" on actors that results from the way that they are connected to others. These approaches may tell us some interesting things about the entire population and its sub-populations; but, they don't tell us very much about the opportunities and constraints facing individuals.

9.1: Introduction to Ego Networks

In the previous chapter we looked at the idea of the amount of "embedding" in whole networks - loosely: the extent to which actors find themselves in social structures characterized by dense, reciprocal, transitive, strong ties. The main theme was to understand and index the extent and nature of the pattern of "constraint" on actors that results from the way that they are connected to others. These approaches may tell us some interesting things about the entire population and its sub-populations; but, they don't tell us very much about the opportunities and constraints facing individuals.

If we want to understand variation in the behavior of individuals, we need to take a closer look at their local circumstances. Describing and indexing the variation across individuals in the way they are embedded in "local" social structures is the goal of the analysis of ego networks. We need some definitions.

- "Ego" is an individual "focal" node. A network has as many egos as it has nodes. Egos can be persons, groups, organizations, or whole societies.
- "Neighborhood" is the collection of ego and all nodes to whom ego has a connection at some path length. In social network analysis, the "neighborhood" is almost always one-step; that is, it includes only ego and actors that are directly adjacent. The neighborhood also includes all of the ties among all of the actors to whom ego has a direct connection. The boundaries of ego networks are defined in terms of neighborhoods.
- "N-step neighborhood" expands the definition of the size of ego's neighborhood by including all nodes to whom ego has a connection at a path length of N, and all the connections among all of these actors. Neighborhoods of greater path length than 1 (i.e. ego's adjacent nodes) are rarely used in social network analysis. When we use the term neighborhood here, we mean the one-step neighborhood.
- "In" and "Out" are other kinds of neighborhoods. Most of the analysis of ego networks uses simple graphs (i.e. graphs that are symmetric, and show only connection/not, not direction). If we are working with a directed graph, it is possible to define different kinds of ego-neighborhoods. An "out" neighborhood would include all the actors to whom ties are directed from ego. An "in" neighborhood would include all the actors who sent ties directly to ego. We might want to define a neighborhood of only those actors to whom ego had reciprocated ties. There isn't a single "right" way to define an ego neighborhood for every research question.

Most analysis of ego networks uses binary data - two actors are connected or they aren't, and this defines the ego neighborhood. But if we have measured the strength of the relation between two actors, and even its valence (positive or negative), we need to make choices about when we are going to decide that another actor is ego's neighbor. With ties that are measured as strengths or probabilities, a reasonable approach is to define a cut-off value (or, better, explore several reasonable alternatives). This reasoning is the origin of strong tie neighborhoods and weak tie neighborhoods. Where the information about ties includes information about positive/negative, the most common approach is to analyze the positive tie neighborhood and the negative tie neighborhood separately.

9.2: Ego Network Data

Ego network data commonly arise in two ways:

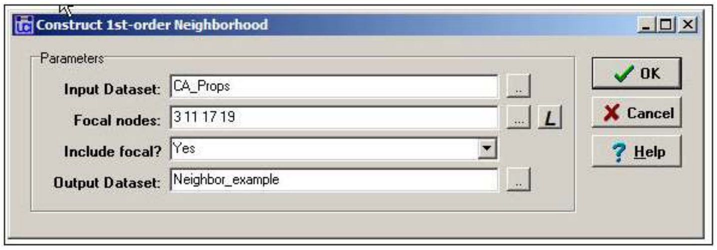
Surveys may be used to collect information on ego networks. We can ask each research subject to identify all of the actors to whom they have a connection, and to report to us (as an informant) what the ties are among these other actors. Alternatively, we could use a two-stage snowball method; first, ask ego to identify others to whom ego has a tie, then ask each of those identified about their ties to each of the others identified.

Data collected in this way cannot directly inform us about the overall embeddedness of the networks in a population, but it can give us information on the prevalence of various kinds of ego networks in even very large populations. When data are collected this way, we essentially have a data structure that is composed of a collection of networks. As the actors in each network are likely to be different people, the networks need to be treated as separate actor-by-actor matrices stored as different data sets (i.e. it isn't a good idea to "stack" the multiple networks in the same data file, because the multiple matrices do not represent multiple relations among the same set of actors).

A modification of the survey method can give rise to a multiplex data structure (that is, a "stack" of actor-by-actor matrices of equal dimension). If we ask each ego to characterize relation with the occupants of social roles (or a particular occupant of a role), and to also report on the relations among occupants of those roles, we can build "conformable" matrices for each ego. For example, suppose that we asked a number of egos "Do you have a male fried or friends in your classroom?", "Do you have a female friend or friends in your classroom?", and "Are your male friends friends of your female friends?", the resulting data for each ego would have three nodes (in the sense of social roles, but not individuals) they could be treated as a type of multiplex data that we will discuss more later on.

The second major way in which ego network data arise is by "extracting" them from regular complete network data. The *Data>Extract* approach can be used to select a single actor and their ties, but would not include the ties among the "alters". The *Data>Subgraphs from partitions* approach could be used if we had previously identified the members of a particular ego neighborhood, and stored this as an attribute vector.

More commonly, though, we would want to extract multiple, or even all of the ego networks from a full network to be stored as separate files. For this task, the *Data>Egonet* tool is ideal. Here is an example of the dialog for using the tool:



Input Dataset: CA_Props

Focal nodes: 3 11 17 19

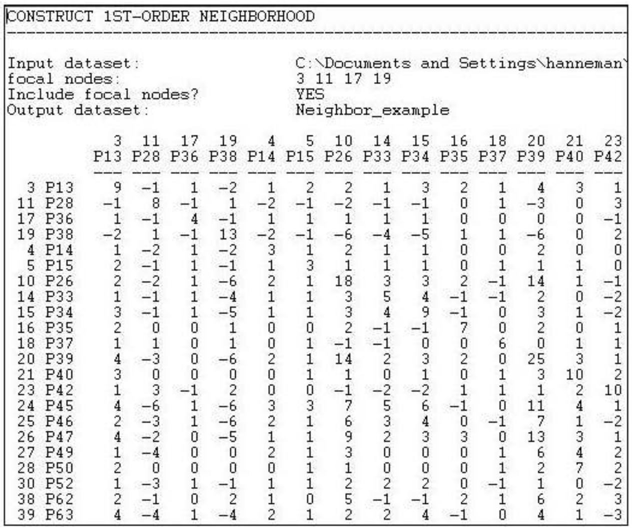
Include focal? Yes

Output Dataset: Neighbor_example

Figure 9.1: *Dialog for Data>Egonet*

Here we are focusing on ballot proposition campaigns in California that are connected by having donors in common (i.e. CA_Props is a proposition-by-proposition valued matrix). We've said that we want to extract a network that includes the 3rd, 11th, 17th, and 19th rows/columns, and all the nodes that are connected to any of these actors. More commonly, we might select a single "ego". The list of focal nodes can be provided either as an attribute file, by typing in the list of row numbers, or by selecting the node labels of the desired actors.

A picture of part of the resulting data, stored as a new file called "Neighbor_example" is shown in Figure 9.2.



```
CONSTRUCT 1ST-ORDER NEIGHBORHOOD
--------------------------------------------------------------------

Input dataset:                  C:\Documents and Settings\hanneman\
focal nodes:                    3 11 17 19
Include focal nodes?            YES
Output dataset:                 Neighbor_example

            3  11  17  19   4   5  10  14  15  16  18  20  21  23
          P13 P28 P36 P38 P14 P15 P26 P33 P34 P35 P37 P39 P40 P42
          --- --- --- --- --- --- --- --- --- --- --- --- --- ---
  3 P13     9  -1   1  -2   1   2   2   1   3   2   1   4   3   1
 11 P28    -1   8  -1   1  -2  -1  -2  -1  -1   0   1  -3   0   3
 17 P36     1  -1   4  -1   1   1   1   1   1   0   0   0   0  -1
 19 P38    -2   1  -1  13  -2  -1  -6  -4  -5   1   1  -6   0   2
  4 P14     1  -2   1  -2   3   1   2   1   1   0   0   2   0   0
  5 P15     2  -1   1  -1   1   3   1   1   1   0   1   1   1   0
 10 P26     2  -2   1  -6   2   1  18   3   3   2  -1  14   1  -1
 14 P33     1  -1   1  -4   1   1   3   5   4  -1  -1   2   0  -2
 15 P34     3  -1   1  -5   1   1   3   4   9  -1   0   3   1  -2
 16 P35     2   0   0   1   0   0   2  -1  -1   7   0   2   0   1
 18 P37     1   1   0   1   0   1  -1  -1   0   0   6   0   1   1
 20 P39     4  -3   0  -6   2   1  14   2   3   2   0  25   3   1
 21 P40     3   0   0   0   0   1   1   0   1   0   1   3  10   2
 23 P42     1   3  -1   2   0   0  -1  -2  -2   1   1   1   2  10
 24 P45     4  -6   1  -6   3   3   7   5   6  -1   0  11   4   1
 25 P46     2  -3   1  -6   2   1   6   3   4   0  -1   7   1  -2
 26 P47     4  -2   0  -5   1   1   9   2   3   3   0  13   3   1
 27 P49     1  -4   0   0   2   1   3   0   0   0   1   6   4   2
 28 P50     2   0   0   0   0   1   1   0   0   0   1   2   7   2
 30 P52     1  -3   1  -1   1   1   2   2   2   0  -1   1   0  -2
 38 P62     2  -1   0   2   1   0   5  -1  -1   2   1   6   2   3
 39 P63     4  -4   1  -4   2   1   2   2   4  -1   0   4   1  -3
```

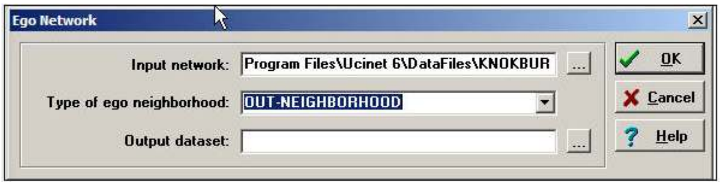
Figure 9.2: *(Partial) output of Data>Egonet*

Extracting sub-graphs, based on a focal actor or set of actors (e.g. "elites") can be a very useful way of looking at a part of a whole network, or the condition of an individual actor. The *Data>Egonet* tool is helpful for creating datasets that are good for graphing and separate analysis - particularly when the networks in which the focal actor/actors embedded are quite large.

It is not necessary, however, to create separate ego network datasets for each actor to be analyzed. The approaches to analysis that we'll review later generate output for the first-order ego network of every node in a dataset. For small datasets, there is often no need to extract separate ego networks.

9.3: Ego Network Density

There are quite a few characteristics of the ego neighborhoods of actors that may be of interest. The *Network>Ego networks>Density* tools in UCINET calculate a substantial number of indexes that describe aspects of the neighborhood of each ego in a data set. Here is an example of the dialog, applied to the Knoke information exchange data (these are binary, directed connections).



Ego Network

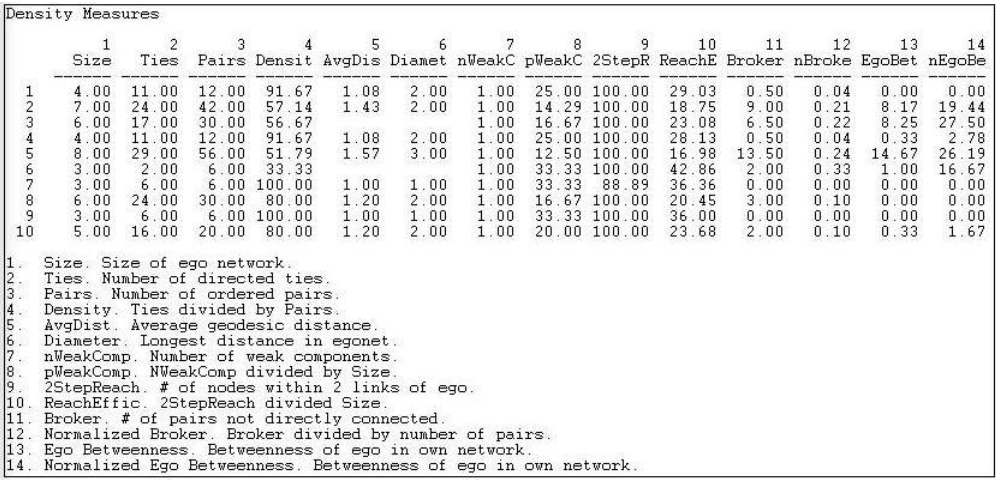
Input network: Program Files\Ucinet 6\DataFiles\KNOKBUR

Type of ego neighborhood: OUT-NEIGHBORHOOD

Output dataset:

Figure 9.3: *Dialog for Network>Ego networks>Density*

In this example, we've decided to examine "out neighborhoods" (in neighborhoods or undirected neighborhoods can also be selected). We've elected not to save the output as a dataset (if you wanted to do further analysis, or treat ego network descriptive statistics as node attributes, you might want to save the results as a file or use in other routines or Netdraw). Here are the results:



```
Density Measures

          1      2      3      4      5      6      7      8      9     10     11     12     13     14
       Size   Ties  Pairs Densit AvgDis Diamet nWeakC pWeakC 2StepR ReachE Broker nBroke EgoBet nEgoBe
     ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1    4.00  11.00  12.00  91.67   1.08   2.00   1.00  25.00 100.00  29.03   0.50   0.04   0.00   0.00
  2    7.00  24.00  42.00  57.14   1.43   2.00   1.00  14.29 100.00  18.75   9.00   0.21   8.17  19.44
  3    6.00  17.00  30.00  56.67                 1.00  16.67 100.00  23.08   6.50   0.22   8.25  27.50
  4    4.00  11.00  12.00  91.67   1.08   2.00   1.00  25.00 100.00  28.13   0.50   0.04   0.33   2.78
  5    8.00  29.00  56.00  51.79   1.57   3.00   1.00  12.50 100.00  16.98  13.50   0.24  14.67  26.19
  6    3.00   2.00   6.00  33.33                 1.00  33.33 100.00  42.86   2.00   0.33   1.00  16.67
  7    3.00   6.00   6.00 100.00   1.00   1.00   1.00  33.33  88.89  36.36   0.00   0.00   0.00   0.00
  8    6.00  24.00  30.00  80.00   1.20   2.00   1.00  16.67 100.00  20.45   3.00   0.10   0.00   0.00
  9    3.00   6.00   6.00 100.00   1.00   1.00   1.00  33.33 100.00  36.00   0.00   0.00   0.00   0.00
 10    5.00  16.00  20.00  80.00   1.20   2.00   1.00  20.00 100.00  23.68   2.00   0.10   0.33   1.67

1.   Size. Size of ego network.
2.   Ties. Number of directed ties.
3.   Pairs. Number of ordered pairs.
4.   Density. Ties divided by Pairs.
5.   AvgDist. Average geodesic distance.
6.   Diameter. Longest distance in egonet.
7.   nWeakComp. Number of weak components.
8.   pWeakComp. NWeakComp divided by Size.
9.   2StepReach. # of nodes within 2 links of ego.
10.  ReachEffic. 2StepReach divided Size.
11.  Broker. # of pairs not directly connected.
12.  Normalized Broker. Broker divided by number of pairs.
13.  Ego Betweenness. Betweenness of ego in own network.
14.  Normalized Ego Betweenness. Betweenness of ego in own network.
```

Figure 9.4: *Ego network density output for Knoke information out neighborhoods*

There's a lot of information here, and we should make a few comments.

Note that there is a line of data for each of the 10 organizations in the dataset. Each line describes the one-step ego neighborhood of a particular actor. Of course, many of the actors are members of many neighborhoods - so each actor may be involved in many lines of data.

- *Size of ego network* is the number of nodes that are one-step out neighbors of ego, plus ego itself. Actor 5 has the largest ego network; actors 6, 7, and 9 have the smallest networks.
- *Number of directed ties* is the number of connections among all the nodes in the ego network. Among the four actors in ego 1's network, there are 11 ties.
- *Number of ordered pairs* is the number of possible directed ties in each ego network. In node 1's network, there are four actors, so there are 4*3 possible directed ties.
- *Density* is, as the output says, the number of ties divided by the number of pairs. That is, what percentage of all possible ties in each ego network are actually present? Note that actors 7 and 9 live in neighborhoods where all actors send information to all other actors; they are embedded in very dense local structures. The welfare rights organization (node 6) lives in a small world where the members are not tightly connected. This kind of difference in the constraints and opportunities facing actors in their local neighborhoods may be very consequential.
- *Average geodesic distance* is the mean of the shortest path lengths among all connected pairs in the ego network. Where everyone is directly connected to everyone (e.g. nodes 7 and 9) this distance is one. In our example, the largest average path length for connected neighbors is for actor 5 (average distances among members of the neighborhood is 1.57).

- *Diameter* of an ego network is the length of the longest path between connected actors (just as it is for any network). The idea of a network diameter is to index the span or extensiveness of the network - how far apart the two furthest actors are. In the current example, they are not very far apart in the ego networks of most actors.
- In addition to these fairly basic and reasonably straightforward measures, the output provides some more exotic measures that get at some quite interesting ideas about ego neighborhoods that have been developed by a number of social network researchers.
- *Number of weak components*: A weak component is the largest number of actors who are connected, disregarding the direction of the ties (a strong component pays attention to the direction of the ties for directed data). If ego was connected to A and B (who are connected to one another), and ego is connected to C and D (who are connected to one another), but A and B are not connected in any way to C and D (except by way of everyone being connected to ego) then there would be two "weak components" in ego's neighborhood. In our example, there are no such cases - each ego is embedded in a single component neighborhood. That is, there are no cases where ego is the only connection between otherwise disjoint sets of actors.
- *Number of weak components divided by size*: The likelihood that there would be more than one weak component in ego's neighborhood would be a function of neighborhood size if connections were random. So, to get a sense of whether ego's role in connecting components is "unexpected" give the size of the network, it is useful to normalize the count of components by size. In our example, since there are no cases of multiple components, this is a pretty meaningless exercise.
- *Two-step reach* goes beyond ego's one-step neighborhood to report the percentage of all actors in the whole network that are within two directed steps of ego. In our example, only node 7 cannot get a message to all other actors within "friend-of-a-friend" distance.
- *Reach efficiency* (two-step reach divided by size) norms the two-step reach by dividing it by size. The idea here is: how much (non redundant) secondary contact do I get for each unit of primary contact? If reach efficiency is high, then I am getting a lot of "bang for my buck" in reaching a wider network for each unit of effort invested in maintaining a primary contact. If my neighbors, on the average, have few contacts that I don't have, I have low efficiency.
- *Brokerage* (number of pairs not directly connected): The idea of brokerage (more on this later) is that ego is the "go-between" for pairs of other actors. In an ego network, ego is connected to every other actor (by definition). If these others are not connected directly to one another, ego may be a "broker", that is, ego falls on the path between the others. One item of interest is simply how much potential for brokerage there is for each actor (how many times pairs of neighbors in ego's network are not directly connected). In our example, actor number 5, who is connected to almost everyone, is in a position to broker many connections.
- *Normalized brokerage* (brokerage divided by number of pairs) assesses the extent to which ego's role is that of broker. One can be in a brokering position a number of times, but this is a small percentage of the total possible connections in a network (e.g. the network is large). Given the large size of actor 5's network, the relative frequency with which actor 5 plays the broker is not so exceptional.
- *Betweenness* is an aspect of the larger concept of "centrality". A later chapter provides a more in-depth treatment of the concept and its application to whole networks. For the moment though, it's pretty easy to get the basic idea. Ego is "between" two other actors if ego lies on the shortest directed path from one to the other. The ego betweenness measure indexes the percentage of all geodesic paths from neighbor to neighbor that pass through ego.
- *Normalized Betweenness* compares the actual betweenness of ego to the maximum possible betweenness in a neighborhood of the size and connectivity of ego's. The "maximum" value for betweenness would be achieved when ego is the center of a "star" network; that is, no neighbors communicate directly with one another, and all directed communications between pairs of neighbors go through ego.

The ideas of "brokerage" and "betweenness" are slightly differing ways of indexing just how "central" or "powerful" ego is within their own neighborhood. This aspect of how an actor's embedding may provide them with strategic advantage has received a great deal of attention. The next two sections, on "structural holes" and "brokerage", elaborate on ways of looking at positional opportunity and constraint of individual actors.

9.4: Structural Holes

In several important works, Ronald Burt coined and popularized the term "structural holes" to refer to some very important aspects of positional advantage/disadvantage of individuals that result from how they are embedded in neighborhoods. Burt's formalization of these ideas, and his development of a number of measures (including the computer program *Structure*, that provides these measures and other tools) has facilitated a great deal of further thinking about how and why the ways that an actor is connected affect their constraints and opportunities, and hence their behavior.

The basic idea is simple, as good ideas often are.

Imagine a network of three actors (A, B, and C), in which each is connected to each of the others, as in Figure 9.5.

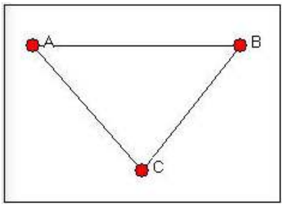


Figure 9.5: *Three actor network with no structural holes*

Let's focus on actor A (of course, in this case, the situations of B and C are identical in this particular network). Suppose that actor A wanted to influence or exchange with another actor. Assume that both B and C may have some interest in interacting or exchanging, as well. Actor A will not be in a strong bargaining position in this network, because both of A's potential exchange partners (B and C) have alternatives to treating with A; they could isolate A, and exchange with one another.

Now imagine that we open a "structural hole" between actors B and C, as in Figure 9.6. That is, a relation or tie is "absent" such that B and C cannot exchange (perhaps they are not aware of one another, or there are very high transaction costs involved in forming a tie).

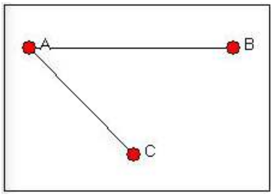
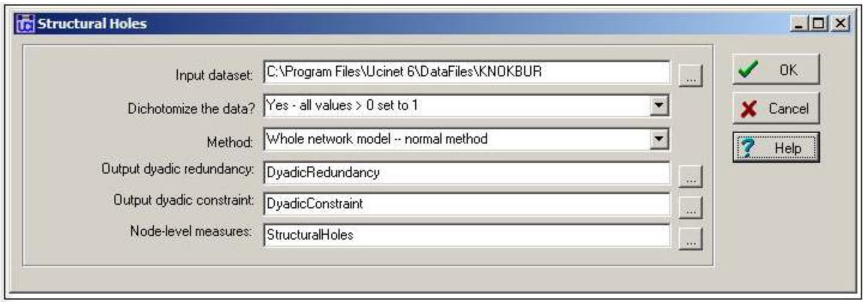


Figure 9.6: *Three actor network with a structural hole*

In this situation, actor A has an advantaged position as a direct result of the "structural hole" between actors B and C. Actor A has two alternative exchange partners; actors B and C have only one choice, if they choose to (or must) enter into an exchange.

Real networks, of course, usually have more actors. But, as networks grow in size, they tend to become less dense (how many relations can each actor support?). As density decreases, more "structural holes" are likely to open in the "social fabric". These holes, and how and where they are distributed can be a source of inequality (in both the strict mathematical sense and the sociological sense) among actors embedded in networks.

Network>Ego Networks>Structural Holes examines the position of each actor in their neighborhood for the presence of structural holes. A number of measures (most proposed by Burt) that describe various aspects of the advantage or disadvantage of the actor are also computed. Figure 9.7 shows a typical dialog box; we're looking at the Knoke information network again.


Structural Holes

Input dataset: C:\Program Files\Ucinet 6\DataFiles\KNOKBUR

Dichotomize the data? Yes - all values > 0 set to 1

Method: Whole network model -- normal method

Output dyadic redundancy: DyadicRedundancy

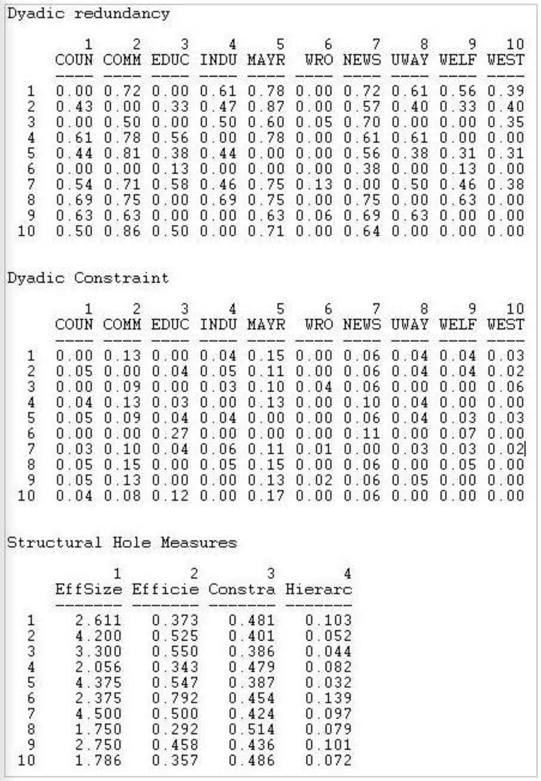
Output dyadic constraint: DyadicConstraint

Node-level measures: StructuralHoles

Figure 9.7: *Network>Ego Networks>Structural Holes dialog*

Measures related to structural holes can be computed on both valued and binary data. The normal practice in sociological research has been to use binary (a relation is present or not). Interpretation of the measures becomes quite difficult with valued data (at least I find it difficult). As an alternative to losing the information that valued data may provide, the input data could be dichotomized (*Transform>Dichotomize*) at various levels of strength. The structural holes measures may be computed for either directed or undirected data - and the interpretation, of course, depends on which is used. Here, we've used the directed binary data. Three output arrays are produced, and can be saved as separate files (or not, as the output reports all three).

The results are shown in Figure 9.8, and need a bit of explanation.



```
Dyadic redundancy

        1    2    3    4    5    6    7    8    9   10
     COUN COMM EDUC INDU MAYR  WRO NEWS UWAY WELF WEST
     ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  1  0.00 0.72 0.00 0.61 0.78 0.00 0.72 0.61 0.56 0.39
  2  0.43 0.00 0.33 0.47 0.87 0.00 0.57 0.40 0.33 0.40
  3  0.00 0.50 0.00 0.50 0.60 0.05 0.70 0.00 0.00 0.35
  4  0.61 0.78 0.56 0.00 0.78 0.00 0.61 0.61 0.00 0.00
  5  0.44 0.81 0.38 0.44 0.00 0.00 0.56 0.38 0.31 0.31
  6  0.00 0.00 0.13 0.00 0.00 0.00 0.38 0.00 0.13 0.00
  7  0.54 0.71 0.58 0.46 0.75 0.13 0.00 0.50 0.46 0.38
  8  0.69 0.75 0.00 0.69 0.75 0.00 0.75 0.00 0.63 0.00
  9  0.63 0.63 0.00 0.00 0.63 0.06 0.69 0.63 0.00 0.00
 10  0.50 0.86 0.50 0.00 0.71 0.00 0.64 0.00 0.00 0.00


Dyadic Constraint

        1    2    3    4    5    6    7    8    9   10
     COUN COMM EDUC INDU MAYR  WRO NEWS UWAY WELF WEST
     ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  1  0.00 0.13 0.00 0.04 0.15 0.00 0.06 0.04 0.04 0.03
  2  0.05 0.00 0.04 0.05 0.11 0.00 0.06 0.04 0.04 0.02
  3  0.00 0.09 0.00 0.03 0.10 0.04 0.06 0.00 0.00 0.06
  4  0.04 0.13 0.03 0.00 0.13 0.00 0.10 0.04 0.00 0.00
  5  0.05 0.09 0.04 0.04 0.00 0.00 0.06 0.04 0.03 0.03
  6  0.00 0.00 0.27 0.00 0.00 0.00 0.11 0.00 0.07 0.00
  7  0.03 0.10 0.04 0.06 0.11 0.01 0.00 0.03 0.03 0.02
  8  0.05 0.15 0.00 0.05 0.15 0.00 0.06 0.00 0.05 0.00
  9  0.05 0.13 0.00 0.00 0.13 0.02 0.06 0.05 0.00 0.00
 10  0.04 0.08 0.12 0.00 0.17 0.00 0.06 0.00 0.00 0.00


Structural Hole Measures

            1        2        3        4
      EffSize  Efficie  Constra  Hierarc
     -------- -------- -------- --------
  1     2.611    0.373    0.481    0.103
  2     4.200    0.525    0.401    0.052
  3     3.300    0.550    0.386    0.044
  4     2.056    0.343    0.479    0.082
  5     4.375    0.547    0.387    0.032
  6     2.375    0.792    0.454    0.139
  7     4.500    0.500    0.424    0.097
  8     1.750    0.292    0.514    0.079
  9     2.750    0.458    0.436    0.101
 10     1.786    0.357    0.486    0.072
```

Figure 9.8: *Structural holes results for the Knoke information exchange network*

- *Dyadic redundancy* means that ego's tie to alter is "redundant". If A is tied to both B and C, and B is tied to C (as in Figure 9.5), A's tie to B is redundant, because A can influence B by way of C. The dyadic redundancy measure calculates, for each actor in ego's neighborhood, how many of the other actors in the neighborhood are also tied to the other. The larger the proportion of others in the neighborhood who are tied to a give "alter", the more "redundant" is ego's direct tie. In the example, we see that actor 1's (COUN) tie to actor 2 (COMM) is largely redundant, as 72% of ego's other neighbors also have ties with COMM. Actors that display high dyadic redundancy are actors who are embedded in local neighborhoods where there are few structural holes.
- *Dyadic constraint* is a measure that indexes the extent to which the relationship between ego and each of the alters in ego's neighborhood "constrains" ego. A full description is given in Burt's 1992 monograph, and the construction of the measure is

somewhat complex. At the core though, A is constrained by its relationship with B to the extent that A does not have many alternatives (has few other ties except that to B), and A's other alternatives are also tied to B. If A has few alternatives to exchanging with B, and if those alternative exchange partners are also tied to B, then B is likely to constrain A's behavior. In our example constraint measures are not very large, as most actors have several ties. COMM and MAYR are, however, exerting considerable numbers of ties, and many of the actors to whom they are tied do not have many independent sources of information.

- *Effective size of the network* (EffSize) is the number of alters that ego has, minus the average number of ties that each alter has to other alters. Suppose that A has ties to three other actors. Suppose that none of these three has ties to any of the others. The effective size of ego's network is three. Alternatively, suppose that A has ties to three others, and that all of the others are tied to one another. A's network size is three, but the ties are "redundant" because A can reach all three neighbors by reaching any one of them. The average degree of the others in this case is 2 (each alter is tied to two other alters). So, the effective size of the network is its actual size (3), reduced by its redundancy (2), to yield an effective size of 1.
- *Efficiency* (Efficie) norms the effective size of ego's network by its actual size. That is, what proportion of ego's ties to its neighborhood are "non-redundant". The effective size of ego's network may tell us something about ego's total impact; efficiency tells us how much impact ego is getting for each unit invested in using ties. An actor can be effective without being efficient; and an actor can be efficient without being effective.
- *Constraint* (Constra) is a summary measure that taps the extent to which ego's connections are to others who are connected to one another. If ego's potential trading partners all have one another as potential trading partners, ego is highly constrained. If ego's partners do not have other alternatives in the neighborhood, they cannot constrain ego's behavior. The logic is pretty simple, but the measure itself is not. It would be good to take a look at Burt's 1992 Structural Holes. The idea of constraint is an important one because it points out that actors who have many ties to others may actually lose freedom of action rather than gain it - depending on the relationships among the other actors.
- *Hierarchy* (Hierarc) is another quite complex measure that describes the nature of the constraint on ego. If the total constraint on ego is concentrated in a single other actor, the hierarchy measure will have a higher value. If the constraint results more equally from multiple actors in ego's neighborhood, hierarchy will be less. The hierarchy measure, in itself, does not assess the degree of constraint. But, among whatever constraint there is on ego, it measures the important property of dependency - inequality in the distribution of constraints on ego across the alters in its neighborhood.

9.5: Brokerage

Burt's approach to understanding how the way that an actor is embedded in its neighborhood is very useful in understanding power, influence, and dependency effects. We'll examine some similar ideas in the chapter on centrality. Burt's underlying approach is that of the rational individual actor who may be attempting to maximize profit or advantage by modifying the way in which they are embedded. The perspective is decidedly "neo-classical".

Fernandez and Gould also examined the ways in which actor's embedding might constrain their behavior. These authors, though, took a quite different approach; they focus on the roles that ego plays in connecting groups. That is, Fernandez and Gould's "brokerage" notions examine ego's relations with its neighborhood from the perspective of ego acting as an agent in relations among groups (though, as a practical matter, the groups in brokerage analysis can be individuals).

To examine the brokerage roles played by a given actor, we find every instance where that actor lies on the directed path between two others. So, each actor may have many opportunities to act as a "broker". For each one of the instances where ego is a "broker", we examine which *kinds* of actors are involved. That is, what are the group memberships of each of the three actors? There are five possible combinations.

In Figure 9.9, the ego who is "brokering" (node B), and both the source and destination nodes (A and C) are all members of the same group. In this case, B is acting as a "coordinator" of actors within the same group as itself.

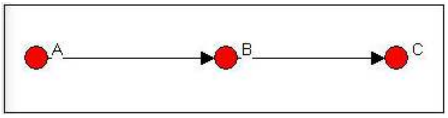


Figure 9.9: *Ego B as "coordinator"*

In Figure 9.10, ego B is brokering a relation between two members of the same group, but is not itself a member of that group. This is called a "consulting" brokerage role.

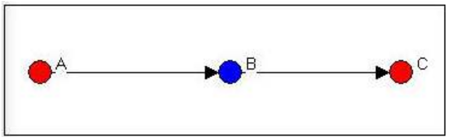


Figure 9.10: *Ego B as "consultant"*

In Figure 9.11, ego B is acting as a gatekeeper. B is a member of a group who is at its boundary, and controls access of outsiders (A) to the group.

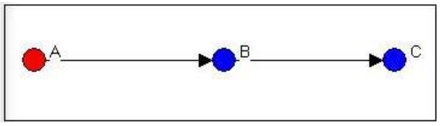


Figure 9.11: *Ego B as "gatekeeper"*

In Figure 9.12, ego B is in the same group as A, and acts as the contact point or representative of the red group to the blue.

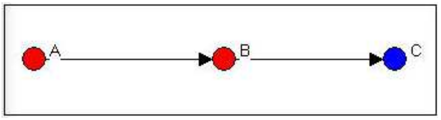


Figure 9.12: *Ego B as "representative"*

Lastly, in Figure 9.13, ego B is brokering a relation between two groups, and is not part of either. This relation is called acting as a "liaison".

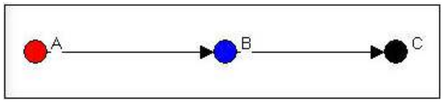
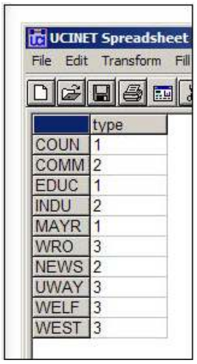


Figure 9.13: *Ego B as "liaison"*

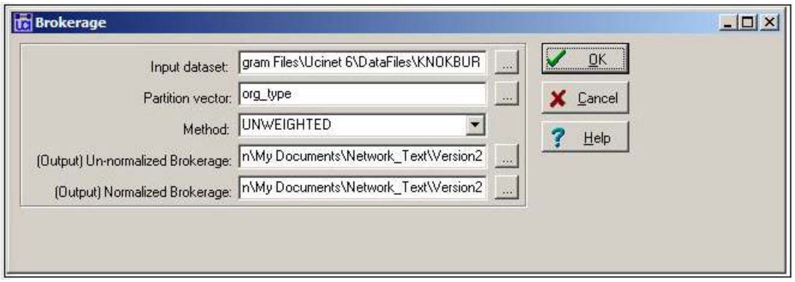
To examine brokerage, you need to create an attribute file that identifies which actor is part of which group. You can select one of the attributes from a user-created attribute file, or use output files from other UCINET routines that store descriptors of nodes as attributes. As an example, we've taken the Knoke information exchange network, and classified each of the organizations as either a general governmental organization (coded 1), a private non-welfare organization (coded 2), or an organizational specialist (coded 3). Figure 9.14 shows the attribute (or partition) as we created it using the UCINET spreadsheet editor.



	type
COUN	1
COMM	2
EDUC	1
INDU	2
MAYR	1
WRO	3
NEWS	2
UWAY	3
WELF	3
WEST	3

Figure 9.14: *Partition vector for Knoke information exchange*

Using the network dataset and the attribute vector we just created, we can run *Network>Ego Networks>Brokerage*, as shown in Figure 9.15.



Input dataset: gram Files\Ucinet 6\DataFiles\KNOKBUR

Partition vector: org_type

Method: UNWEIGHTED

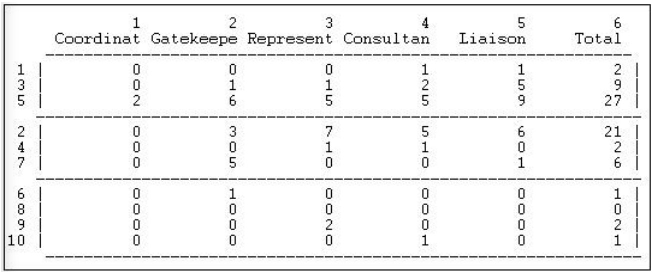
(Output) Un-normalized Brokerage: n\My Documents\Network_Text\Version2

(Output) Normalized Brokerage: n\My Documents\Network_Text\Version2

Figure 9.15: *Network>Ego Networks>Brokerage dialog box for Knoke information exchange*

The option "unweighted" needs a little explanation. Suppose that actor B was brokering a relation between actors A and C, and was acting as a "liaison". In the unweighted approach, this would count as one such relation for actor B. But, suppose that there was some other actor D who also was acting as a liaison between A and C. In the "weighted" approach, both B and D would get 1/2 of the credit for this role; in the unweighted approach, both B and D would get full credit. Generally, if we are interested in ego's relations, the unweighted approach would be used. If we were more interested in group relations, a weighted approach might be a better choice.

The output produced by *Network>Ego Networks>Brokerage* is quite extensive. We'll break it up into a few parts and discuss them separately. The first piece of the output (Figure 9.16) is a census of the number of times that each actor serves in each of the five roles.

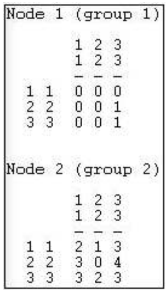


```
              1          2          3          4          5          6
       Coordinat Gatekeepe Represent Consultan    Liaison      Total
     ------------------------------------------------------------
  1 |          0          0          0          1          1          2 |
  3 |          0          1          1          2          5          9 |
  5 |          2          6          5          5          9         27 |
   -------------------------------------------------------------------
  2 |          0          3          7          5          6         21 |
  4 |          0          0          1          1          0          2 |
  7 |          0          5          0          0          1          6 |
   -------------------------------------------------------------------
  6 |          0          1          0          0          0          1 |
  8 |          0          0          0          0          0          0 |
  9 |          0          0          2          0          0          2 |
 10 |          0          0          0          1          0          1 |
     ------------------------------------------------------------
```

Figure 9.16: *Unnormalized brokerage scores for Knoke information network*

The actors have been grouped together into "partitions" for presentation; actors 1, 3, and 5, for example, form the first type of organization. Each row counts the raw number of times that each actor plays each of the five roles in the whole graph. Two actors (5 and 2) are the main sources of interconnection among the three organizational populations. Organizations in the third population (6, 8, 9, 10), the welfare specialists, have overall low rates of brokerage. Organizations in the first population (1, 3, 5), the government organizations, seem to be more heavily involved in liaison than other roles. Organizations in the second population (2, 4, 7), non-governmental generalists, play more diverse roles. Overall, there is very little coordination within each of the populations.

We might also be interested in how frequently each actor is involved in relations among and within each of the groups. Figure 9.17 shows these results for the first two nodes.



```
Node 1 (group 1)

        1 2 3
        1 2 3
        - - -
  1 1   0 0 0
  2 2   0 0 1
  3 3   0 0 1

Node 2 (group 2)

        1 2 3
        1 2 3
        - - -
  1 1   2 1 3
  2 2   3 0 4
  3 3   3 2 3
```

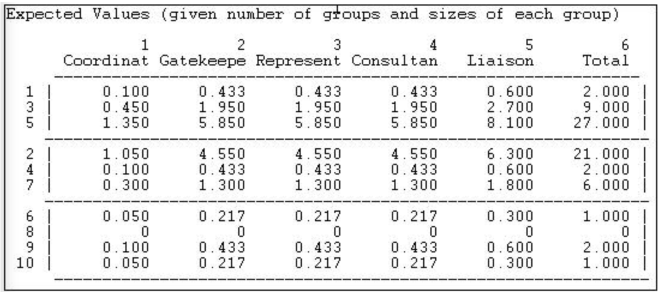
Figure 9.17: *Group-to-group brokerage map*

We see that actor 1 (who is in group 1) plays no role in connections from group 1 to itself or the other groups (i.e. the zero entries in the first row of the matrix). Actor 1 does, however, act a a "liaison" in making a connection from group 2 to group 3. Actor 1 also acts as a "consultant" in connecting a member of group 3 to another member of group 3. The very active actor 2 does not broker relations within group 2, but is heavily involved in ties in both directions of all three groups to one another, and relations among members of groups 1 and 3.

These two descriptive maps can be quite useful in characterizing the "role" that each ego is playing in the relations among groups by way of their inclusion in its local neighborhood. These roles may help us to understand how each ego may have opportunities and constraints in access to the resources of the social capital of groups as well as individuals. The overall maps also inform us about the degree and form of cohesion within and between the groups.

There may be some danger of "over-interpreting" the information about individuals' brokerage roles as representing meaningful acts of "agency". In any population in which there are connections, partitioning will produce brokerage - even if the partitions are not meaningful, or even completely random. Can we have any confidence that the patterns we are seeing in real data are actually different from a random result?

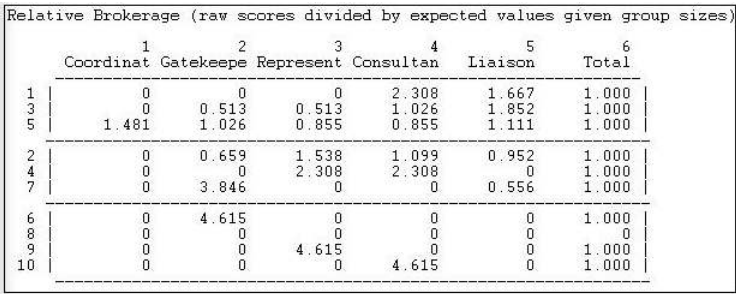
In Figure 9.18, we see the number of relations of each type that would be expected by pure random processes. We ask: What if actors were assigned to groups as we specify, and each actor has the same number of ties to other actors that we actually observe; but, the ties are distributed at random across the available actors? What if the pattern of roles was generated entirely by the number of groups of various sizes, rather than representing efforts by the actors to deliberately construct their neighborhoods to deal with the constraints and opportunities of group relations?



```
Expected Values (given number of groups and sizes of each group)

                 1          2          3          4          5          6
          Coordinat Gatekeepe Represent Consultan    Liaison      Total
        ------------------------------------------------------------------
   1 |       0.100      0.433      0.433      0.433      0.600      2.000 |
   3 |       0.450      1.950      1.950      1.950      2.700      9.000 |
   5 |       1.350      5.850      5.850      5.850      8.100     27.000 |
      -------------------------------------------------------------------
   2 |       1.050      4.550      4.550      4.550      6.300     21.000 |
   4 |       0.100      0.433      0.433      0.433      0.600      2.000 |
   7 |       0.300      1.300      1.300      1.300      1.800      6.000 |
      -------------------------------------------------------------------
   6 |       0.050      0.217      0.217      0.217      0.300      1.000 |
   8 |           0          0          0          0          0          0 |
   9 |       0.100      0.433      0.433      0.433      0.600      2.000 |
  10 |       0.050      0.217      0.217      0.217      0.300      1.000 |
        ------------------------------------------------------------------
```

Figure 9.18: *Expected values under random assignment*

If we examine the actual brokerage relative to this random expectation, we can get a better sense of which parts of which actors' roles are "significant". That is, occur much more frequently that we would expect in a world characterized by groups, but random relations among them.



```
Relative Brokerage (raw scores divided by expected values given group sizes)

                 1          2          3          4          5          6
          Coordinat Gatekeepe Represent Consultan    Liaison      Total
        ------------------------------------------------------------------
   1 |          0          0          0      2.308      1.667      1.000 |
   3 |          0      0.513      0.513      1.026      1.852      1.000 |
   5 |      1.481      1.026      0.855      0.855      1.111      1.000 |
     ---------------------------------------------------------------------
   2 |          0      0.659      1.538      1.099      0.952      1.000 |
   4 |          0          0      2.308      2.308          0      1.000 |
   7 |          0      3.846          0          0      0.556      1.000 |
     ---------------------------------------------------------------------
   6 |          0      4.615          0          0          0      1.000 |
   8 |          0          0          0          0          0          0 |
   9 |          0          0      4.615          0          0      1.000 |
  10 |          0          0          0      4.615          0      1.000 |
        ------------------------------------------------------------------
```

Figure 9.19: *Normalized brokerage scores*

The normalized brokerage scores in this example need to be treated with a little caution. As with most "statistical" approaches, larger samples (more actors) produce more stable and meaningful results. Since our network does not contain large numbers of relations, and does not have high density, there are many cases where the expected number of relations is small, and finding no such relations empirically is not surprising. Both actor 2 and actor 5, who do broker many relations, do not have profiles that differ greatly from what we would expect by chance. The lack of large deviations from expected values suggests that we might want to have a good bit of caution in interpreting our seemingly interesting descriptive data as being highly "significant".

9.S: Ego Networks (Summary)

In this chapter we've taken another look at the notion of embedding; this time, our focus has been on the individual actor, rather than the network as a whole.

The fundamental idea here is that the ways in which individuals are attached to macro-structure is often by way of their local connections. It is the local connections that most directly constrain actors, and provide them with access to opportunities. Examining the ego-networks of individuals can provide insight into why one individual's perceptions, identity, and behavior differ from another's. Looking at the demography of ego networks in a whole population can tell us a good bit about its differentiation and cohesion - from a micro point of view.

In the next several chapters we will examine additional concepts and algorithms that have been developed in social network analysis to describe important dimensions of the ways in which individuals and structures interact. We'll start with one of the most important, but also most troublesome, concepts: power.

CHAPTER OVERVIEW

10: Centrality and Power

Social network analysis methods provide some useful tools for addressing one of the most important (but also one of the most complex and difficult) aspects of social structure: the sources and distribution of power. The network perspective suggests that the power of individual actors is not an individual attribute, but arises from their relations with others. Whole social structures may also be seen as displaying high levels or low levels of power as a result of variations in the patterns of ties among actors. And, the degree of inequality or concentration of power in a population may be indexed.

10.1: Introduction to Centrality and Power

The Several Faces of Power

All sociologists would agree that power is a fundamental property of social structures. There is much less agreement about what power is, and how we can describe and analyze its causes and consequences. In this chapter, we will look at some of the main approaches that social network analysis has developed to study power, and the closely related concept of centrality.

Network training has contributed a number of important insights about social power. Perhaps more importantly, the network approach emphasizes that power is inherently relational. An individual does not have power in the abstract, they have power because they can dominate others - ego's power is alter's dependence. Because power is a consequence of patterns of relations, the amount of power in social structures can vary. If a system is very loosely coupled (low density), not much power can be exerted; in high density systems, there is the potential for greater power. Power is both a systemic (macro) and relational (micro) property. The amount of power in a system and its distribution across actors are related, but are not the same thing. Two systems can have the same amount of power, but it can be equally distributed in one and unequally distributed in another. Power in social networks may be viewed either as a micro property (i.e. it describes relations between actors), or as a macro property (i.e. one that describes the entire population); as with other key sociological concepts, the macro and micro are closely connected in social network thinking.

Network analysts often describe the way that an actor is embedded in a relational network as imposing constraints on the actor, and offering the actor opportunities. Actors that face fewer constraints, and have more opportunities than others are in favorable structural positions. Having a favored position means that an actor may extract better bargains in exchanges, have greater influence, and that the actor will be a focus for deference and attention from those in less favored positions.

But, what do we mean by "having a favored position" and having "more opportunities" and "fewer constraints"? There are no single correct and final answers to these difficult questions. But, network analysis has made important contributions in providing precise definitions and concrete measures of several different approaches to the notion of the power that attaches to positions in structures of social relations.

To understand the approaches that network analysis uses to study power, it is useful to first think about some very simple systems. Consider the three simple graphs of networks in Figures 10.1, 10.2, and 10.3, which are called the "star", "line", and "circle".

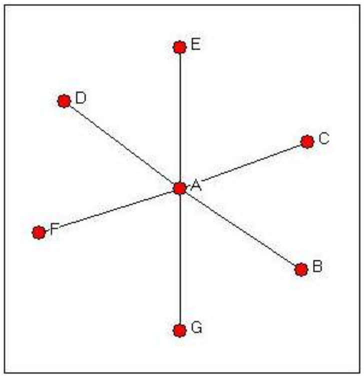


Figure 10.1: *"Star" network*

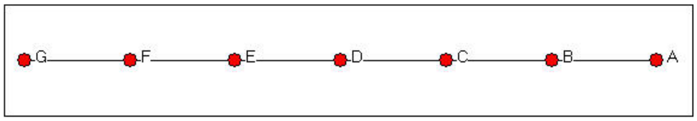


Figure 10.2: *"Line" network*

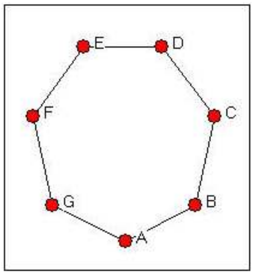


Figure 10.3: *"Circle" network*

A moment's inspection ought to suggest that actor A has a highly favored structural position in the star network, if the network is describing a relationship such as resource exchange or resource sharing. But, exactly why is it that actor A has a "better" position than all of the others in the star network? What about the position of A in the line network? Is being at the end of the line an advantage or a disadvantage? Are all of the actors in the circle network really in exactly the same structural position?

We need to think about why structural location can be advantageous or disadvantageous to actors. Let's focus our attention on why actor A is so obviously at an advantage in the star network.

Degree: In the star network, actor A has more opportunities and alternatives than other actors. If actor D elects to not provide A with a resource, A has a number of other places to go to get it; however, if D elects to not exchange with A, then D will not be able to exchange at all. The more ties an actor has, the more power they (may) have. In the star network, Actor A has degree six, all other actors have degree one. This logic underlies measures of centrality and power based on *actor degree*, which we will discuss below. Actors who have more ties have greater opportunities because they have choices. This autonomy makes them less dependent on any specific other actor, and hence more powerful.

Now, consider the circle network in terms of degree. Each actor has exactly the same number of alternative trading partners (or degree), so all positions are equally advantaged or disadvantaged.

In the line network, matters are a bit more complicated. The actors at the end of the line (A and G) are actually at a structural disadvantage, but all others are apparently equal (actually, it's not really quite that simple). Generally, though, actors that are more central to the structure, in the sense of having higher degree or more connections, tend to have favored positions, and hence more power.

Closeness: The second reason why actor A is more powerful than the other actors in the start network is that actor A is *closer* to more actors than any other actor. Power can be exerted by direct bargaining and exchange. But power also comes from acting as a "reference point" by which other actors judge themselves, and by being a center of attention who's views are heard by larger numbers of actors. Actors who are able to reach other actors at shorter path lengths, or who are more reachable by other actors at shorter path lengths have favored positions. This structural advantage can be translated into power. In the star network, actor A is at a geodesic distance of one from all other actors; each other actor is at a geodesic distance of two from all other actors (but A). This logic of structural advantage underlies approaches that emphasize the distribution of closeness and distance as a source of power.

Now consider the circle network in terms of actor closeness. Each actor lies at different path lengths from the other actors, but all actors have identical distributions of closeness, and again would appear to be equal in terms of their structural positions. In the line network, the middle actor (D) is closer to all other actors than are the set C, E, the set B, F, and the set A, G. Again, the actors at the ends of the line, or at the periphery, are at a disadvantage.

Betweenness: The third reason that actor A is advantaged in the star network is because actor A lies *between* each other pairs of actors, and no other actors lie between A and other actors. If A wants to contact F, A may simply do so. If F wants to contact B, they must do so by way of A. This gives actor A the capacity to broker contacts among other actors - to extract "service charges" and to isolate actors or prevent contacts. The third aspect of a structurally advantaged position then is in being between other actors.

In the circle network, each actor lies between each other pair of actors. Actually, there are two pathways connecting each pair of actors, and each third actor lies on one, but not on the other of them. Again, all actors are equally advantaged or disadvantaged. In the line network, our end points (A, G) do not lie between any pairs, and have no brokering power. Actors closer to the middle of the chain lie on more pathways among pairs, and are again in an advantaged position.

Each of these three ideas - degree, closeness, and betweenness - has been elaborated on in a number of ways. We will examine three such elaborations briefly here.

Network analysts are more likely to describe their approaches as descriptions of centrality than of power. Each of the three approaches (degree, closeness, betweenness) describe the locations of individuals in terms of how close they are to the "center" of the action in a network - though the definitions of what it means to be at the center differ. It is more correct to describe network approaches this way - measures of centrality - than as measures of power. But, as we have suggested here, there are several reasons why central positions tend to be powerful positions.

10.2: Degree Centrality

Actors who have more ties to other actors may be in advantaged positions. Because they have many ties, they may have alternative ways to satisfy needs, and hence are less dependent on other individuals. Because they have many ties, they may have access to, and be able to call on, more of the resources of the network as a whole. Because they have many ties, they are often third parties and deal makers in exchanges among others, and are able to benefit from this brokerage. So, a very simple, but often very effective measure of an actor's centrality and power potential is their degree.

In undirected data, actors differ from one another only in how many connections they have. With directed data, however, it can be important to distinguish centrality based on in-degree from centrality based on out-degree. If an actor receives many ties, they are often said to be *prominent*, or to have high *prestige*. That is, many other actors seek to direct ties to them, and this may indicate their importance. Actors who have unusually high out-degree are actors who are able to exchange with many others, or make many others aware of their views. Actors who display high out-degree centrality are often said to be *influential* actors.

Recall Knoke's data on information exchanges among organizations operating in the social welfare field, shown in Figure 10.4.

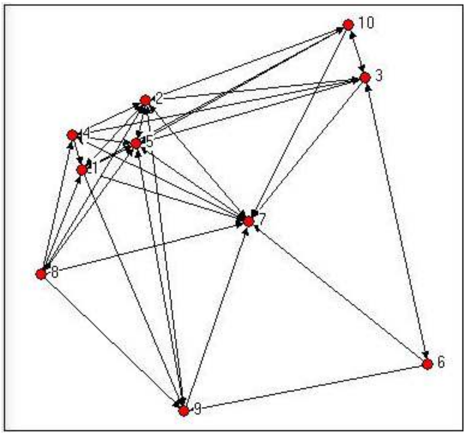


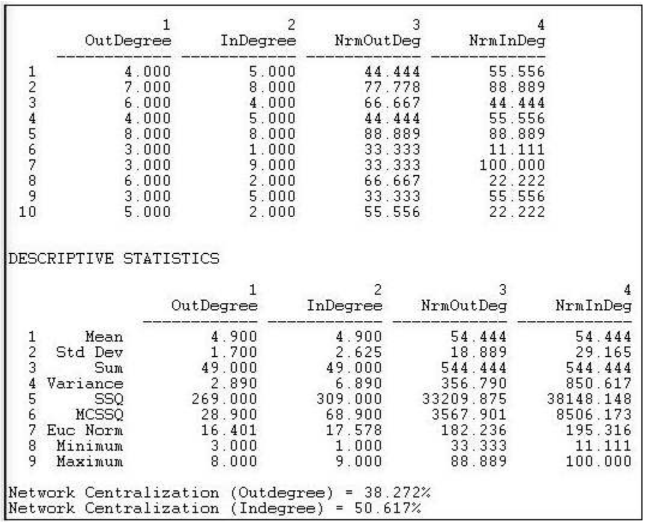
Figure 10.4: *Knoke's information exchange network*

Simply counting the number of in-ties and out-ties of the nodes suggests that certain actors are more "central" here (e.g. 2, 5, 7). It also appears that this network as a whole may have a group of central actors, rather than a single "star". We can see "centrality" as an attribute of individual actors as a consequence of their position; we can also see how "centralized" the graph as a whole is - how unequal is the distribution of centrality.

Freeman's Approach

Linton Freeman (one of the authors of UCINET) developed basic measures of the centrality of actors based on their degree, and the overall centralization of graphs.

Figure 10.5 shows the output of *Network>Centrality>Degree* applied to out-degrees and to the in-degrees of the Knoke information network. The centrality can also be computed ignoring the direction of ties (i.e. a tie in either direction is counted as a tie).



```
              1            2            3            4
      OutDegree     InDegree    NrmOutDeg     NrmInDeg
   ------------ ------------ ------------ ------------
 1        4.000        5.000       44.444       55.556
 2        7.000        8.000       77.778       88.889
 3        6.000        4.000       66.667       44.444
 4        4.000        5.000       44.444       55.556
 5        8.000        8.000       88.889       88.889
 6        3.000        1.000       33.333       11.111
 7        3.000        9.000       33.333      100.000
 8        6.000        2.000       66.667       22.222
 9        3.000        5.000       33.333       55.556
10        5.000        2.000       55.556       22.222

DESCRIPTIVE STATISTICS

                       1            2            3            4
               OutDegree     InDegree    NrmOutDeg     NrmInDeg
            ------------ ------------ ------------ ------------
1      Mean        4.900        4.900       54.444       54.444
2   Std Dev        1.700        2.625       18.889       29.165
3       Sum       49.000       49.000      544.444      544.444
4 Variance         2.890        6.890      356.790      850.617
5       SSQ      269.000      309.000    33209.875    38148.148
6     MCSSQ       28.900       68.900     3567.901     8506.173
7 Euc Norm        16.401       17.578      182.236      195.316
8   Minimum        3.000        1.000       33.333       11.111
9   Maximum        8.000        9.000       88.889      100.000

Network Centralization (Outdegree) = 38.272%
Network Centralization (Indegree) = 50.617%
```

Figure 10.5: *Freeman degree centrality and graph centralization of Knoke information network*

Actors #5 and #2 have the greatest out-degrees, and might be regarded as the most influential (though it might matter to whom they are sending information, this measure does not take that into account). Actors #5 and #2 are joined by #7 (the newspaper) when we examine in-degree. That other organizations share information with these three would seem to indicate a desire on the part of others to exert influence. This is an act of deference, or a recognition that the positions of actors 5, 2, and 7 might be worth trying to influence. If we were interested in comparing across networks of different sizes or densities, it might be useful to "standardize" the measures of in-degree and out-degree. In the last two columns of the first panel of results above, all the degree counts have been expressed as percentages of the number of actors in the network, less one (ego).

The next panel of results speaks to the "meso" level or analysis. That is, what does the distribution of the actor's degree centrality scores look like? On the average, actors have a degree of 4.9, which is quite high, given that there are only nine other actors. We see that the range of in-degree is slightly larger (minimum and maximum) than that of out-degree, and that there is more variability across the actors in in-degree than out-degree (standard deviations and variances). The range and variability of degree (and other network properties) can be quite important, because it describes whether the population is homogeneous or heterogeneous in structural positions. One could examine whether the variability is high or low relative to the typical scores by calculating the coefficient of variation (standard deviation divided by mean, times 100) for in-degree and out-degree. By the rules of thumb that are often used to evaluate coefficients of variation, the current values (35 for out-degree and 53 for in-degree) are moderate. Clearly, however, the population is more homogeneous with regard to out-degree (influence) than with regard to in-degree (prominence).

The last bit of information provided by the output above are Freeman's *graph centralization measures*, which describe the population as a whole - the macro level. These are very useful statistics, but require a bit of explanation.

Remember our "star" network from the discussion above (if not, go review it)? The star network is the most centralized or most unequal possible network for any number of actors. In the star network, all the actors but one have degree of one, and the "star" has degree of the number of actors, less one. Freeman felt that it would be useful to express the degree of variability in the degrees of actors in our observed network as a percentage of that in a star network of the same size. This is how the Freeman graph centralization measures can be understood: they express the degree of inequality or variance in our network as a percentage of that of a perfect star network of the same size. In the current case, the out-degree graph centralization is 51% and the in-degree graph centralization 38% of these theoretical maximums. We would arrive at the conclusion that there is a substantial amount of concentration or centralization in this whole network. That is, the power of individual actors varies rather substantially, and this means that, overall, positional advantages are rather unequally distributed in this network.

Bonacich's Approach

Philip Bonacich proposed a modification of the degree centrality approach that has been widely accepted as superior to the original measure. Bonacich's idea, like most good ones, is pretty simple. The original degree centrality approach argues that actors who have more connections are more likely to be powerful because they can directly affect more other actors. This makes sense, but having the same degree does not necessarily make actors equally important.

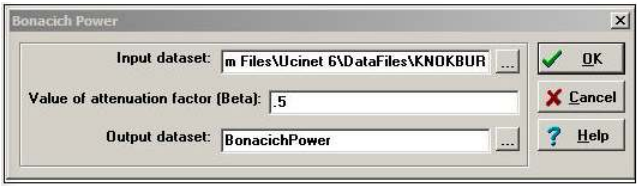
Suppose that Bill and Fred each have five close friends. Bill's friends, however, happen to be pretty isolated folks, and don't have many other friends, save Bill. In contrast, Fred's friends each also have lots of friends, who have lots of friends, and so on. Who is more central? We would probably agree that Fred is, because the people he is connected to are better connected than Bill's people. Bonacich argued that one's centrality is a function of how many connections one has, and how many of the connections the actors in the neighborhood had.

While we have argued that more central actors are more likely to be more powerful actors, Bonacich questioned this idea. Compare Bill and Fred again. Fred is clearly more central, but is he more powerful? One argument would be that one is likely to be more influential if one is connected to central others - because one can quickly reach a lot of other actors with one's message. But if the actors that you are connected to are, themselves, well connected, they are not highly dependent on you - they have many contacts, just as you do. If, on the other hand, the people to whom you are connected are not, themselves, well connected, then they are dependent on you. Bonacich argued that being connected to connected others makes an actor central, but not powerful. Somewhat ironically, being connected to others that are not well connected makes one powerful, because these other actors are dependent on you - whereas well connected actors are not.

Bonacich proposed that both centrality and power were a function of the connections of the actors in one's neighborhood. The more connections the actors in your neighborhood have, the more central you are. The fewer the connections the actors in your neighborhood, the more powerful you are. There would seem to be a problem with building an algorithm to capture these ideas. Suppose A and B are connected. Actor A's power and centrality are functions of her own connections, and also the connections of actor B. Similarly, actor B's power and centrality depend on actor A's. So, each actor's power and centrality depends on each other actor's power simultaneously.

There is a way out of this chicken-and-egg type of problem. Bonacich showed that, for symmetric systems, an iterative estimation approach to solving this simultaneous equations problem would eventually converge to a single answer. One begins by giving each actor an estimated centrality equal to their own degree, plus a weighted function of the degrees of the actors to whom they were connected. Then, we do this again, using the first estimates (i.e. we again give each actor an estimated centrality equal to their own score plus the first scores of those to whom they are connected). As we do this numerous times, the relative sizes (not the absolute sizes) of all actors' scores will come to be the same. The scores can then be re-expressed by scaling by constants.

Let's examine the centrality and power scores for our information exchange data. First, we examine the case where the score for each actor is a positive function of their own degree, and the degrees of the others to whom they are connected. We do this by selecting a positive weight of the "attenuation factor" or Beta parameter) in the dialog of *Network>Centrality>Power*, as shown in Figure 10.6.


Bonacich Power

Input dataset: m Files\Ucinet 6\DataFiles\KNOKBUR

Value of attenuation factor (Beta): .5

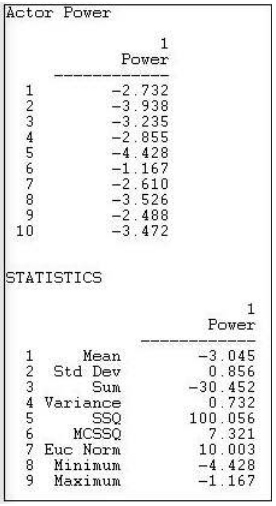
Output dataset: BonacichPower

Figure 10.6: *Dialog for computing Bonacich's power measures*

The "attenuation factor" indicates the effect of one's neighbor's connections on ego's power. Where the attenuation factor is positive (between zero and one), being connected to neighbors with more connections makes one powerful. This is a straightforward extension of the degree centrality idea.

Bonacich also had a second idea about power, based on the notion of "dependency". If ego has neighbors who do not have many connections to others, those neighbors are likely to be dependent on ego, making ego more powerful. Negative values of the attenuation factor (between zero and negative one) compute power based on this idea.

Figures 10.7 and 10.8 show the Bonacich measures for positive and negative beta values.

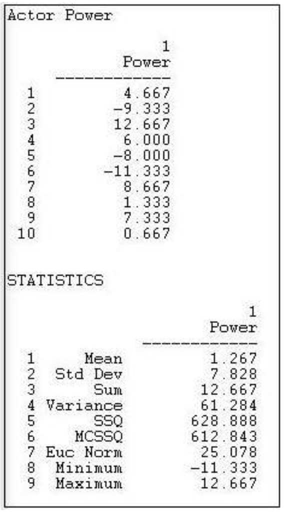


```
Actor Power

                  1
              Power
       ------------
  1          -2.732
  2          -3.938
  3          -3.235
  4          -2.855
  5          -4.428
  6          -1.167
  7          -2.610
  8          -3.526
  9          -2.488
 10          -3.472

STATISTICS

                           1
                       Power
                ------------
  1      Mean         -3.045
  2   Std Dev          0.856
  3       Sum        -30.452
  4  Variance          0.732
  5       SSQ        100.056
  6     MCSSQ          7.321
  7  Euc Norm         10.003
  8   Minimum         -4.428
  9   Maximum         -1.167
```

Figure 10.7: *Network>Centrality>Power with beta = +0.50*

If we look at the absolute value of the index scores, we see the familiar story. Actors #5, and #2 are clearly the most central. This is because they have high degree, and because they are connected to each other, and to other actors with high degree. Actors 8 and 10 also appear to have high centrality by this measure - this is a new result. In these cases, it is because the actors are connected to all of the other high degree points. These actors don't have extraordinary numbers of connections, but they have "the right connections".

Let's take a look at the power side of the index, which is calculated by the same algorithm, but gives negative weights to connections with well connected others, and positive weights for connections to weakly connected others.



```
Actor Power

                  1
              Power
       ------------
  1           4.667
  2          -9.333
  3          12.667
  4           6.000
  5          -8.000
  6         -11.333
  7           8.667
  8           1.333
  9           7.333
 10           0.667

STATISTICS

                           1
                       Power
                ------------
  1      Mean          1.267
  2   Std Dev          7.828
  3       Sum         12.667
  4  Variance         61.284
  5       SSQ        628.888
  6     MCSSQ        612.843
  7  Euc Norm         25.078
  8   Minimum        -11.333
  9   Maximum         12.667
```

Figure 10.8: *Network>Centrality>Power with beta = -0.50*

Not surprisingly, these results are very different from many of the others we've examined. With a negative attenuation parameter, we have a quite different definition of power - having weak neighbors, rather than strong ones. Actors numbers 2 and 6 are distinguished because their ties are mostly ties to actors with high degree - making actors 2 and 6 "weak" by having powerful neighbors. Actors 3, 7, and 9 have more ties to neighbors who have few ties - making them "strong" by having weak neighbors. You might want to scan the diagram again to see if you can see these differences.

The Bonacich approach to degree-based centrality and degree-based power are fairly natural extensions of the idea of degree centrality based on adjacencies. One is simply taking into account the connections of one's connections, in addition to one's own connections. The notion that power arises from connection to weak others, as opposed to strong others is an interesting one, and points to yet another way in which the positions of actors in network structures endow them with different potentials.

10.3: Closeness Centrality

Degree centrality measures might be criticized because they only take into account the immediate ties that an actor has, or the ties of the actor's neighbors, rather than indirect ties to all others. One actor might be tied to a large number of others, but those others might be rather disconnected from the network as a whole. In a case like this, the actor could be quite central, but only in a local neighborhood.

Closeness centrality approaches emphasize the distance of an actor to all others in the network by focusing on the distance from each actor to all others. Depending on how one wants to think of what it means to be "close" to others, a number of slightly different measures can be defined.

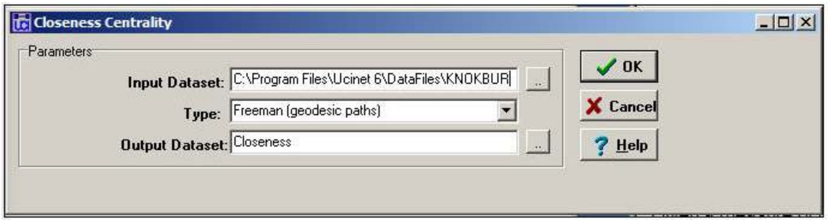
Path Distances

Network>Centrality>Closeness provides a number of alternative ways of calculating the "far-ness" of each actor from all others. Far-ness is the sum of the distance (by various approaches) from each ego to all others in the network.

"Far-ness" is then transformed into "nearness" as the reciprocal of far-ness. That is, nearness = one divided by far-ness. "Nearness" can be further standardized by norming against the minimum possible nearness for a graph of the same size and connection.

Given a measure of nearness or far-ness for each actor, we can again calculate a measure of inequality in the distribution of distances across the actors, and express "graph centralization" relative to that of the idealized "star" network.

Figure 10.9 shows a dialog for calculating closeness measures of centrality and graph centralization.



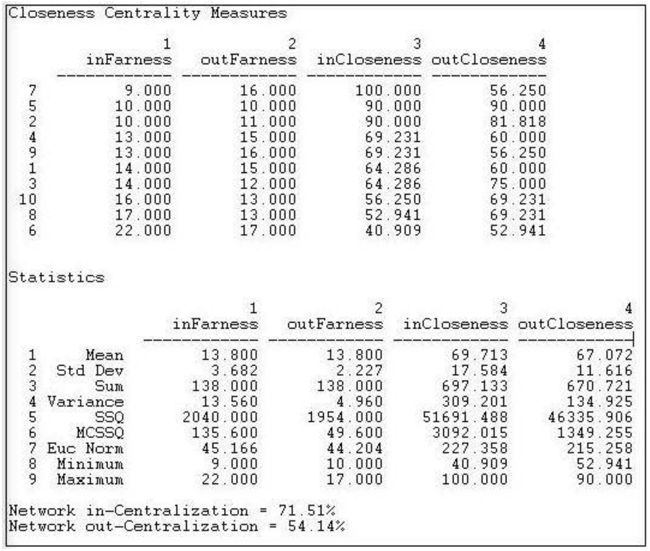
Input Dataset: C:\Program Files\Ucinet 6\DataFiles\KNOKBUR

Type: Freeman (geodesic paths)

Output Dataset: Closeness

Figure 10.9: *Dialog for Network>Centrality>Closeness*

Several alternative approaches to measuring "far-ness" are available in the *type* setting. The most common is probably the *geodesic path* distance. Here, "far-ness" is the sum of the lengths of the shortest paths from ego (or to ego) from all other nodes. Alternatively, the *reciprocal* of this, or "nearness" can be calculated. Alternatively, one may focus on *all paths*, not just geodesics or *all trails*. Figure 10.10 shows the results for the Freeman geodesic path approach.



```
Closeness Centrality Measures

                1            2            3            4
        inFarness   outFarness  inCloseness outCloseness
     ------------ ------------ ------------ ------------
  7         9.000       16.000      100.000       56.250
  5        10.000       10.000       90.000       90.000
  2        10.000       11.000       90.000       81.818
  4        13.000       15.000       69.231       60.000
  9        13.000       16.000       69.231       56.250
  1        14.000       15.000       64.286       60.000
  3        14.000       12.000       64.286       75.000
 10        16.000       13.000       56.250       69.231
  8        17.000       13.000       52.941       69.231
  6        22.000       17.000       40.909       52.941


Statistics

                           1            2            3            4
                   inFarness   outFarness  inCloseness outCloseness
                ------------ ------------ ------------ ------------
  1      Mean         13.800       13.800       69.713       67.072
  2   Std Dev          3.682        2.227       17.584       11.616
  3       Sum        138.000      138.000      697.133      670.721
  4  Variance         13.560        4.960      309.201      134.925
  5       SSQ       2040.000     1954.000    51691.488    46335.906
  6     MCSSQ        135.600       49.600     3092.015     1349.255
  7  Euc Norm         45.166       44.204      227.358      215.258
  8   Minimum          9.000       10.000       40.909       52.941
  9   Maximum         22.000       17.000      100.000       90.000

Network in-Centralization = 71.51%
Network out-Centralization = 54.14%
```

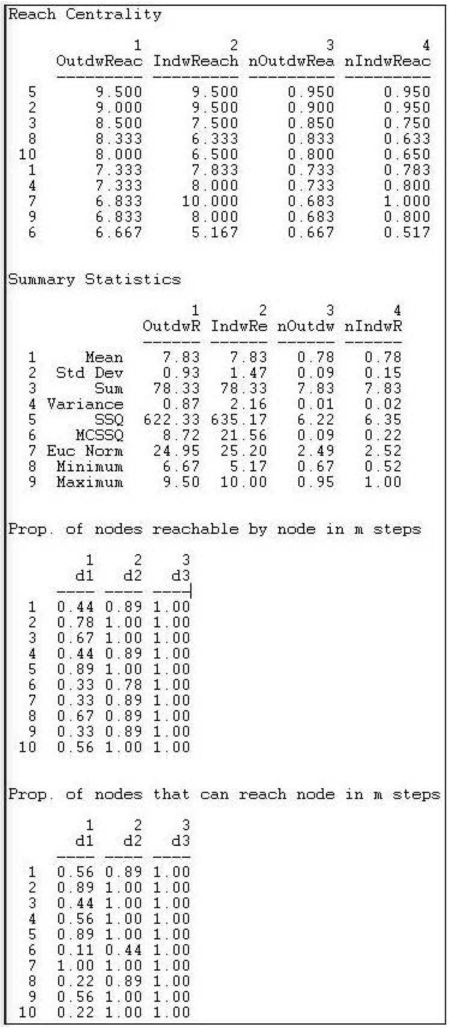
Figure 10.10: *Geodesic path closeness centrality for Knoke information network*

Since the information network is directed, separate closeness and far-ness can be computed for sending and receiving. We see that actor 6 has the largest sum of geodesic distances from other actors (in-far-ness of 22) and to other actors (out-far-ness of 17). The far-ness figures can be re-expressed as nearness (the reciprocal of far-ness) and normed relative to the greatest nearness observed in the graph (here, the in-closeness of actor 7).

Summary statistics on the distribution of the nearness and far-ness measures are also calculated. We see that the distribution of out-closeness has less variability than in-closeness, for example. This is also reflected in the graph in-centralization (71.5%) and out-centralization (54.1%) measures; that is, in-distances are more unequally distributed than are out-distances.

Reach

Another way of thinking about how close an actor is to all others is to ask what portion of all others ego can reach in one step, two steps, three steps, etc. The routine *Network>Centrality>Reach Centrality* calculates some useful measures of how close each actor is to all others. Figure 10.11 shows the results for the Knoke information network.



```
Reach Centrality

                1          2          3          4
        OutdwReac  IndwReach nOutdwRea nIndwReac
        --------- ---------- --------- ---------
  5         9.500      9.500     0.950     0.950
  2         9.000      9.500     0.900     0.950
  3         8.500      7.500     0.850     0.750
  8         8.333      6.333     0.833     0.633
 10         8.000      6.500     0.800     0.650
  1         7.333      7.833     0.733     0.783
  4         7.333      8.000     0.733     0.800
  7         6.833     10.000     0.683     1.000
  9         6.833      8.000     0.683     0.800
  6         6.667      5.167     0.667     0.517

Summary Statistics

                    1      2      3      4
               OutdwR IndwRe nOutdw nIndwR
               ------ ------ ------ ------
  1      Mean    7.83   7.83   0.78   0.78
  2   Std Dev    0.93   1.47   0.09   0.15
  3       Sum   78.33  78.33   7.83   7.83
  4 Variance     0.87   2.16   0.01   0.02
  5       SSQ  622.33 635.17   6.22   6.35
  6     MCSSQ    8.72  21.56   0.09   0.22
  7 Euc Norm    24.95  25.20   2.49   2.52
  8  Minimum     6.67   5.17   0.67   0.52
  9  Maximum     9.50  10.00   0.95   1.00

Prop. of nodes reachable by node in m steps

        1    2    3
       d1   d2   d3
     ---- ---- ----
  1  0.44 0.89 1.00
  2  0.78 1.00 1.00
  3  0.67 1.00 1.00
  4  0.44 0.89 1.00
  5  0.89 1.00 1.00
  6  0.33 0.78 1.00
  7  0.33 0.89 1.00
  8  0.67 0.89 1.00
  9  0.33 0.89 1.00
 10  0.56 1.00 1.00

Prop. of nodes that can reach node in m steps

        1    2    3
       d1   d2   d3
     ---- ---- ----
  1  0.56 0.89 1.00
  2  0.89 1.00 1.00
  3  0.44 1.00 1.00
  4  0.56 1.00 1.00
  5  0.89 1.00 1.00
  6  0.11 0.44 1.00
  7  1.00 1.00 1.00
  8  0.22 0.89 1.00
  9  0.56 1.00 1.00
 10  0.22 1.00 1.00
```

Figure 10.11: *Reach centrality for Knoke information network*

An index of the "reach distance" from each ego to (or from) all others is calculated. Here, the maximum score (equal to the number of nodes) is achieved when every other is one step from ego. The reach closeness sum becomes less as actors are two steps, three steps, and so on (weights of 1/2, 1/3, etc.). These scores are then expressed in "normed" form by dividing by the largest observed reach value.

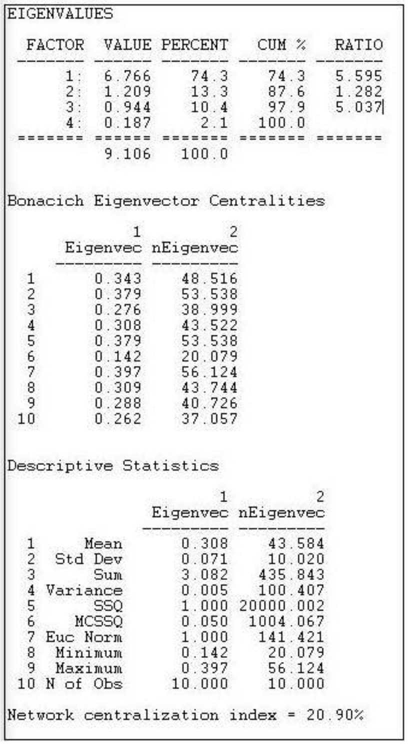
The final two tables are quite easy to interpret. The first of these shows what proportion of other nodes can be reached from each actor at one, two, and three steps (in our example, all others are reachable in three steps or less). The last table shows what proportions of others can reach ego at one, two, and three steps. Note that everyone can contact the newspaper (actor 7) in one step.

Eigenvector of Geodesic Distance

The closeness centrality measure described above is based on the sum of the geodesic distances from each actor to all others (far-ness). In larger and more complex networks than the example we've been considering, it is possible to be somewhat misled by this measure. Consider two actors, A and B. Actor A is quite close to a small and fairly closed group within a larger network, and rather distant from many of the members of the population. Actor B is at a moderate distance from all of the members of the population. The far-ness measures for actor A and actor B could be quite similar to magnitude. In a sense, however, actor B is really more "central" than actor A in this example, because B is able to reach more of the network with the same amount of effort.

The eigenvector approach is an effort to find the most central actors (i.e. those with the smallest far-ness from others) in terms of the "global" or "overall" structure of the network, and to pay less attention to patterns that are more "local". The method used to do this (factor analysis) is beyond the scope of the current text. In a general way, what factor analysis does is to identify "dimensions" of the distances among actors. The location of each actor with respect to each dimension is called an "eigenvalue", and the collection of such values is called the "eigenvector". Usually, the first dimension captures the "global" aspects of distances among actors; second and further dimensions capture more specific and local substructures.

The UCINET *Network>Centrality>Eigenvector* routine calculates individual actor centrality, and graph centralization using weights on the first eigenvector. A limitation of the routine is that it does not calculate values for asymmetric data. So, our measures here are based on the notion of "any connection".



```
EIGENVALUES

  FACTOR  VALUE PERCENT   CUM %   RATIO
 ------- ------ ------- ------- -------
      1:  6.766    74.3    74.3   5.595
      2:  1.209    13.3    87.6   1.282
      3:  0.944    10.4    97.9   5.037
      4:  0.187     2.1   100.0
 ======= ====== ======= ======= =======
          9.106   100.0


Bonacich Eigenvector Centralities

                1          2
         Eigenvec nEigenvec
        --------- ---------
  1         0.343    48.516
  2         0.379    53.538
  3         0.276    38.999
  4         0.308    43.522
  5         0.379    53.538
  6         0.142    20.079
  7         0.397    56.124
  8         0.309    43.744
  9         0.288    40.726
 10         0.262    37.057


Descriptive Statistics

                         1          2
                  Eigenvec nEigenvec
                 --------- ---------
  1      Mean        0.308    43.584
  2   Std Dev        0.071    10.020
  3       Sum        3.082   435.843
  4  Variance        0.005   100.407
  5       SSQ        1.000 20000.002
  6     MCSSQ        0.050  1004.067
  7  Euc Norm        1.000   141.421
  8   Minimum        0.142    20.079
  9   Maximum        0.397    56.124
 10  N of Obs       10.000    10.000

Network centralization index = 20.90%
```

Figure 10.12: *Eigenvector centrality and centralization for Knoke information network*

The first set of statistics, the eigenvalues, tell us how much of the overall pattern of distances among actors can be seen as reflecting the global pattern (the first eigenvalue), and more local, or additional patterns. We are interested in the percentage of the overall variation in distances that is accounted for by the first factor. Here, this percentage is 74.3%. This means that about 3/4 of all of the distances among actors are reflective of the main dimension or pattern. If this amount is not large (say over 70%), great caution should be exercised in interpreting the further results, because the dominant pattern is not doing a very complete job of describing the data. The first eigenvalue should also be considerably larger than the second (here, the ratio of the first eigenvalue to the second is about 5.6 to 1). This means that the dominant pattern is, in a sense, 5.6 times as "important" as the secondary pattern.

Next, we turn our attention to the scores of each of the cases on the 1st eigenvector. Higher scores indicate that actors are "more central" to the main pattern of distances among all of the actors, lower values indicate that actors are more peripheral. The results are very similar to those for our earlier analysis of closeness centrality, with #7, #5, and #2 being most central, and actor #6 being most peripheral. Usually the eigenvalue approach will do what it is supposed to do: give us a "cleaned-up" version of the closeness centrality measures, as it does here. It is a good idea to examine both, and to compare them.

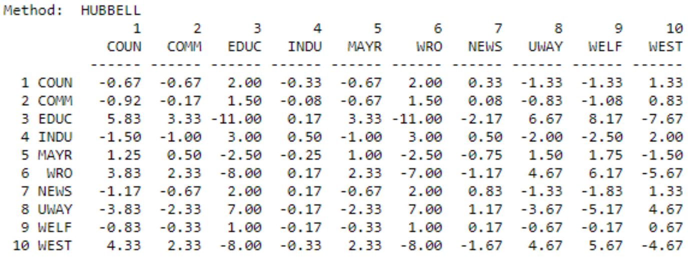
Last, we examine the overall centralization of the graph, and the distribution of centralities. There is relatively little variability in centralities (standard deviation 0.07) around the mean (0.31). This suggests that, overall, there are not great inequalities in actor centrality or power, when measured in this way. Compared to the pure "star" network, the degree of inequality or concentration of the Knoke data is only 20.9% of the maximum possible. This is much less than the network centralization measure for the "raw" closeness measure (49.3), and suggests that some of the apparent differences in power using the raw closeness approach may be due more to local than to global inequalities.

Geodesic distances among actors are a reasonable measure of one aspect of centrality - or positional advantage. Sometimes these advantages may be more local, and sometimes more global. The factor-analytic approach is one approach that may sometimes help us to focus on the more global pattern. Again, it is not that one approach is "right" and the other "wrong". Depending on the goals of our analysis, we may wish to emphasize one or the other aspects of the positional advantages that arise from centrality.

Hubbell, Katz, Taylor, Stephenson, and Zelen Influence Measures

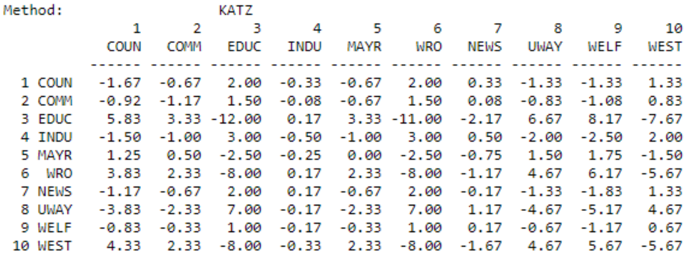
The geodesic closeness and eigenvalue approaches consider the closeness of connection to all other actors, but only by the "most efficient" path (the geodesic). In some cases, power or influence may be expressed through all of the pathways that connect an actor to all others. Several measures of closeness based on all connections of ego to others are available from *Network>Centrality>Influence*.

Even if we want to include all connections between two actors, it may not make a great deal of sense to consider a path of length 10 as important as a path of length 1. The Hubbell and Katz approaches count the total connections between actors (ties for undirected data, both sending and receiving ties for directed data). Each connection, however, is given a weight according to its length. The greater the length, the weaker the connection. How much weaker the connection becomes with increasing length depends on an "attenuation" factor. In our example, below, we have used an attenuation factor of 0.5. That is, an adjacency receives a weight of one, a walk of length two receives a weight of 0.5, a connection of length three receives a weight of 0.5 squared (0.25), etc. The Hubbell and Katz approaches are very similar. Katz includes an identity matrix (a connection of each actor with itself) as the strongest connection; the Hubbell approach does not. As calculated by UCINET, both approaches "norm" the results to range from largest negative distances (that is, the actors are very close relative to the other pairs, or have high cohesion) to large positive numbers (that is, the actors have large distance relative to others). The results of the Hubbell and Katz approaches are shown in Figures 10.13 and 10.14.



```
Method:  HUBBELL
                  1      2      3      4      5      6      7      8      9     10
               COUN   COMM   EDUC   INDU   MAYR    WRO   NEWS   UWAY   WELF   WEST
             ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1 COUN      -0.67  -0.67   2.00  -0.33  -0.67   2.00   0.33  -1.33  -1.33   1.33
  2 COMM      -0.92  -0.17   1.50  -0.08  -0.67   1.50   0.08  -0.83  -1.08   0.83
  3 EDUC       5.83   3.33 -11.00   0.17   3.33 -11.00  -2.17   6.67   8.17  -7.67
  4 INDU      -1.50  -1.00   3.00   0.50  -1.00   3.00   0.50  -2.00  -2.50   2.00
  5 MAYR       1.25   0.50  -2.50  -0.25   1.00  -2.50  -0.75   1.50   1.75  -1.50
  6  WRO       3.83   2.33  -8.00   0.17   2.33  -7.00  -1.17   4.67   6.17  -5.67
  7 NEWS      -1.17  -0.67   2.00   0.17  -0.67   2.00   0.83  -1.33  -1.83   1.33
  8 UWAY      -3.83  -2.33   7.00  -0.17  -2.33   7.00   1.17  -3.67  -5.17   4.67
  9 WELF      -0.83  -0.33   1.00  -0.17  -0.33   1.00   0.17  -0.67  -0.17   0.67
 10 WEST       4.33   2.33  -8.00  -0.33   2.33  -8.00  -1.67   4.67   5.67  -4.67
```

Figure 10.13: *Hubbell dyadic influence for the Knoke information network*



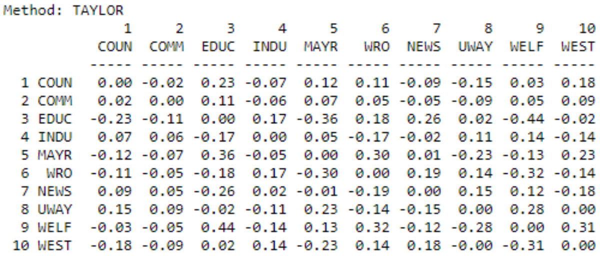
```
Method:                   KATZ
                  1      2      3      4      5      6      7      8      9     10
               COUN   COMM   EDUC   INDU   MAYR    WRO   NEWS   UWAY   WELF   WEST
             ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1 COUN      -1.67  -0.67   2.00  -0.33  -0.67   2.00   0.33  -1.33  -1.33   1.33
  2 COMM      -0.92  -1.17   1.50  -0.08  -0.67   1.50   0.08  -0.83  -1.08   0.83
  3 EDUC       5.83   3.33 -12.00   0.17   3.33 -11.00  -2.17   6.67   8.17  -7.67
  4 INDU      -1.50  -1.00   3.00  -0.50  -1.00   3.00   0.50  -2.00  -2.50   2.00
  5 MAYR       1.25   0.50  -2.50  -0.25   0.00  -2.50  -0.75   1.50   1.75  -1.50
  6  WRO       3.83   2.33  -8.00   0.17   2.33  -8.00  -1.17   4.67   6.17  -5.67
  7 NEWS      -1.17  -0.67   2.00   0.17  -0.67   2.00  -0.17  -1.33  -1.83   1.33
  8 UWAY      -3.83  -2.33   7.00  -0.17  -2.33   7.00   1.17  -4.67  -5.17   4.67
  9 WELF      -0.83  -0.33   1.00  -0.17  -0.33   1.00   0.17  -0.67  -1.17   0.67
 10 WEST       4.33   2.33  -8.00  -0.33   2.33  -8.00  -1.67   4.67   5.67  -5.67
```

Figure 10.14: *Katz dyadic influence for the Knoke information network*

As with all measures of pair-wise properties, one could analyze the data much further. We could see which individuals are similar to which others (that is, are there different groups or strata defined by the similarity of their total connections to all others in the network?). Our interest might also focus on the whole network, where we might examine the degree of variance, and the shape of the distribution of the dyads' connections. For example, a network in which the total connections among all pairs of actors might be expected to behave very differently than one where there are radical differences among actors.

The Hubbell and Katz approach may make the most sense when applied to symmetric data, because they pay no attention to the directions of connections (i.e. A's ties directed to B are just as important as B's ties to A in defining the distance or solidarity - closeness - between them). If we are more specifically interested in the influence of A on B in a directed graph, the Taylor influence approach provides an interesting alternative.

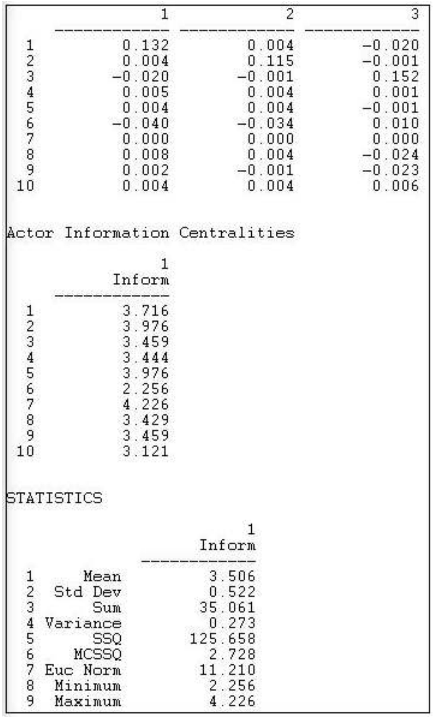
The Taylor measure, like the others, uses all connections, and applies an attenuation factor. Rather than standardizing on the whole resulting matrix, however, a different approach is adopted. The column marginals for each actor are subtracted from the row marginals, and the result is then normed (what did he say?!). Translated into English, we look at the balance between each actor sending connections (row marginals) and their receiving connections (column marginals). Positive values then reflect a preponderance of sending over receiving to the other actor of the pair - or a balance of influence between the two. Note that the newspaper (#7) shows as being a net influencer with respect to most other actors in the result below, while the welfare rights organization (#6) has a negative balance of influence with most other actors. The results for the Knoke information network are shown in Figure 10.15.



```
Method: TAYLOR
               1     2     3     4     5     6     7     8     9    10
            COUN  COMM  EDUC  INDU  MAYR   WRO  NEWS  UWAY  WELF  WEST
           ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   1 COUN   0.00 -0.02  0.23 -0.07  0.12  0.11 -0.09 -0.15  0.03  0.18
   2 COMM   0.02  0.00  0.11 -0.06  0.07  0.05 -0.05 -0.09  0.05  0.09
   3 EDUC  -0.23 -0.11  0.00  0.17 -0.36  0.18  0.26  0.02 -0.44 -0.02
   4 INDU   0.07  0.06 -0.17  0.00  0.05 -0.17 -0.02  0.11  0.14 -0.14
   5 MAYR  -0.12 -0.07  0.36 -0.05  0.00  0.30  0.01 -0.23 -0.13  0.23
   6  WRO  -0.11 -0.05 -0.18  0.17 -0.30  0.00  0.19  0.14 -0.32 -0.14
   7 NEWS   0.09  0.05 -0.26  0.02 -0.01 -0.19  0.00  0.15  0.12 -0.18
   8 UWAY   0.15  0.09 -0.02 -0.11  0.23 -0.14 -0.15  0.00  0.28  0.00
   9 WELF  -0.03 -0.05  0.44 -0.14  0.13  0.32 -0.12 -0.28  0.00  0.31
  10 WEST  -0.18 -0.09  0.02  0.14 -0.23  0.14  0.18 -0.00 -0.31  0.00
```

Figure 10.15: *Taylor dyadic influence for the Knoke information network*

Yet another measure based on attenuating and norming all pathways between each actor and all others was proposed by Stephenson and Zelen, and can be computed with *Network>Centrality>Information*. This measure, shown in Figure 10.16, provides a more complex norming of the distances from each actor to each other, and summarizes the centrality of each actor with the harmonic mean of its distance to the others.



```
                     1            2            3
          ------------ ------------ ------------
   1             0.132        0.004       -0.020
   2             0.004        0.115       -0.001
   3            -0.020       -0.001        0.152
   4             0.005        0.004        0.001
   5             0.004        0.004       -0.001
   6            -0.040       -0.034        0.010
   7             0.000        0.000        0.000
   8             0.008        0.004       -0.024
   9             0.002       -0.001       -0.023
  10             0.004        0.004        0.006

Actor Information Centralities

                     1
                Inform
          ------------
   1             3.716
   2             3.976
   3             3.459
   4             3.444
   5             3.976
   6             2.256
   7             4.226
   8             3.429
   9             3.459
  10             3.121

STATISTICS

                              1
                         Inform
                   ------------
   1      Mean            3.506
   2   Std Dev            0.522
   3       Sum           35.061
   4  Variance            0.273
   5       SSQ          125.658
   6     MCSSQ            2.728
   7  Euc Norm           11.210
   8   Minimum            2.256
   9   Maximum            4.226
```

Figure 10.16: *Stephenson and Zelen information centrality of Knoke information network*

The (truncated) top panel shows the dyadic distance of each actor to each other. The summary measure is shown in the middle panel, and information about the distribution of the centrality scores is shown in the statistics section.

As with most other measures, the various approaches to the distance between actors and in the network as a whole provide a menu of choices. No one definition to measuring distance will be the "right" choice for a given purpose. Sometimes we don't really know, beforehand, what approach might be best, and we may have to try and test several.

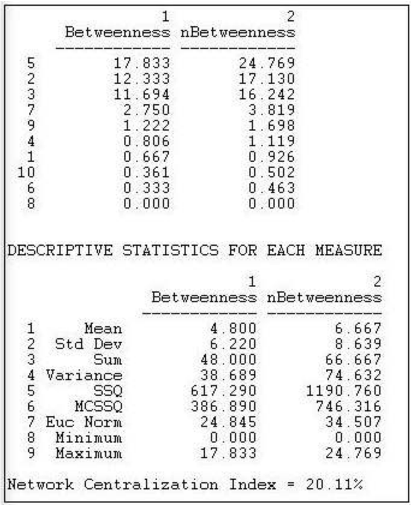
10.4: Betweenness Centrality

Suppose that I want to influence you by sending you information, or make a deal to exchange some resources. But, in order to talk to you, I must go through an intermediary. For example, let's suppose that I wanted to try to convince the Chancellor of my university to buy me a new computer. According to the rules of our bureaucratic hierarchy, I must forward my request through my department chair, a dean, and an executive vice chancellor. Each one of these people could delay the request, or even prevent my request from getting through. This gives the people who lie "between" me and the Chancellor power with respect to me. To stretch the example just a bit more, suppose that I also have an appointment in the school of business, as well as one in the department of sociology. I might forward my request to the Chancellor by both channels. Having more than one channel makes me less dependent, and, in a sense, more powerful.

For networks with binary relations, Freeman created some measures of the centrality of individual actors based on their betweenness, as well as overall graph centralization. Freeman, Borgatti, and White extended the basic approach to deal with valued relations.

Freeman's Approach to Binary Relations

With binary data, betweenness centrality views an actor as being in a favored position to the extent that the actor falls on the geodesic paths between other pairs of actors in the network. That is, the more people depend on me to make connections with other people, the more power I have. If, however, two actors are connected by more than one geodesic path, and I am not on all of them, I lose some power. Using the computer, it is quite easy to locate the geodesic paths between all pairs of actors, and to count up how frequently each actor falls on each of these pathways. If we add up, for each actor, the proportion of times that they are "between" other actors for the sending of information in the Knoke data, we get a measure of actor centrality. We can norm this measure by expressing it as a percentage of the maximum possible betweenness that an actor could have had. *Network>Centrality>Betweenness>Nodes* can be used to calculate Freeman's betweenness measures for actors. The results for the Knoke information network are shown in Figure 10.17.

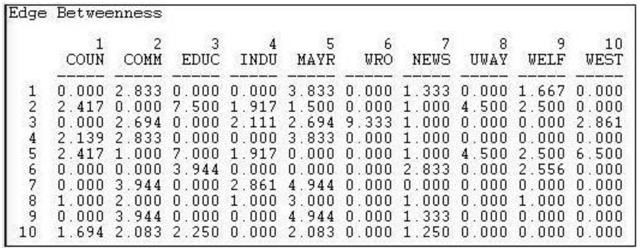


```
                   1            2
         Betweenness nBetweenness
        ------------ ------------
  5           17.833       24.769
  2           12.333       17.130
  3           11.694       16.242
  7            2.750        3.819
  9            1.222        1.698
  4            0.806        1.119
  1            0.667        0.926
 10            0.361        0.502
  6            0.333        0.463
  8            0.000        0.000


DESCRIPTIVE STATISTICS FOR EACH MEASURE

                            1            2
                  Betweenness nBetweenness
                 ------------ ------------
  1      Mean           4.800        6.667
  2   Std Dev           6.220        8.639
  3       Sum          48.000       66.667
  4  Variance          38.689       74.632
  5       SSQ         617.290     1190.760
  6     MCSSQ         386.890      746.316
  7  Euc Norm          24.845       34.507
  8   Minimum           0.000        0.000
  9   Maximum          17.833       24.769

Network Centralization Index = 20.11%
```

Figure 10.17: *Freeman node betweenness for Knoke information network*

We can see that there is a lot of variation in actor betweenness (from zero to 17.83), and that there is quite a bit of variation (std. dev. = 6.2 relative to a mean betweenness of 4.8). Despite this, the overall network centralization is relatively low. This makes sense, because we know that fully one half of all connections can be made in this network without the aid of any intermediary - hence there cannot be a lot of "betweenness". In the sense of structural constraint, there is not a lot of "power" in this network. Actors #2, #3, and #5 appear to be relatively a good bit more powerful than others by this measure. Clearly, there is a structural basis for these actors to perceive that they are "different" from others in the population. Indeed, it would not be surprising if these three actors saw themselves as the movers-and-shakers, and the deal-makers that made things happen. In this sense, even though there is not very much betweenness power in the system, it could be important for group formation and stratification.

Another way to think about betweenness is to ask which relations are most central, rather than which actors. Freeman's definition can be easily applied: a relation is between to the extent that it is part of the geodesic between pairs of actors. Using this idea, we can calculate a measure of the extent to which each relation in a binary graph is between. In UCINET, this is done with *Network>Centrality>Betweenness>Lines (edges)*. The results for the Knoke information network are shown in Figure 10.18.



```
Edge Betweenness

          1      2      3      4      5      6      7      8      9     10
       COUN   COMM   EDUC   INDU   MAYR    WRO   NEWS   UWAY   WELF   WEST
      ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1   0.000  2.833  0.000  0.000  3.833  0.000  1.333  0.000  1.667  0.000
  2   2.417  0.000  7.500  1.917  1.500  0.000  1.000  4.500  2.500  0.000
  3   0.000  2.694  0.000  2.111  2.694  9.333  1.000  0.000  0.000  2.861
  4   2.139  2.833  0.000  0.000  3.833  0.000  1.000  0.000  0.000  0.000
  5   2.417  1.000  7.000  1.917  0.000  0.000  1.000  4.500  2.500  6.500
  6   0.000  0.000  3.944  0.000  0.000  0.000  2.833  0.000  2.556  0.000
  7   0.000  3.944  0.000  2.861  4.944  0.000  0.000  0.000  0.000  0.000
  8   1.000  2.000  0.000  1.000  3.000  0.000  1.000  0.000  1.000  0.000
  9   0.000  3.944  0.000  0.000  4.944  0.000  1.333  0.000  0.000  0.000
 10   1.694  2.083  2.250  0.000  2.083  0.000  1.250  0.000  0.000  0.000
```

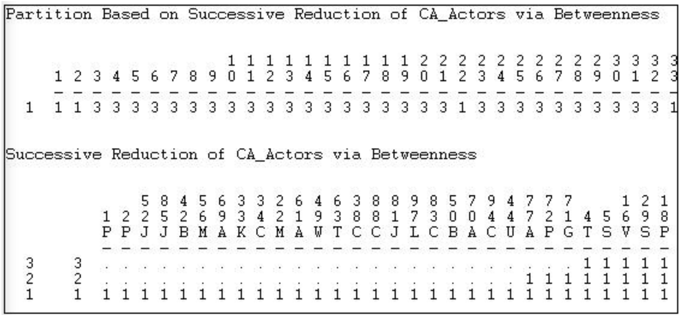
Figure 10.18: *Freeman edge betweenness for Knoke information network*

A number of the relations (or potential relations) between pairs of actors are not parts of any geodesic paths (e.g. the relation from actor 1 to actor 3). Betweenness is zero if there is no tie, or if a tie that is present is not part of any geodesic path. There are some quite central relations in the graph. For example, the tie from the board of education (actor 3) to the welfare rights organization (actor 6). This particular high value arises because without the tie to actor 3, actor 6 would be largely isolated.

Suppose A has ties to B and C. B has ties to D and E; C has ties to F and G. Actor A will have high betweenness, because it connects two branches of ties, and lies on many geodesic paths. Actors B and C also have betweenness, because they lie between A and their "subordinates". But actors D, E, F, and G have zero betweenness.

One way of identifying hierarchy in a set of relations is to locate the "subordinates". These actors will be ones with no betweenness. If we then remove these actors from the graph, some of the remaining actors won't be between any more - so they are one step up in the hierarchy. We can continue doing this "hierarchical reduction" until we've exhausted the graph; what we're left with is a map of the levels of hierarchy.

Network>Centrality>Betweenness>Hierarchical Reduction is an algorithm that identifies which actors fall at which levels of a hierarchy (if there is one). Since there is very little hierarchy in the Knoke data, we've illustrated this instead with a network of large donors to political campaigns in California, who are "connected" if they contribute to the same campaign. A part of the results is shown in Figure 10.19.



```
Partition Based on Successive Reduction of CA_Actors via Betweenness


                  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3
    1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 1  1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 1


Successive Reduction of CA_Actors via Betweenness


            5 8 4 5 6 3 3 2 6 4 6 3 8 8 9 8 5 7 9 4 7 7 7     1 2 1
        1 2 2 5 2 6 9 3 4 2 1 9 3 8 8 1 7 3 0 0 4 4 7 2 1 4 5 6 9 8
        P P J J B M A K C M A W T C C J L C B A C U A P G T S V S P
        - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 3   3  . . . . . . . . . . . . . . . . . . . . . . . . . 1 1 1 1 1
 2   2  . . . . . . . . . . . . . . . . . . . . . . 1 1 1 1 1 1 1 1
 1   1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Figure 10.19: *Hierarchical reduction by betweenness for California political donors (truncated)*

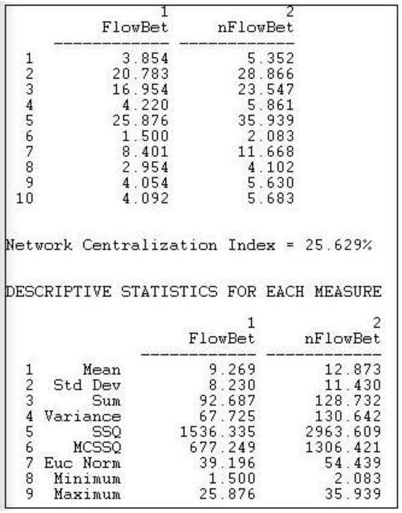
In these data, it turns out that a three-level hierarchy can be identified. The first portion of the output shows a partition (which can be saved as a file, and used as an attribute to color a graph) of the node's level in the hierarchy. The first two nodes, for example, are at the lowest level (1) of the hierarchy, while the third node is at the third level. The second portion of the output has rearranged the nodes to show which actors are included at the lowest betweenness (level one, or everyone); which drop out at level 2 (that is, are the most subordinate; e.g. actors 1, 2, 52); and successive levels. Our data has a hierarchical depth of only three.

Flow Centrality

The betweenness centrality measure we examined above characterizes actors as having positional advantage, or power, to the extent that they fall on the shortest (geodesic) pathway between other pairs of actors. The idea is that actors who are "between" other actors, and on whom other actors must depend to conduct exchanges, will be able to translate this broker role into power.

Suppose that two actors want to have a relationship, but the geodesic path between them is blocked by a reluctant broker. If there exists another pathway, the two actors are likely to use it, even if it is longer and "less efficient". In general, actors may use all of the pathways connecting them, rather than just geodesic paths. The flow approach to centrality expands the notion of betweenness centrality. It assumes that actors will use all pathways that connect them, proportionally to the length of the pathways. Betweenness is measured by the proportion of the entire flow between two actors (that is, through all of the pathways connecting them) that occurs on paths of which a given actor is a part. For each actor, then, the measure adds up how involved that actor is in all of the flows between all other pairs of actors (the amount of computation with more than a couple actors can be pretty intimidating!). Since the magnitude of this index number would be expected to increase with sheer size of the network and with network density, it is useful to standardize it by calculating the flow betweenness of each actor in ratio to the total flow betweenness that does not involve the actor.

The algorithm *Network>Centrality>Flow Betweenness* calculates actor and graph flow betweenness centrality measures. Results of applying this to the Knoke information network are shown in Figure 10.20.



```
                  1            2
            FlowBet      nFlowBet
       ------------ ------------
  1           3.854        5.352
  2          20.783       28.866
  3          16.954       23.547
  4           4.220        5.861
  5          25.876       35.939
  6           1.500        2.083
  7           8.401       11.668
  8           2.954        4.102
  9           4.054        5.630
 10           4.092        5.683

Network Centralization Index = 25.629%

DESCRIPTIVE STATISTICS FOR EACH MEASURE

                            1            2
                      FlowBet     nFlowBet
                 ------------ ------------
  1     Mean            9.269       12.873
  2  Std Dev            8.230       11.430
  3      Sum           92.687      128.732
  4 Variance           67.725      130.642
  5      SSQ         1536.335     2963.609
  6    MCSSQ          677.249     1306.421
  7 Euc Norm           39.196       54.439
  8  Minimum            1.500        2.083
  9  Maximum           25.876       35.939
```

Figure 10.20: *Flow betweenness centrality for Knoke information network*

By this more complete measure of betweenness centrality, actors #2 and #5 are clearly the most important mediators. Actor #3, who was fairly important when we considered only geodesic flows, appears to be rather less important. While the overall picture does not change a great deal, the elaborated definition of betweenness does give us a somewhat different impression of who is most central in this network.

Some actors are clearly more central than others, and the relative variability in flow betweenness of the actors is fairly great (the standard deviation of normed flow betweenness is 8.2 relative to a mean of 9.2, giving a coefficient of relative variation). Despite this relatively high amount of variation, the degree of inequality, or concentration in the distribution of flow betweenness centralities among the actors is fairly low - relative to that of a pure star network (the network centralization index is 25.6%). This is slightly higher than the index for the betweenness measure that was based only on geodesic distances.

10.E: Centrality and Power (Exercises)

Review Questions

1. What is the difference between "centrality" and "centralization"?
2. Why is an actor who has higher degree a more "central" actor?
3. How does Bonacich's influence measure extend the idea of degree centrality?
4. Can you explain why an actor who has the smallest sum of geodesic distances to all other actors is said to be the most "central" actor, using the "closeness" approach?
5. How does the "flow" approach extend the idea of "closeness" as an approach to centrality?
6. What does it mean to say that an actor lies "between" two other actors? Why does betweenness give an actor power or influence?
7. How does the "flow" approach extend the idea of "betweenness" as an approach to centrality?
8. Most approaches suggest that centrality confers power and influence. Bonacich suggests that power and influence are not the same thing. What is Bonacich's argument? How does Bonacich measure the power of an actor?

Application Questions

1. Think of the readings from the first part of the course. Which studies used the ideas of structural advantage, centrality, power, and influence? What kind of approach did each use: degree, closeness, or betweenness?
2. Can you think of any circumstances where being "central" might make one less influential? Less powerful?
3. Consider a directed network that describes a hierarchical bureaucracy, where the relationship is "gives orders to". Which actors have highest degree? Are they the most powerful and influential? Which actors have high closeness? Which actors have high betweenness?
4. Can you think of a real-world example of an actor who might be powerful but not central? Who might be central, but not powerful?

10.S: Summary

Power arises from occupying advantageous positions in networks of relations. Three basic sources of advantage are high degree, high closeness, and high betweenness. In simple structures (such as the star, circle, or line), these advantages tend to covary. In more complex and larger networks, there can be considerable disjuncture between these characteristics of a position - so that an actor may be located in a position that is advantageous in some ways, and disadvantageous in others.

We have reviewed three basic approaches to the "centrality" of individuals' positions, and some elaborations on each of the three main ideas of degree, closeness, and betweenness. This review is not exhaustive. The question of how structural position confers power remains a topic of active research and considerable debate. As you can see, different definitions and measures can capture different ideas about where power comes from, and can result in some rather different insights about social structures.

In the last chapter and this one, we have emphasized that social network analysis methods give us, at the same time, views of individuals and of whole populations. One of the most enduring and important themes in the study of human social organization, however, is the importance of social units that lie between the two poles of individuals and whole populations. In the next chapter, we will turn our attention to how network analysis methods describe and measure the differentiation of sub-populations.

CHAPTER OVERVIEW

11: Cliques and Sub-groups

One of the most common interests of structural analysts is in the "sub-structures" that may be present in a network. The dyads, triads, and ego-centered neighborhoods that we examined earlier can all be though of as sub-structures. In this chapter, we'll consider some approaches to identifying larger groupings.

11.1: Introduction to Cliques and Sub-groups

One of the most common interests of structural analysts is in the "sub-structures" that may be present in a network. The dyads, triads, and ego-centered neighborhoods that we examined earlier can all be though of as sub-structures. In this chapter, we'll consider some approaches to identifying larger groupings.

Many of the approaches to understanding the structure of a network emphasize how dense connections are built-up from simpler dyads and triads to more extended dense clusters such as "cliques". This view of social structure focuses attention on how solidarity and connection of large social structures can be built up out of small and tight components: a sort of "bottom up" approach. Network analysts have developed a number of useful definitions and algorithms that identify how larger structures are compounded from smaller ones: cliques, n-cliques, n-clans, and k-plexes all look at networks in this way.

Divisions of actors into groups and sub-structures can be a very important aspect of social structure. It can be important in understanding how the network as a whole is likely to behave. Suppose the actors in one network form two non-overlapping groups; and, suppose that the actors in another network also form two groups, but that the memberships overlap (some people are members of both groups). Where the groups overlap, we might expect that conflict between them is less likely that when the groups don't overlap. Where the groups overlap, mobilization and diffusion may spread rapidly across the entire network; where the groups don't overlap, traits may occur in one group and not diffuse to the other.

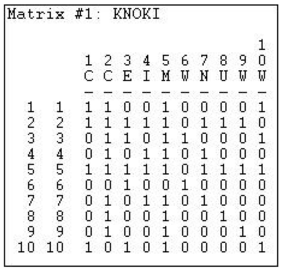
Knowing how an individual is embedded in the structure of groups within a net may also be critical to understanding his/her behavior. For example, some people may act as "bridges" between groups (cosmopolitans, boundary spanners, or "brokers" that we examined earlier). Others may have all of their relationships within a single group (locals or insiders). Some actors may be part of a tightly connected and closed elite, while others are completely isolated from this group. Such differences in the ways that individuals are embedded in the structure of groups within a network can have profound consequences for the ways that these actors see their "society", and the behaviors that they are likely to practice.

We can also look for sub-structure from the "top-down". Looking at the whole network, we can think of sub-structures as areas of the graph that seem to be locally dense, but separated to some degree, from the rest of the graph. This idea has been applied in a number of ways: components, blocks/cutpoints, K-cores, Lambda sets and bridges, factions, and f-groups will be discussed here.

The idea that some regions of a graph may be less connected to the whole than others may lead to insights into lines of cleavage and division. Weaker parts in the "social fabric" also create opportunities for brokerage and less constrained action. So, the numbers and sizes of regions, and their "connection topology" may be consequential for predicting both the opportunities and constraints facing groups and actors, as well as predicting the evolution of the graph itself.

Most computer algorithms for locating sub-structures operate on binary symmetric data. We will use the Knoke information exchange data for most of the illustrations again in this chapter. Where algorithms allow it, the directed form of the data will be used. Where symmetric data are called for, we will analyze "strong ties". That is, we will symmetrize the data by insisting that ties must be reciprocated in order to count; that is, a tie only exists if xy and yx are both present.

The resulting reciprocity-symmetric data matrix is shown in Figure 11.1.



```
Matrix #1: KNOKI

                                     1
             1  2  3  4  5  6  7  8  9  0
             C  C  E  I  M  W  N  U  W  W
             -  -  -  -  -  -  -  -  -  -
  1   1      1  1  0  0  1  0  0  0  0  1
  2   2      1  1  1  1  1  0  1  1  1  0
  3   3      0  1  1  0  1  1  0  0  0  1
  4   4      0  1  0  1  1  0  1  0  0  0
  5   5      1  1  1  1  1  0  1  1  1  1
  6   6      0  0  1  0  0  1  0  0  0  0
  7   7      0  1  0  1  1  0  1  0  0  0
  8   8      0  1  0  0  1  0  0  1  0  0
  9   9      0  1  0  0  1  0  0  0  1  0
 10  10      1  0  1  0  1  0  0  0  0  1
```

Figure 11.1: *Knoke information network symmetrized with reciprocated ties*

Insisting that information move in both directions between the parties in order for the two parties to be regarded as "close" makes theoretical sense, and substantially lessens the density of the matrix. Matrices that have very high density, almost by definition, are likely to have few distinctive sub-groups or cliques. It might help to graph these data as in Figure 11.2.

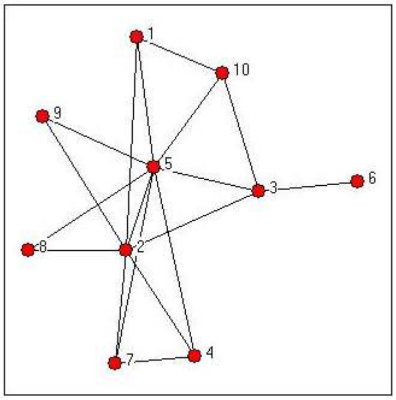


Figure 11.2: *Graph of Knoke information strong symmetric ties*

The diagram suggests a number of things. Actors #5 and #2 appear to be in the middle of the action - in the sense that they are members of many of the groupings, and serve to connect them, by co-membership. The connection of sub-graphs by actors can be an important feature. We can also see that there is one case (#6) that is not a member of any sub-group (other than a dyad). If you look closely, you will see that dyads and triads are the most common sub-graphs here - and despite the substantial connectivity of the graph, tight groupings larger than this seem to be few. It is also apparent from visual inspection that most of the sub-groupings are connected - that groups overlap.

Answers to the main questions about a graph, in terms of its sub-structures, may be apparent from inspection:

- How separate are the sub-graphs? Do they overlap and share members, or do they divide or factionalize the network?
- How large are the connected sub-graphs? Are there a few big groups, or a larger number of small groups?
- Are there particular actors that appear to play network roles? For example, act as nodes that connect the graph, or who are isolated from groups?

The formal tools and concepts of sub-graph structure help to more rigorously define ideas like this. Various algorithms can then be applied to locate, list, and study sub-graph features. Obviously, there are a number of possible groupings and positions in sub-structures, depending on our definitions. In this chapter, we will look at the most common of these ideas.

11.2: Bottom-up Approaches

In a sense, all networks are composed of groups (or sub-graphs). When two actors have a tie, they form a "group". One approach to thinking about the group structure of a network begins with this most basic group, and seeks to see how far this kind of close relationship can be extended. This is a useful way of thinking, because sometimes more complex social structures evolve, or emerge, from very simple ones.

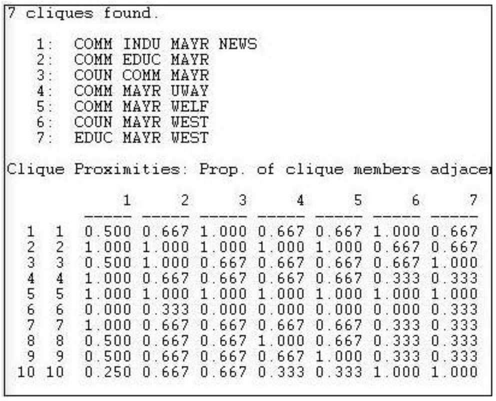
A clique extends the dyad by adding to it members who are tied to all of the members of the group. This strict definition can be relaxed to include additional nodes that are not quite so tightly tied (n-cliques, n-clans, and k-plexes). The notion, however, is to build outward from single ties to "construct" the network. A map of the whole network can be built up by examining the sizes of various cliques and clique-like groupings, and noting their sizes and overlaps.

These kinds of approaches to thinking about the sub-structures of networks tend to emphasize how the macro might emerge out of the micro. They tend to focus our attention on individuals first, and try to understand how they are embedded in the web of overlapping groups in the larger structure. I make a point of these seemingly obvious ideas because it is also possible to approach the question of the sub-structure of networks from the top-down. Usually, both approaches are worthwhile and complementary. We will turn our attention first to "bottom-up" thinking.

Cliques

The idea of a clique is relatively simple. At the most general level, a clique is a sub-set of a network in which the actors are more closely and intensely tied to one another than they are to other members of the network. In terms of friendship ties, for example, it is not unusual for people in human groups to form "cliques" on the basis of age, gender, race, ethnicity, religion/ideology, and many other things. The smallest "cliques" are composed of two actors: the dyad. But dyads can be "extended" to become more and more inclusive - forming strong or closely connected regions in graphs. A number of approaches to finding groups in graphs can be developed by extending the close-coupling of dyads to larger structures.

The formal definition of a "clique" as it is used in network analysis is much more narrow and precise than the general notion of a high local density. Formally, a clique is the maximum number of actors who have all possible ties present among themselves. A "Maximal complete sub-graph" is such a grouping, expanded to include as many actors as possible. The UCINET algorithm *Network>Subgroups>Cliques* produces a census of all cliques, and some useful additional analysis. The result, applied to our symmetrized Knoke information matrix, is shown in Figures 11.3 through 11.5.

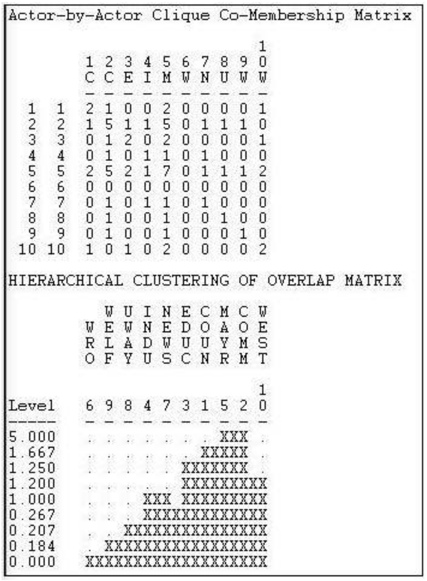


```
7 cliques found.

   1:   COMM INDU MAYR NEWS
   2:   COMM EDUC MAYR
   3:   COUN COMM MAYR
   4:   COMM MAYR UWAY
   5:   COMM MAYR WELF
   6:   COUN MAYR WEST
   7:   EDUC MAYR WEST

Clique Proximities: Prop. of clique members adjace

              1     2     3     4     5     6     7
          ----- ----- ----- ----- ----- ----- -----
  1  1    0.500 0.667 1.000 0.667 0.667 1.000 0.667
  2  2    1.000 1.000 1.000 1.000 1.000 0.667 0.667
  3  3    0.500 1.000 0.667 0.667 0.667 0.667 1.000
  4  4    1.000 0.667 0.667 0.667 0.667 0.333 0.333
  5  5    1.000 1.000 1.000 1.000 1.000 1.000 1.000
  6  6    0.000 0.333 0.000 0.000 0.000 0.000 0.333
  7  7    1.000 0.667 0.667 0.667 0.667 0.333 0.333
  8  8    0.500 0.667 0.667 1.000 0.667 0.333 0.333
  9  9    0.500 0.667 0.667 0.667 1.000 0.333 0.333
 10 10    0.250 0.667 0.667 0.333 0.333 1.000 1.000
```

Figure 11.3: *Clique and actor-by-clique analysis of reciprocity-symmetrized Knoke information network*

There are seven maximal complete sub-graphs present in these data (see if you can find them in Figure 11.2). The largest one is composed of four of the ten actors, and all of the other smaller cliques share some overlap with some part of the largest clique. The second panel shows how "adjacent" each actor (row) is to each clique (column). Actor 1, for example, is adjacent to 2/3 of the members of clique 5. There is a very high degree of common membership in these data.

We might be interested in the extent to which these sub-structures overlap, and which actors are most "central" and most "isolated" from the cliques. We can examine these questions by looking at "co-membership".

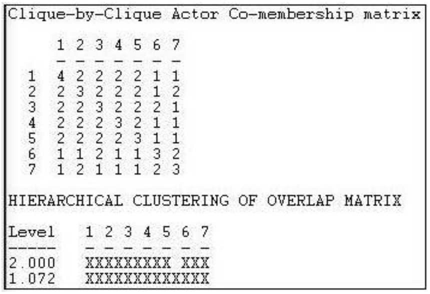


```
Actor-by-Actor Clique Co-Membership Matrix

                          1
        1 2 3 4 5 6 7 8 9 0
        C C E I M W N U W W
        - - - - - - - - - -
 1  1   2 1 0 0 2 0 0 0 0 1
 2  2   1 5 1 1 5 0 1 1 1 0
 3  3   0 1 2 0 2 0 0 0 0 1
 4  4   0 1 0 1 1 0 1 0 0 0
 5  5   2 5 2 1 7 0 1 1 1 2
 6  6   0 0 0 0 0 0 0 0 0 0
 7  7   0 1 0 1 1 0 1 0 0 0
 8  8   0 1 0 0 1 0 0 1 0 0
 9  9   0 1 0 0 1 0 0 0 1 0
10 10   1 0 1 0 2 0 0 0 0 2

HIERARCHICAL CLUSTERING OF OVERLAP MATRIX

          W U I N E C M C W
        W E W N E D O A O E
        R L A D W U U Y M S
        O F Y U S C N R M T

                          1
Level   6 9 8 4 7 3 1 5 2 0
-----   - - - - - - - - - -
5.000   . . . . . . . XXX .
1.667   . . . . . . XXXXX .
1.250   . . . . . XXXXXXX .
1.200   . . . . . XXXXXXXXX
1.000   . . . XXX XXXXXXXXX
0.267   . . . XXXXXXXXXXXXX
0.207   . . XXXXXXXXXXXXXXX
0.184   . XXXXXXXXXXXXXXXXX
0.000   XXXXXXXXXXXXXXXXXXX
```

Figure 11.4: *Actor-by-actor analysis of reciprocity-symmetrized Knoke information network*

The first panel here shows how many cliques each pair of actors are both members of. It is immediately apparent that actor #6 is a complete isolate, and that actors #2 and #5 overlap with almost all other actors in at least one clique. We see that actors #2 and #5 are "closest" in the sense of sharing membership in five of the seven cliques. We can take this kind of analysis one step further by using single linkage agglomerative cluster analysis to create a "joining sequence" based on how many clique memberships actors have in common. This is shown in the second panel of Figure 11.4. We see that actors 2 and 5 are "joined" first as being close because they share 5 clique memberships in common.

Moving to still a higher level, we can look at the extent to which the cliques overlap with one another, as measured by the numbers of members in common, as in Figure 11.5.



```
Clique-by-Clique Actor Co-membership matrix

     1 2 3 4 5 6 7
     - - - - - - -
 1   4 2 2 2 2 1 1
 2   2 3 2 2 2 1 2
 3   2 2 3 2 2 2 1
 4   2 2 2 3 2 1 1
 5   2 2 2 2 3 1 1
 6   1 1 2 1 1 3 2
 7   1 2 1 1 1 2 3

HIERARCHICAL CLUSTERING OF OVERLAP MATRIX

Level   1 2 3 4 5 6 7
-----   - - - - - - -
2.000   XXXXXXXXX XXX
1.072   XXXXXXXXXXXXX
```

Figure 11.5: *Clique-by-clique analysis of reciprocity-symmetrized Knoke information network*

A cluster analysis of the closeness of the cliques shows that cliques 6 and 7 are (a little) separate from the other cliques.

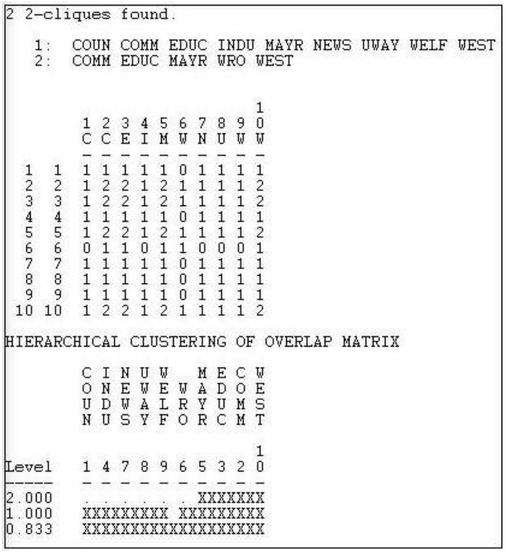
You might note that the (rather lengthy) output again points to the multi-level nature of network analysis. We can see actors related to actors to define groups; we can see actors related to groups; and we can see groups related to groups in this analysis of the clique structure.

Insisting that every member of a clique be connected to every other member is a very strong definition of what we mean by a group. There are a number of ways in which this restriction could be relaxed. Two major approaches are the n-clique/n-clan approach and the k-plex approach.

N-cliques

The strict clique definition (maximal fully-connected sub-graph) may be too strong for many purposes. It insists that every member of a sub-group have a direct tie with each and every other member. You can probably think of cases of "cliques" where at least some members are not so tightly or closely connected. There are two major ways that the "clique" definition has been "relaxed" to try to make it more helpful and general.

One alternative is to define an actor as a member of a clique if they are connected to every other member of the group at a distance greater than one. Usually, the path distance two is used. This corresponds to being "a friend of a friend". This approach to defining sub-structures is called n-clique, where n stands for the length of the path allowed to make a connection to all other members. *Network>Subgroups>N-Cliques* finds these sub-structures and performs overlap analysis. Figure 11.6 shows the census of n-cliques for n=2.



```
2 2-cliques found.

   1:   COUN COMM EDUC INDU MAYR NEWS UWAY WELF WEST
   2:   COMM EDUC MAYR WRO WEST


                           1
         1 2 3 4 5 6 7 8 9 0
         C C E I M W N U W W
         - - - - - - - - - -
  1  1   1 1 1 1 1 0 1 1 1 1
  2  2   1 2 2 1 2 1 1 1 1 2
  3  3   1 2 2 1 2 1 1 1 1 2
  4  4   1 1 1 1 1 0 1 1 1 1
  5  5   1 2 2 1 2 1 1 1 1 2
  6  6   0 1 1 0 1 1 0 0 0 1
  7  7   1 1 1 1 1 0 1 1 1 1
  8  8   1 1 1 1 1 0 1 1 1 1
  9  9   1 1 1 1 1 0 1 1 1 1
 10 10   1 2 2 1 2 1 1 1 1 2

HIERARCHICAL CLUSTERING OF OVERLAP MATRIX

         C I N U W   M E C W
         O N E W E W A D O E
         U D W A L R Y U M S
         N U S Y F O R C M T

                           1
Level    1 4 7 8 9 6 5 3 2 0
-----    - - - - - - - - - -
2.000    . . . . . . XXXXXXX
1.000    XXXXXXXXX XXXXXXXXX
0.833    XXXXXXXXXXXXXXXXXXX
```

Figure 11.6: *N-cliques of reciprocity-symmetrized Knoke information network (n=2)*

The cliques that we saw before have been made more inclusive by the relaxed definition of group membership. The first n-clique includes everyone but actor #6. The second is more restricted, and includes #6 (WRO), along with two elements of the core. Because our definition of how closely linked actors must be to be members of a clique has been relaxed, there are fewer maximal cliques. With larger and fewer sub-groups, the mayor (#5) no longer appears to be quite so critical. With the more relaxed definition, there is now an "inner circle" of actors that are members of both larger groupings. This can be seen in the co-membership matrix, and by clustering.

N-clans

The n-clique approach tends to find long and stringy groupings rather than the tight and discrete ones of the maximal approach. In some cases, n-cliques can be found that have a property that is probably undesirable for many purposes: it is possible for members of n-cliques to be connected by actors who are not, themselves, members of the clique. For most sociological applications, this is quite troublesome.

To overcome this problem, some analysis have suggested restricting n-cliques by insisting that the total span or path distance between any two members of an n-clique also satisfy a condition. The additional restriction has the effect of forcing all ties among members of an n-clique to occur by way of other members of the n-clique. This is the n-clan approach. *Network>Subgroups>N-Clan* can be used to produce a clique analysis using the n-clan rule. For the Knoke information matrix, as symmetrized here, the result is identical to the n-clique approach.

The n-clique and n-clan approaches provide an alternative to the stricter "clique" definition, and this more relaxed approach often makes good sense with sociological data. In essence, the n-clique approach allows an actor to be a member of a clique even if they do not have ties to all other clique members; just so long as they do have ties to some member, and are no further away than n steps (usually 2) from all members of the clique. The n-clan approach is a relatively minor modification on the n-clique approach that requires that all the ties among actors occur through other members of the group.

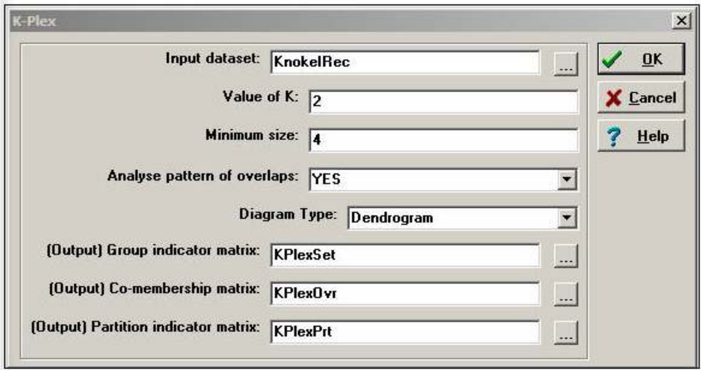
If one is uncomfortable with regarding the fried of a clique member as also being a member of the clique (the n-clique approach), one might consider an alternative way of relaxing the strict assumptions of the clique definition - the k-plex approach.

K-plexes

An alternative way of relaxing the strong assumptions of the "Maximal Complete Sub-Graph" is to allow that actors may be members of a clique even if they have ties to all but k other members. For example, if A has ties with B and C, but not D; while both B and C have ties with D, all four actors could fall in clique under the k-plex approach. This approach says that a node is a member of a clique of size n if it has direct ties to n-k members of that clique.

The k-plex approach would seem to have quite a bit in common with the n-clique approach, but k-plex analysis often gives quite a different picture of the sub-structures of a graph. Rather than the large and "stringy" groupings sometimes produced by n-clique analysis, k-plex analysis tends to find relatively large numbers of smaller groupings. This tends to focus attention on overlaps and co-presence (centralization) more than solidarity and reach.

In our example, below, we have allowed k to be equal to two, but insisted that a k-plex grouping include at least four members. That is, an actor is considered to be a member of a clique if that actor has ties to all but two others (at a minimum, half) in that clique. Figure 11.7 shows the dialog of *Network>Subgroups>K-Plex* that specifies our definition.



Input dataset: KnokelRec

Value of K: 2

Minimum size: 4

Analyse pattern of overlaps: YES

Diagram Type: Dendrogram

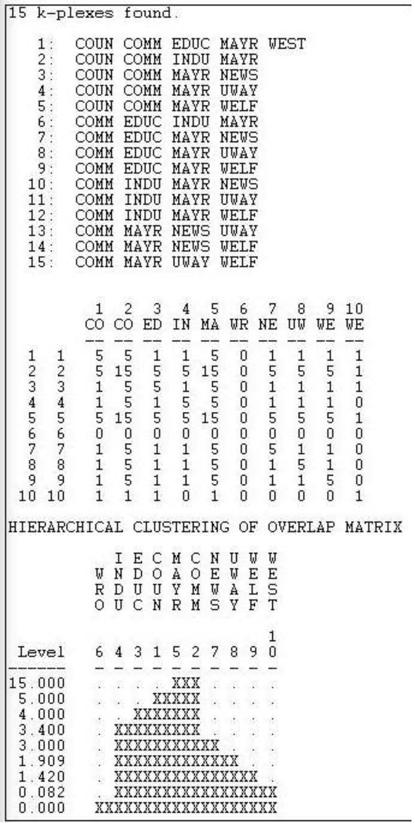
(Output) Group indicator matrix: KPlexSet

(Output) Co-membership matrix: KPlexOvr

(Output) Partition indicator matrix: KPlexPrt

Figure 11.7: *Dialog of Network>Subgroups>K-Plex for groups of at least four with k=2*

The results of the k-plex analysis are shown in Figure 11.8.



```
15 k-plexes found.

   1:   COUN COMM EDUC MAYR WEST
   2:   COUN COMM INDU MAYR
   3:   COUN COMM MAYR NEWS
   4:   COUN COMM MAYR UWAY
   5:   COUN COMM MAYR WELF
   6:   COMM EDUC INDU MAYR
   7:   COMM EDUC MAYR NEWS
   8:   COMM EDUC MAYR UWAY
   9:   COMM EDUC MAYR WELF
  10:   COMM INDU MAYR NEWS
  11:   COMM INDU MAYR UWAY
  12:   COMM INDU MAYR WELF
  13:   COMM MAYR NEWS UWAY
  14:   COMM MAYR NEWS WELF
  15:   COMM MAYR UWAY WELF


           1  2  3  4  5  6  7  8  9 10
          CO CO ED IN MA WR NE UW WE WE
          -- -- -- -- -- -- -- -- -- --
  1  1     5  5  1  1  5  0  1  1  1  1
  2  2     5 15  5  5 15  0  5  5  5  1
  3  3     1  5  5  1  5  0  1  1  1  1
  4  4     1  5  1  5  5  0  1  1  1  0
  5  5     5 15  5  5 15  0  5  5  5  1
  6  6     0  0  0  0  0  0  0  0  0  0
  7  7     1  5  1  1  5  0  5  1  1  0
  8  8     1  5  1  1  5  0  1  5  1  0
  9  9     1  5  1  1  5  0  1  1  5  0
 10 10     1  1  1  0  1  0  0  0  0  1
HIERARCHICAL CLUSTERING OF OVERLAP MATRIX

            I E C M C N U W W
          W N D O A O E W E E
          R D U U Y M W A L S
          O U C N R M S Y F T

                            1
 Level    6 4 3 1 5 2 7 8 9 0
------    - - - - - - - - - -
15.000    . . . . XXX . . . .
 5.000    . . . XXXXX . . . .
 4.000    . . XXXXXXX . . . .
 3.400    . XXXXXXXXX . . . .
 3.000    . XXXXXXXXXXX . . .
 1.909    . XXXXXXXXXXXXX . .
 1.420    . XXXXXXXXXXXXXXX .
 0.082    . XXXXXXXXXXXXXXXXX
 0.000    XXXXXXXXXXXXXXXXXXX
```

Figure 11.8: *Analysis of k-plex groups in Knoke reciprocity-symmetrized information network*

The COMM is present in every k-component; the MAYR is present in all but one. Clearly these two actors are "central" in the sense of playing a bridging role among multiple slightly different social circles. Again we note that organization #6 (WRO) is not a member of any k-plex clique. The k-plex method of defining cliques tends to find "overlapping social circles" when compared to the maximal or n-clique methods.

The k-plex approach to defining sub-structures makes a good deal of sense for many problems. It requires that members of a group have ties to (most) other group members - ties by way of intermediaries (like the n-clique approach) do not qualify a node for membership. The picture of group structure that emerges from k-plex approaches can be rather different from that of n-clique analysis. Again, it is not that one is "right" and the other "wrong". Depending on the goals of the analysis, both can yield valuable insights into the sub-structure of groups.

K-cores

A k-core is a maximal group of actors, all of whom are connected to some number (k) of other members of the group. To be included in a k-plex, an actor must be tied to all but k other actors in the group. The k-core approach is more relaxed, allowing actors to join the group if they are connected to k members, regardless of how many other members they may not be connected to. By varying the value of k (that is, how many members of the group you have to be connected to), different pictures can emerge. K-cores can be (and usually are) more inclusive than k-plexes. And, as k becomes smaller, group sizes will increase.

NetDraw includes a tool for identifying and coloring a graph according to its k-cores. The UCINET algorithm for identifying k-cores is located at *Network>Regions>K-Core*.

In our example data, if we require that each member of a group have ties to 3 other members (a 3-core), a rather large central group of actors is identified {1,2,3,4,5,7,10}. Each of the seven members of this core has ties to at least three others. If we relax the criterion to require only two ties, actors 8 and 9 are added to the group (and 6 remains an isolate). If we require only one tie (really, the same thing as a component), all actors are connected.

The k-core definition is intuitively appealing for some applications. If an actor has ties to a sufficient number of members of a group, they may feel tied to that group - even if they don't know many, or even most, members. It may be that identity depends on connection, rather than on immersion in a sub-group.

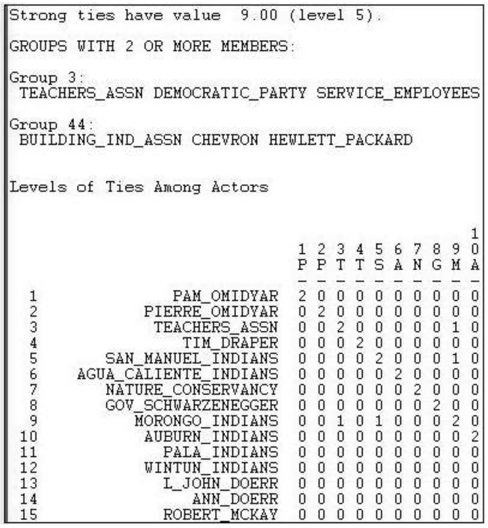
F-groups

All of the approaches we've examined so far deal with binary (and usually symmetric) relations. If we have information on the strength, cost, or probability of relations, we might also want to apply "bottom-up" thinking to find maximal groups. One approach would be to simply dichotomize the data (maybe at several different cut-points). *Network>Subgroups>f-Groups* is an algorithm that builds on this idea, and combines it with the notion that larger groups are composed of triadic relations.

F-groups identify maximal groups made up of "strongly transitive" and "weakly transitive" triads. A strong tie triad is formed when, if there is a tie XY and a tie YZ, there is also a tie XZ that is equal in value to the XY and YZ ties. A weakly transitive triad is formed if the ties XY and YZ are both stronger than the tie XZ, but the tie XZ is greater than some cut-off value.

Network>Subgroups>f-Groups takes the value of a strong tie to be equal to the largest valued tie in a graph. The user selects the cut-off value for what constitutes a weak tie.

Figure 11.9 shows the results of using this algorithm to identify strong and weak groups among the top 100 donors to California political campaigns. The value of the relation in these data is the number of campaigns to which donors both contributed. We have set our cut-off for a "weak tie" to be three campaigns in common.



```
Strong ties have value  9.00 (level 5).

GROUPS WITH 2 OR MORE MEMBERS:

Group 3:
 TEACHERS_ASSN DEMOCRATIC_PARTY SERVICE_EMPLOYEES

Group 44:
 BUILDING_IND_ASSN CHEVRON HEWLETT_PACKARD


Levels of Ties Among Actors


                                                1
                              1 2 3 4 5 6 7 8 9 0
                              P P T T S A N G M A
                              - - - - - - - - - -
  1               PAM_OMIDYAR 2 0 0 0 0 0 0 0 0 0
  2            PIERRE_OMIDYAR 0 2 0 0 0 0 0 0 0 0
  3             TEACHERS_ASSN 0 0 2 0 0 0 0 0 1 0
  4                TIM_DRAPER 0 0 0 2 0 0 0 0 0 0
  5        SAN_MANUEL_INDIANS 0 0 0 0 2 0 0 0 1 0
  6     AGUA_CALIENTE_INDIANS 0 0 0 0 0 2 0 0 0 0
  7        NATURE_CONSERVANCY 0 0 0 0 0 0 2 0 0 0
  8        GOV_SCHWARZENEGGER 0 0 0 0 0 0 0 2 0 0
  9           MORONGO_INDIANS 0 0 1 0 1 0 0 0 2 0
 10            AUBURN_INDIANS 0 0 0 0 0 0 0 0 0 2
 11              PALA_INDIANS 0 0 0 0 0 0 0 0 0 0
 12            WINTUN_INDIANS 0 0 0 0 0 0 0 0 0 0
 13              L_JOHN_DOERR 0 0 0 0 0 0 0 0 0 0
 14                 ANN_DOERR 0 0 0 0 0 0 0 0 0 0
 15              ROBERT_MCKAY 0 0 0 0 0 0 0 0 0 0
```

Figure 11.9: *F-groups among California political donors (truncated)*

There happen to be two f-groups in these data. One is composed of strongly transitive ties, and is moderately large (seven members). "Group 3" (meaning that the first member of this group is node 3, the California Teacher's Association) contains a number of actors among whom all ties have the value 9 (the highest value in the graph). The members are listed in the top part of the output; the bottom part of the output shows the same results in matrix form, with "1" indicating co-presence in a weak component, and "2" indicating co-presence in a strongly transitive component. Our second component is a weakly transitive one, composed of the Building Industry Association and two large corporations (Chevron Oil and Hewlett-Packard). This is a grouping in which all the ties satisfy the criteria of weak transitivity.

11.3: Top-down Approaches

The approaches we've examined to this point start with the dyad, and see if this kind of tight structure can be extended outward. Overall structure of the network is seen as "emerging" from overlaps and couplings of smaller components. Certainly, this is a valid way of thinking about large structures and their component parts. The bottom-up approach may focus our attention on the underlying dynamic processes by which actors build networks.

Some might prefer, however, to start with the entire network as their frame of reference, rather than the dyad. Approaches of this type tend to look at the "whole" structure, and identify "sub-structures" as parts that are locally denser than the field as a whole. In a sense, this more macro lens is looking for "holes" or "vulnerabilities" or "weak spots" in the overall structure or solidarity of the network. These holes or weak spots define lines of division or cleavage in the larger group, and point to how it might be decomposed into smaller units. This top-down perspective leads us to think of dynamics that operate at the level of group-selection, and to focus on the constraints under which actors construct networks.

There are numerous ways that one might define the divisions and "weak spots" in a network. Below are some of the most common approaches.

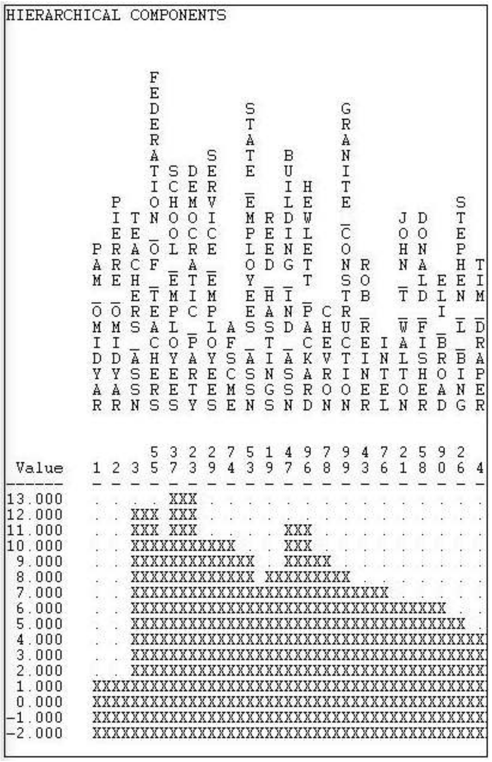
Components

Components of a graph are sub-graphs that are connected within, but disconnected between sub-groups. If a graph contains one or more "isolates", these actors are components. More interesting components are those which divide the network into separate parts, and where each part has several actors who are connected to one another (we pay no attention to how closely connected).

For directed graphs (in contrast to simple graphs), we can define two different kinds of components. A weak component is a set of nodes that are connected, regardless of the direction of ties. A strong component requires that there be a directed path from A to B in order for the two to be in the same component.

Since the Knoke information network has a single component, it isn't very interesting as an example. Let's look instead at the network of large donors to California political campaigns, where the strength of the relation between two actors is defined by the number of times that they contributed on the same side of an issue.

UCINET provides two algorithms for doing a census of components. *Network>Regions>Components>Simple graphs* is used for binary data. In addition to identifying the members of the components, it calculates a number of statistical measures of graph fragmentation. *Network>Regions>Components>Valued Graphs* can be used to examine the hierarchy of components as the cut-off value of tie strength is increasingly relaxed. Figure 11.10 shows partial results for the California political donations data.



```
HIERARCHICAL COMPONENTS

               F
               E
               D         S         G
               E         T         R
               R         A         A
               A     S   T   B     N
               T S D E   E   U     I
               I C E R   _   I H   T
           P   O H M V   E   L E   E           S
           I T N O O I   M R D W   _     J D   T
           E E _ O C C   P E I L   C     O O   E
         P R A O L R E   L E N E   O     H N   P
         A R C F _ A _   O D G T   N R   N A   H T
         M E H _ E T E   Y _ _ T   S O   _ L E E I
         _ _ E T M I M   E H I _   T B   T D L N M
         O O R E P C P   E A N P C R _   _ _ I _ _
         M M S A L _ L A S S D A H U R   W F _ L D
         I I _ C O P O F _ T _ C E C E I A I B _ R
         D D A H Y A Y S A I A K V T I N L S R B A
         Y Y S E E R E C S N S A R I N T T H O I P
         A A S R E T E M S G S R O O E E O E A N E
         R R N S S Y S E N S N D N N R L N R D G R

               5 3 2 2 7 5 1 4 9 7 9 4 7 2 5 9 2
 Value   1 2 3 5 7 3 9 4 3 9 7 6 8 9 3 6 1 8 0 6 4
------   - - - - - - - - - - - - - - - - - - - - -
13.000   . . . . XXX . . . . . . . . . . . . . . .
12.000   . . XXX XXX . . . . . . . . . . . . . . .
11.000   . . XXX XXX . . . . XXX . . . . . . . . .
10.000   . . XXXXXXXXXXX . . XXX . . . . . . . . .
 9.000   . . XXXXXXXXXXXXX . XXXXX . . . . . . . .
 8.000   . . XXXXXXXXXXXXX XXXXXXXXX . . . . . . .
 7.000   . . XXXXXXXXXXXXXXXXXXXXXXXXXXX . . . . .
 6.000   . . XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX . .
 5.000   . . XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX .
 4.000   . . XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 3.000   . . XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 2.000   . . XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 1.000   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 0.000   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-1.000   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-2.000   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Figure 11.10: *Weak component hierarchy for California political donors (truncated)*

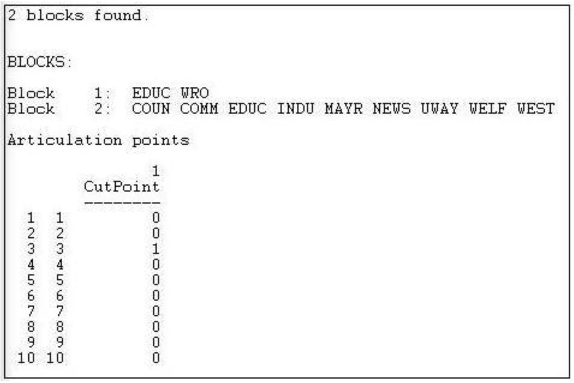
If we set a very high cut-off value of 13 issues in common, then our graph has only one non-isolate component (made up of the Democratic Party and the School Employees Union). Progressively lower cut-offs produce multiple, separate components until we reach a value of 7 issues in common. At this point, the non-isolated nodes all become connected into a single component.

Just as the strict definition of a "clique" may be too strong to capture the meaning of the concept of a maximal group, the notion of a component may be too strong to find all the meaningful weak-points, holes, and locally dense sub-parts of a larger graph. So, we will examine some more flexible approaches.

Blocks and Cutpoints (Bi-components)

An alternative approach to finding the key "weak" spots in a graphs is to ask: if a node were removed, would the structure become divided into un-connected parts? If there are such nodes, they are called "*cutpoints*". And, one can imagine that such cutpoints may be particularly important actors - who may act as brokers among otherwise disconnected groups. The divisions into which cutpoints divide a graph are called *blocks*. We can find the maximal non-separable sub-graphs (blocks) of a graph by locating the cutpoints. That is, we try to find the nodes that connect the graph (if there are any). Another name for a block is a "*bi-component*".

The UCINET algorithm *Network>Regions>Bi-Component* locates and identifies blocks and cutpoints. In Figure 11.11, we've applied it to the original Knoke symmetrized reciprocity data.



```
2 blocks found.

BLOCKS:

Block    1:  EDUC WRO
Block    2:  COUN COMM EDUC INDU MAYR NEWS UWAY WELF WEST

Articulation points

                 1
          CutPoint
          --------
 1  1            0
 2  2            0
 3  3            1
 4  4            0
 5  5            0
 6  6            0
 7  7            0
 8  8            0
 9  9            0
10 10            0
```

Figure 11.11: Cutpoints and blocks in the Knoke information network

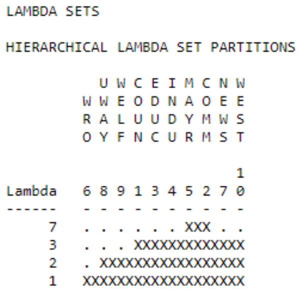
Two blocks are identified, with EDUC a member of both. This means that if EDUC (node 3) were removed, the WRO would become isolated. Node 3, then, is a cutpoint. You might want to verify this by eye, by glancing back at the graph of this network.

Components analysis locates parts of the graph that are disconnected; bi-components analysis locates parts that are vulnerable. Both approaches focus attention on key actors.

Lambda Sets and Bridges

An alternative approach is to ask if there are certain connections in the graph which, if removed, would result in a disconnected structure. In our example, the only relationship that qualifies is that between EDUC and WRO. But, since this would only lop off one actor, rather than really altering the network, it is not very interesting. However, it is possible to approach the question in a more sophisticated way. The Lambda set approach ranks each of the relationships in the network in terms of importance by evaluating how much of the flow among actors in the net go through each link. It then identifies sets of relationships which, if disconnected, would most greatly disrupt the flow among all of the actors. The math and computation is rather extreme, though the idea is fairly simple.

Network>Subgroups>Lambda Set locates the vulnerable "bridges" between pairs of actors. Figure 11.12 shows the results for the Knoke (reciprocity-symmetrized) information network.



```
LAMBDA SETS

HIERARCHICAL LAMBDA SET PARTITIONS

           U W C E I M C N W
         W W E O D N A O E E
         R A L U U D Y M W S
         O Y F N C U R M S T

                           1
Lambda   6 8 9 1 3 4 5 2 7 0
------   - - - - - - - - - -
     7   . . . . . . XXX . .
     3   . . . XXXXXXXXXXXXX
     2   . XXXXXXXXXXXXXXXXX
     1   XXXXXXXXXXXXXXXXXXX
```

Figure 11.12: *Lambda sets in the Knoke information network*

This approach identifies the #2 to #5 (MAYR to COMM) linkage as the most important one in the graph - in the sense that it carries a great deal of traffic, and the graph would be most disrupted if it were removed. This result can be confirmed by looking at the graph, where we see that most actors are connected to most other actors by way of the linkage between #2 and #5. Considerably less critical are linkages between 2 and 5 and actors 1, 3, 4, 7, and 10. Again, a glance at the figure shows these organizations to be a sort of "outer circle" around the core.

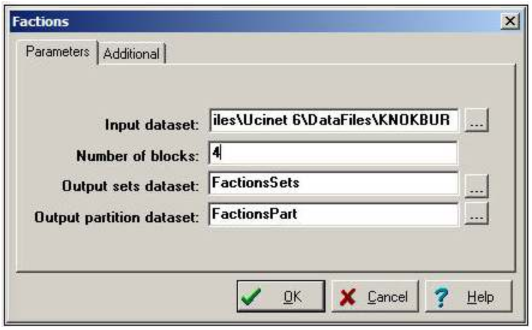
The lambda set idea has moved us quite far away from the strict components idea. Rather than emphasizing the "decomposition" or separation of the structure into unconnected components, the lambda set idea is a more "continuous" one. It highlights points at which the fabric of connection is most vulnerable to disruption.

Factions

Imagine a society in which each person was closely tied to all others in their own sub-population (that is, all sub-populations are cliques), and there are no connections at all among sub-populations (that is, each sub-population is a component). Most real populations do not look like this, but the "ideal type" of complete connection within and complete disconnection between sub-groups is a useful reference point for assessing the degree of "factionalization" in a population.

If we took all the members of each "faction" in this ideal-typical society, and put their rows and columns together in an adjacency matrix (i.e. permuted the matrix), we would see a distinctive pattern of "1-blocks" and "0-blocks". All connections among actors within a faction would be present, all connections between actors in different factions would be absent.

Network>Subgroups>Factions is an algorithm that finds the optimal arrangement of actors into factions to maximize similarity to the ideal type, and measures how well the data actually fit the ideal type. Figure 11.13 shows the dialog for using this tool.



Factions

Parameters | Additional

Input dataset: iles\Ucinet 6\DataFiles\KNOKBUR

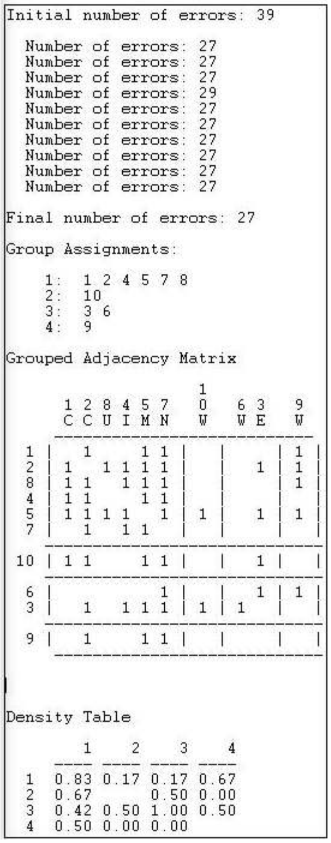
Number of blocks: 4

Output sets dataset: FactionsSets

Output partition dataset: FactionsPart

Figure 11.13: *Dialog for Network>Subgroups>Factions*

Notice that you must specify how many factions (blocks) you would like the algorithm to find. If you have a prior hypothesis that a given population was divided into two factions, you could "test" this hypothesis by seeing how much error remained after defining two optimal factions. More commonly, we might use this tool in an exploratory way, examining the results from several runs with differing numbers of factions. As with any exploratory technique, it is a matter of judgment which solution is most helpful. After running several alternative numbers of blocks, we settled on four as meaningful for our purposes. This result is shown in Figure 11.14.



```
Initial number of errors: 39

  Number of errors: 27
  Number of errors: 27
  Number of errors: 27
  Number of errors: 29
  Number of errors: 27
  Number of errors: 27
  Number of errors: 27
  Number of errors: 27
  Number of errors: 27
  Number of errors: 27

Final number of errors: 27

Group Assignments:

    1:  1 2 4 5 7 8
    2:  10
    3:  3 6
    4:  9

Grouped Adjacency Matrix

                    1
      1 2 8 4 5 7   0   6 3   9
      C C U I M N   W   W E   W
     ----------------------------
  1 |   1     1 1 |   |     | 1 |
  2 | 1   1 1 1 1 |   |   1 | 1 |
  8 | 1 1   1 1 1 |   |     | 1 |
  4 | 1 1     1 1 |   |     |   |
  5 | 1 1 1 1   1 | 1 |   1 | 1 |
  7 |   1   1 1   |   |     |   |
    -----------------------------
 10 | 1 1     1 1 |   |   1 |   |
    -----------------------------
  6 |           1 |   |   1 | 1 |
  3 |   1   1 1 1 | 1 | 1   |   |
    -----------------------------
  9 |   1     1 1 |   |     |   |
     ----------------------------


Density Table

        1    2    3    4
     ---- ---- ---- ----
  1  0.83 0.17 0.17 0.67
  2  0.67      0.50 0.00
  3  0.42 0.50 1.00 0.50
  4  0.50 0.00 0.00
```

Figure 11.14: *Four-faction solution for the directed Knoke information network*

The "Final number of errors" can be used as a measure of the "goodness of fit" of the "blocking" of the matrix. This count (27 in this case) is the sum of the number of zeros within factions (where all the ties are supposed to be present in the ideal type) plus the number of ones in the non-diagonal blocks (ties between members of different factions, which are supposed to be absent in the ideal type). Since there are 49 total ties in our data, being wrong on the locations of 27 is not a terribly good fit. It is, however, the best we can do with four "factions".

The four factions are identified, and we note that two of them are individuals (10, 9) and one is a dyad (3,6).

The "blocked" or "grouped" adjacency matrix shows a picture of the solution. We can see that there is quite a lot of density "off the main diagonal" where there shouldn't be any. The final panel of the results reports the "block densities" as the number of ties that are present in blocks as proportions of all possible ties.

This approach corresponds nicely to the intuitive notion that the groups of a graph can be defined by a combination of local high density and the presence of "structural holes" between some sets of actors and others. The picture then not only identifies actual or potential factions, but also tells us about the relations among the factions - potential allies and enemies, in some cases.

11.E: Cliques and Sub-groups (Exercises)

Review Questions

1. Can you explain the term "maximal complete sub-graph"?
2. How do n-cliques and n-clans "relax" the definition of a clique?
3. Give an example of when it might be more useful to use an n-clique or n-clan approach instead of a strict clique.
4. How do k-plexes and k-cores "relax" the definition of a clique?
5. Give an example of when it might be more useful to use a k-plex or k-core approach instead of a strict clique.
6. What is a component of a graph?
7. How does the idea of a "block" relax the strict definition of a component?
8. Are there any cutpoints in the "star" network? In the "line" network? In the "circle" network?
9. How does the idea of a lambda set relax the strict definition of a component?
10. Are there any "bridges" in a strict hierarchy network?

Application Questions

1. Think of the readings from the first part of the course. Which studies used the ideas of group sub-structures? What kinds of approaches were used: cliques, clans, plexes, etc.?
2. Try to apply the notion of group sub-structures at different levels of analysis. Are there sub-structures within the kinship group of which you are a part? How is the population of Riverside divided into sub-structures? Are there sub-structures in the population of Universities in the United States? Are the nations in the world system divided into sub-structures in some way?
3. How might the lives of persons who are "cutpoints" be affected by having this kind of a structural position? Can you think of an example?
4. Can you think of a real-world (or literary) example of a population with sub-structures? How might the sub-structures in your real world case be described using the formal concepts (are the sub-structures "clans" or "factions", etc.).

11.S: Cliques and Sub-groups (Summary)

One of the most interesting things about social structures is their sub-structure in terms on groupings or cliques. The number, size, and connections among the sub-groupings in a network can tell us a lot about the likely behavior of the network as a whole. How fast will things move across the actors in the network? Will conflicts most likely involve multiple groups, or two factions? To what extent do the sub-groups and social structures overlap one another? All of these aspects of sub-group structure can be very relevant to predicting the behavior of the network as a whole.

The location of individuals in nets can also be thought of in terms of cliques or sub-groups. Certain individuals may act as "bridges" among groups, others may be isolates; some actors may be cosmopolitans, and others locals in terms of their group affiliations. Such variation in the ways that individuals are connected to groups or cliques can be quite consequential for their behavior as individuals.

In this section we have briefly reviewed some of the most important definitions of "sub-groups" or "cliques" and examined the results of applying these definitions to a set of data. We have seen that different definitions of what a clique is can give rather different pictures of the same reality.

CHAPTER OVERVIEW

12: Positions and Roles - The Idea of Equivalence

12.1: Introduction to the Idea of Equivalence

We have been examining some of the ways that structural analysts look at network data. We began by looking for patterns in the overall structure (e.g. connectedness, density, etc.) and the embeddedness of each actor (e.g. geodesic distances, centrality). Next, we introduced a second major way of going about examining network data by looking for "sub-structure", or groupings of actors that are closer to one another than they are to other groupings. For example, we looked at the meaning of "cliques", "blocks", and "bridges" as ways of thinking about and describing how the actors in a network may be divided into sub-groups on the basis of their patterns of relations with one another.

All of this, while sometimes a bit technical, is pretty easy to grasp conceptually. The central node of a "star" network is "closer" to all other members than any other member - a simple (if very important) idea that we can grasp. A clique as a "maximal complete sub-graph" sounds tough, but, again, is easy to grasp. It is simply the biggest collection of folks who all have connections with everyone else in the group. Again, the idea is not difficult to grasp, because it is really quite concrete: we can see and feel cliques.

Now we are going to turn our attention to somewhat more abstract ways of making sense of the patterns of relations among social actors: the analysis of "equivalence classes". Being able to define, theorize about, and analyze data in terms of equivalence is important because we want to be able to make generalizations about social behavior and social structure. That is, we want to be able to state principles that hold for all groups, all organizations, all societies, etc. To do this, we must think about actors not as individual unique persons (which they are), but as examples of categories - sets of actors who are, in some defined way, "equivalent". As an empirical task, we need to be able to group together actors who are the most similar, and to describe what makes them similar, and to describe what makes them different, as a category, from members of other categories.

Sociological thinking uses abstract categories routinely. "Working class", "middle class", "upper class" are one such set of categories that describe social positions. "Men" and "women" are really labels for categories of persons who are more similar within category than between category - at least for the purposes of understanding and predicting some aspects of their social behavior. When categories like these are used as parts of sociological theories, they are being used to describe the "social roles" or "social positions" typical of members of the category.

Many of the category systems used by sociologists are based on "attributes" of individual actors that are in common across actors. If I state that "European-American males, ages 45-64 are likely to have relatively high incomes" I am talking about a group of people who are demographically similar - they share certain attributes (maleness, European ancestry, biological age, and income). Structural analysis is not particularly concerned with systems of categories (i.e. variables) that are based on descriptions of similarity of individual attributes (some radical structural analysis would even argue that such categories are not really "sociological" at all). Structural analysis seek to define categories and variables in terms of similarities of the patterns of relations among actors, rather than attributes of actors. That is, the definition of a category, or a "social role" or "social position" depends upon its relationship to another category. Social roles and positions, structural analysts argue, are inherently "relational". That's pretty abstract in itself. Some examples can make the point.

What is the social role "husband"? One useful way to think about it is as a set of patterned interactions with a member or members of some other social categories: "wife" and "child" (and probably others). Each one of these categories (i.e. husband, wife, child) can only be defined by regularities in the patterns of relationships with members of other categories (there are a number of types of relations here - monetary, emotional, ritual, sexual, etc.). That is, family and kinship roles are inherently relational. The network analyst translates this idea by saying that there are "equivalence classes" of husband, wife, child, etc.

What is a "worker"? We could mean a person who does labor (an attribute, actually one shared by all humans). A more sociologically interesting definition was given by Marx as a person who sells control of their labor power to a capitalist. Note that the meaning of "worker" depends upon a capitalist - and vice versa. It is the relation (in this case, as Marx would say, a relation of exploitation) between occupants of the two roles that defines the meaning of the roles.

The point is: to the structural analyst, the building blocks of social structure are "social roles" or "social positions". These social roles or positions are defined by regularities in the patterns of relations among actors, not attributes of the actors themselves. We identify and study social roles and positions by studying relations among actors, not by studying attributes of individual actors. Even things that appear to be "attributes of individuals" such as race, religion, and age can be thought of as short-hand labels for patterns of relations. For example, "white" as a social category is really a short-hand way of referring to persons who typically have a common form of relationships with members of another category - "non-whites". Things that might at first appear to be attributes of individuals are really just ways of saying that an individual falls in a category that has certain patterns of characteristic relationships with members of other categories.

12.2: Approaches to Network Positions and Social Roles

Because "positions" or "roles" or "social categories" are defined by "relations" among actors, we can identify and empirically define social positions using network data. In an intuitive way, we would say that two actors have the same "position" or "role" to the extent that their pattern of relationships with other actors is the same. But, there are a couple things about this intuitive definition that are troublesome.

First, what relations do we take into account, among whom, in seeking to identify which actors are similar and which are not? The relations that I have with the university (such as "Professor") are similar in some ways to the relations that my students have with the university: we are both governed by many of the same rules, practices, and procedures. The relations I have with the university are very different from those of my students in some ways (e.g. the university pays me, students pay the university). Which relations should count and which ones not, in trying to describe the roles of "professor" and "student"? Indeed, why am I examining relations among my students, me, and the university, instead of including, say, members of the state legislature? There is no simple answer about what the "right relations" are to examine; and, there is no simple answer about who the relevant set of "actors" are. It all depends upon the purposes of our investigation, the theoretical perspective we are using, and the populations to which we would like to be able to generalize our findings. Social network data analytic methods are of little use in answering these conceptual questions.

The second problem with our intuitive definition of a "role" or "position" is this: assuming that I have a set of actors and a set of relations that make sense for studying a particular question, what do I mean that actors who share the same position are similar in their pattern of relationships or ties? The idea of "similarity" has to be rather precisely defined. Again, there is no single and clear "right" answer for all purposes of investigation. But, there are rigorous ways of thinking about what it means to be "similar" and there are rigorous ways of actually examining data to define social roles and social positions empirically. These are the issues where there are some ways in which widely used methods can provide guidance.

12.3: Defining Equivalence or Similarity

What do we mean when we say that two actors have "similar" patterns of relations, and hence are both members of the same role or social position? Network analysts most broadly defines two nodes (or other more elaborate structures) as similar if they fall in the same "equivalence class". Frankly, that's no immediate help. But it does say that there is something that would cause us to say two actors (or other structures) are members of a "class" that is different from other "classes".

Now it becomes a question of what features of an actor's position place them into a "class" with other actors? In what way are they "equivalent"?

There are many ways in which actors could be defined as "equivalent" based on their relations with others. For example, we could create two "equivalence classes" of actors with out-degree of zero, and actors with out-degree of more than zero. Indeed, a very large number of the algorithms we've examined group sets of actors into categories based on some commonality in their positions in graphs.

Three particular definitions of "equivalence" have been particularly useful in applying graph theory to the understanding of "social roles" and "structural positions". We will look at these in the next three sections on "structural equivalence", "automorphic equivalence", and "regular equivalence". Of these, "automorphic" has rarely been used in substantive work.

The basic ideas of these three kinds of equivalence are easily illustrated with a simple graph (developed by Wasserman and Faust). Consider Figure 12.1, a simple graph of the relations among nine actors "A" to "I".

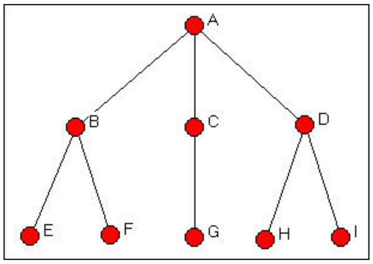


***Figure 12.1:** Wasserman-Faust network to illustrate equivalence classes*

This graph provides particularly clear examples of how structural, automorphic, and regular equivalence differ. Let's look in more detail at these ideas, starting with the most restrictive notion of what it means for actors to be equivalent.

Structural Equivalence

Two nodes are said to be exactly structurally equivalent if they have the same relationships to all other nodes. Structural equivalence is easy to grasp (though it can be operationalized in a number of ways) because it is very specific: two actors must be exactly substitutable in order to be structurally equivalent.

In Figure $12.3.1$, there are seven "structural equivalence classes". Can you find them?

- There is no actor who has exactly the same set of ties as actor A (ties to B, C, and D), so actor A is in a class by itself.
- The same is true for actors B, C, and D. Each of these actors has a unique set of ties to others, so they form three classes, each with one member.
- E and F, however, fall in the same structural equivalence class. Each has a single tie; and that tie is to actor B. Since E and F have exactly the same pattern of ties with all other actors, they are structurally equivalent.
- Actor G, again, is in a class by itself. Its profile of ties with the other nodes in the diagram is unique.
- Finally, actors H and I fall in the same structural equivalence class. That is, they have exactly the same pattern of ties to all other actors.

Actors that are structurally equivalent are in identical "positions" in the structure of the diagram. Whatever opportunities and constraints operate on one member of a class are also present for the others. The nodes in a structural equivalence class are, in a sense, in the same position with regard to all other actors.

Because exact structural equivalence is likely to be rare (particularly in large networks), we often are interested in examining the degree of structural equivalence, rather than the simple presence or absence of exact equivalence.

Structural equivalence is the "strongest" form of equivalence that network analysts usually consider. If we soften the requirements just a bit, we can often find some interesting other patterns of equivalence.

Automorphic Equivalence

The idea of structural equivalence is powerful because it identifies actors that have the same position, or who are completely substitutable. But, even intuitively, you can probably imagine other "less strict" definitions of what it means for two actors to be similar or equivalent.

Suppose that the graph in Figure $12.3.1$ described a franchise group of hamburger restaurants. Actor A is the central headquarters, actors B, C, and D are the managers of three different stores. Actors E and F are workers at one store; G is the lone worker at a second store; H and I are workers at the third store.

Even though actor B and actor D are not structurally equivalent (they doe have the same boss, but not the same workers), they do seem to be "equivalent" in a different sense. Both manager B and manager D report to a boss (in this case, the same boss), and each has exactly two workers. These are different people, but the two managers seem somehow equivalent. If we swapped them, and also swapped the four workers, all of the distances among all the actors in the graph would be exactly identical. In fact, actors B and D form an "automorphic" equivalence class.

In Figure $12.3.1$, there are actually five automorphic equivalence classes: {A}, {B, D}, {C}, {E, F, H, I}, and {G}. These classes are groupings whose members would remain at the same distance from all other actors if they were swapped, and, members of other classes were also swapped.

The idea of automorphic equivalence is that sets of actors can be equivalent by being embedded in local structures that have the same patterns of ties - "parallel" structures. Large scale populations of social actors (perhaps like hamburger restaurant chains) can display a great deal of this sort of "structural replication". The faces are different, but the structures are identical.

Note that the less strict definition of "equivalence" had reduced the number of classes. If we are willing to go one important step further, we can reduce the complexity still further.

Regular Equivalence

Two nodes are said to be regularly equivalent if they have the same profile of ties with other sets of actors that are also regularly equivalent. This is a complicated way of saying something that we recognize intuitively.

Two mothers, for example, are "equivalent" because each has a certain pattern of ties with a husband, children, and in-laws (for one example - but one that is very culturally relative). The two mothers do not have ties to the same husband (usually) or the same children or in-laws. That is, they are not "structurally equivalent". Because different mothers may have different numbers of husbands, children, and in-laws, they will not be automorphically equivalent. But they are similar because they have the same relationships with some member or members of another set of actors (who are themselves regarded as equivalent because of the similarity of their ties to a member of the set "mother").

This is an obvious notion, but a critical one. Regular equivalence sets describe the "social roles" that are the basic building blocks of all social institutions. Actors that are regularly equivalent to not necessarily fall in the same network positions or locations with respect to other individual actors; rather, they have the same kinds of relationships with some members of other sets of actors.

In Figure $12.3.1$, there are three regular equivalence classes. The first is actor A; the second is composed of the three actors B, C, and D; the third is composed of the remaining five actors E, F, G, H, and I.

The easiest class to see is the give actors across the bottom of the figure (E, F, G, H, and I). These actors are regularly equivalent to one another because a) they have no tie with any actor in the first class (that is, with actor A) and b) each has a tie with an actor in the second class (either B, C, or D). Each of the five actors, then, has an identical pattern of ties with actors in the other classes.

Actors B, C, and D form a class because a) they have a tie with a member of the first class (that is, with actor A) and b) they each have a tie with a member of the third class. B and D actually have ties with two members of the third class, whereas actor C has a tie to only one member of the third class; this doesn't matter, as there is a tie to some member of the third class.

Actor A is in a class by itself, defined by a) a tie to at least one member of class two and b) no tie to any member of class three.

As with structural and automorphic equivalence, exact regular equivalence may be rare in a large population with many equivalence classes. Approximate regular equivalence can be very meaningful though, because it gets at the notion of which actors fall in which social roles, and how social roles (not role occupants) relate to one another.

12.E: Positions and Roles - The Idea of Equivalence (Exercises)

Review Questions

1. How are network roles and social roles different from network "sub-structures" as ways of describing social networks?
2. Explain the differences among structural, automorphic, and regular equivalence.
3. Actors who are structurally equivalent have the same patterns of ties to the same other actors. How do correlation, distance, and match measures index this kind of equivalence or similarity?
4. If the adjacency matrix for a network can be blocked into perfect sets of structurally equivalent actors, all blocks will be filled with zeros or with ones. Why is this?
5. If two actors have identical geodesic distances to all other actors, they are (probably) automorphically equivalent. Why does having identical distances to all other actors make actors "substitutable" but not necessarily structurally equivalent?
6. Regularly equivalent actors have the same pattern of ties to the same kinds of other actors - but not necessarily the same distances to all other actors, or ties to the same other actors. Why is this kind of equivalence particularly important in sociological analysis?

Application Questions

1. Think of the readings from the first part of the course. Did any studies use the idea of structural equivalence or network role? Did any studies use the idea of regular equivalence or social role?
2. Think about the star network. How many sets of structurally equivalent actors are there? What are the sets of automorphically equivalent actors? Regularly equivalent actors? What about the circle network?
3. Examine the line network carefully - this one's a little more tricky. Describe the structural equivalence and regular equivalence sets in a line network.
4. Consider our classical hierarchical bureaucracy, defined by a network of directed ties of "order giving" from the top to the bottom. Make an adjacency matrix for a simple bureaucracy like this. Block the matrix according to the regular equivalence sets; block the matrix according to structural equivalence sets. How (and why) do these blockings differ? How to the permuted matrices differ?
5. Think about some social role (e.g. "mother"). What would you say are the kinds of ties with other social roles that could be used to identify which persons in a population were "mothers" and which were not? Note the relational character of social roles - one social role can only be defined with respect to others. Provide some examples of social roles from an area of interest to you.

12.S: Positions and Roles - The Idea of Equivalence (Summary)

The three types of equivalence (structural, automorphic, and regular) have progressively less strict definitions of what it means for two actors to be "equivalent". And, as we make the definitions less strict (which is not the same as making them less precise!), we are able to understand social networks at increasing levels of abstraction.

Structural equivalence is the most "concrete" form of equivalence. Two actors are exactly structurally equivalent if they have exactly the same ties to exactly the same other individual actors. Pure structural equivalence can be quite rare in social relations, but approximations to it may not be so rare. In studying a single population, two actors who are approximately structurally equivalent are facing pretty much the same sets of constraints and opportunities. Commonly we would say that two actors who are approximately structurally equivalent are in approximately the same position in a structure.

Automorphic equivalence is a bit more relaxed. Two actors may not be tied to the same others, but if they are embedded in the same way in the larger structure, they are equivalent. With automorphic equivalence, we are searching for classes of actors who are at the same distance from other sets of actors - that is, we are trying to find parallel or substitutable sub-structures (rather than substitutable individual actors).

Regular equivalence deserves special attention because it gets at the idea of the "role" that an actor plays with respect to occupants of other "roles" in a structure. The idea of a social role, which is "institutionalized" by normative and sanctioned relationships to other roles is at the very core of the entire sociological perspective.

The definitions of the forms of equivalence discussed here are quite precise (though my discussion doesn't have much mathematical rigor). The notions of equivalence provide quite rigorous ways of defining and thinking about core analytical tools in sociology - individuals' positions in groups, types of structures, and social roles. This is a huge advance over the sometimes quite imprecise and contradictory verbal treatments found in much of our literature.

But, real world social networks are often quite messy, may not be fully realized (that is, not in equilibrium), and/or may be badly measured. The search for equivalence in real data can be a somewhat complicated matter with a number of vexing choices to be made. We'll spend some time with these practical issues in the next three chapters.

CHAPTER OVERVIEW

13: Measures of Similarity and Structural Equivalence

In this rather lengthy chapter we are going to do three things. First, we will focus on how we can measure the similarity of actors in a network based on their relations to other actors.Second, we will very quickly look at two tools that are very commonly used for visualizing the patterns of similarity and dissimilarity/distance among actors. Third, we will examine the most commonly used approaches for finding structural equivalence classes.

13.1: Introduction to Measures of Similarity and Structural Equivalence

In this rather lengthy chapter we are going to do three things.

First, we will focus on how we can measure the similarity of actors in a network based on their relations to other actors. The whole idea of "equivalence" that we discussed in the last chapter is an effort to understand the pattern of relationships in a graph by creating classes, or groups of actors who are "equivalent" in one sense or another. All of the methods for identifying such groupings are based on first measuring the similarity or dissimilarity of actors, and then searching for patterns and simplifications. We will first review the most common approaches to indexing the similarities of actors based on their relations with other actors.

Second, we will very quickly look at two tools that are very commonly used for visualizing the patterns of similarity and dissimilarity/distance among actors. Multi-dimensional scaling and hierarchical cluster analysis are widely used tools for both network and non-network data. They are particularly helpful in visualizing the similarity or distance among cases, and for identifying classes of similar cases.

Third, we will examine the most commonly used approaches for finding structural equivalence classes. That is, methods for identifying groups of nodes that are similar in their patterns of ties to all other nodes. These methods (and those for other kinds of "equivalence" in the next two chapters) use the ideas of similarity/distance between actors as their starting point, and these methods most often use clustering and scaling as a way of visualizing results. In addition, the "block model" is also commonly used to describe structural similarity classes.

13.2: Measuring Similarity/Dissimilarity

We might try to assess which nodes are most similar to which other nodes intuitively by looking at a graph. We would notice some important things. It would seem that actors 2, 5 and 7 might be structurally similar in that they seem to have reciprocal ties with each other and almost everyone else. Actors 6, 8, and 10 are "regularly" similar in that they are rather isolated, but they are not structurally similar because they are connected to quite different sets of actors. But, beyond this, it is really rather difficult to assess equivalence rigorously by just looking at a diagram.

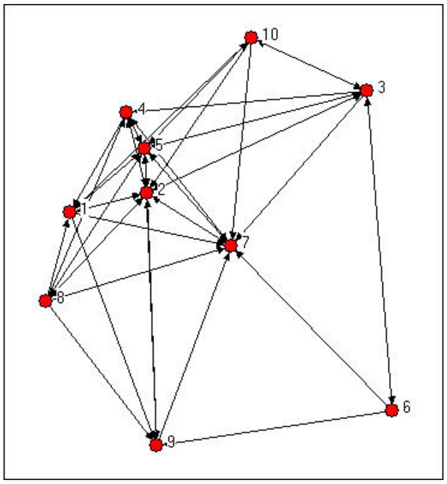
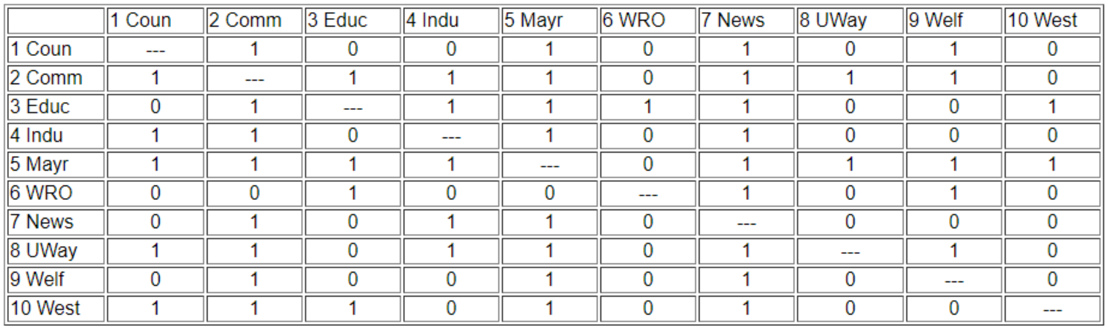


Figure 13.1: *Knoke directed information network*

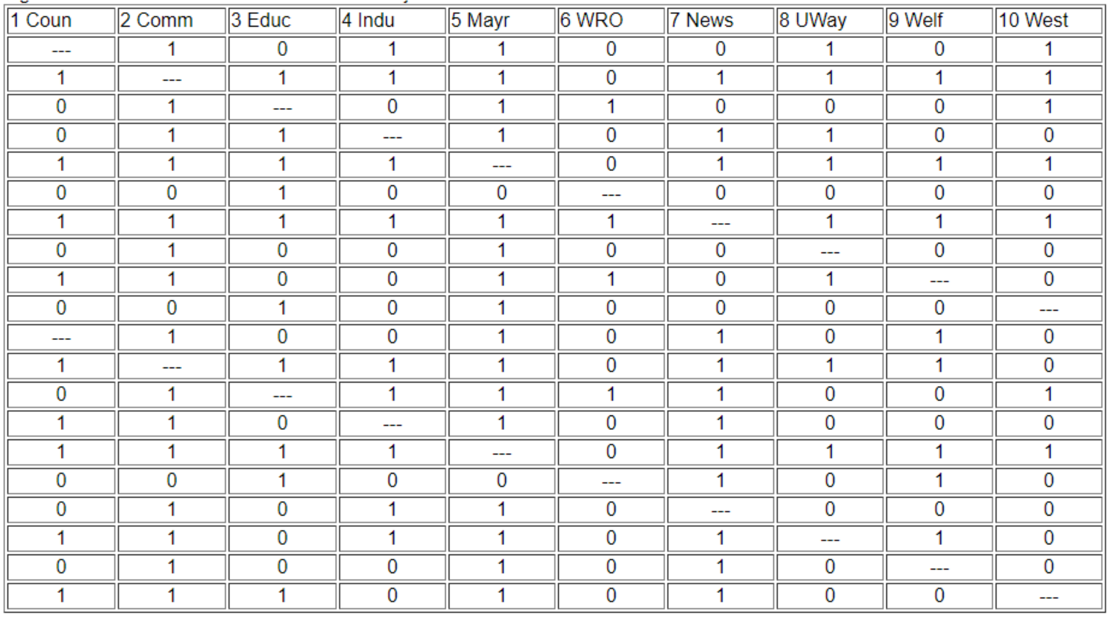
We can be a lot more precise in assessing similarity if we use the matrix representation of the network instead of the diagram. This also lets us use the computer to do some of the quite tedious jobs involved in calculating index numbers to assess similarity. The original data matrix has been reproduced below as Figure 13.2. Many of the features that were apparent in the diagram are also easy to grasp in the matrix. If we look across the rows and count out-degrees, and if we look down the columns (to count in-degree) we can see who the central actors are and who are the isolates. But, even more generally, we can see that two actors are structurally equivalent to the extent that the profile of scores in their rows and columns are similar. Finding automorphic equivalence and regular equivalence is not so simple. But, since these other forms are less restrictive (and hence simplifications of the structural classes), we begin by measuring how similar each actor's ties are to all other actors.



	1 Coun	2 Comm	3 Educ	4 Indu	5 Mayr	6 WRO	7 News	8 UWay	9 Welf	10 West
1 Coun	---	1	0	0	1	0	1	0	1	0
2 Comm	1	---	1	1	1	0	1	1	1	0
3 Educ	0	1	---	1	1	1	1	0	0	1
4 Indu	1	1	0	---	1	0	1	0	0	0
5 Mayr	1	1	1	1	---	0	1	1	1	1
6 WRO	0	0	1	0	0	---	1	0	1	0
7 News	0	1	0	1	1	0	---	0	0	0
8 UWay	1	1	0	1	1	0	1	---	1	0
9 Welf	0	1	0	0	1	0	1	0	---	0
10 West	1	1	1	0	1	0	1	0	0	---

Figure 13.2: *Adjacency matrix for Knoke information network*

Two actors may be said to be structurally equivalent if they have the same patterns of ties with other actors. This means that the entries in the rows and columns for one actor are identical to those of another. If the matrix were symmetric, we would need only to scan pairs of rows (or columns). But, since these data are on directed ties, we should examine the similarity of sending and receiving of ties (of course, we might be interested in structural equivalence with regard to only sending, or only receiving ties). We can see the similarity of the actors if we expand the matrix in Figure 13.2 by listing the row vectors followed by the column vectors for each actor as a single column, as we have in Figure 13.3.



1 Coun	2 Comm	3 Educ	4 Indu	5 Mayr	6 WRO	7 News	8 UWay	9 Welf	10 West
---	1	0	1	1	0	0	1	0	1
1	---	1	1	1	0	1	1	1	1
0	1	---	0	1	1	0	0	0	1
0	1	1	---	1	0	1	1	0	0
1	1	1	1	---	0	1	1	1	1
0	0	1	0	0	---	0	0	0	0
1	1	1	1	1	1	---	1	1	1
0	1	0	0	1	0	0	---	0	0
1	1	0	0	1	1	0	1	---	0
0	0	1	0	1	0	0	0	0	---
---	1	0	0	1	0	1	0	1	0
1	---	1	1	1	0	1	1	1	0
0	1	---	1	1	1	1	0	0	1
1	1	0	---	1	0	1	0	0	0
1	1	1	1	---	0	1	1	1	1
0	0	1	0	0	---	1	0	1	0
0	1	0	1	1	0	---	0	0	0
1	1	0	1	1	0	1	---	1	0
0	1	0	0	1	0	1	0	---	0
1	1	1	0	1	0	1	0	0	---

Figure 13.3: *Concatenated row and column adjacencies for Knoke information network*

The ties of each actor (both in and out) are now represented as a column of data. We can now measure the similarity of each pair of columns to index the similarity of the two actors, forming a pair-wise matrix of similarities. We could also get at the same idea in reverse, by indexing the dissimilarity or "distance" between the scores in any two columns.

There are any number of ways to index similarity and distance. In the next two sections we'll briefly review the most commonly used approaches when the ties are measured as values (i.e. strength or cost or probability) and as binary.

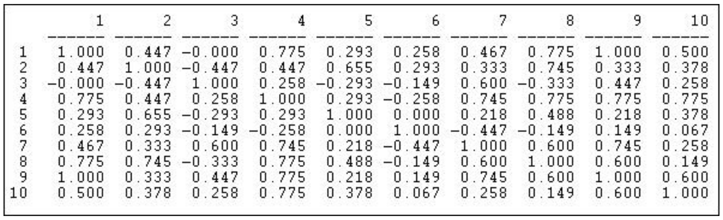
The goal here is to create an actor-by-actor matrix of the similarity (or distance) measures. Once we have done this, we can apply other techniques for visualizing the similarities in the actor's patterns of relations with other actors.

Valued Relations

A common approach for indexing the similarity of two valued variables is the degree of linear association between the two. Exactly the same approach can be applied to the vectors that describe the relationship strengths of two actors to all other actors. As with any measures of linear association, linearity is a key assumption. It is often wise, even when data are at the interval level (e.g. volume of trade from one nation to all others) to consider measures with weaker assumptions (like measures of association designed for ordinal variables).

Pearson Correlation Coefficients, Covariances, and Cross-Products

The correlation measure of similarity is particularly useful when the data on ties are "valued", that is, tell us about the strength and direction of association, rather than simple presence or absence. Pearson correlations range from -1.00 (meaning that the two actors have exactly the opposite ties to each other), through zero (meaning that knowing one actor's tie to a third party doesn't help us at all in guessing what the other actor's tie to the third party might be), to +1.00 (meaning that the two actors always have exactly the same tie to other actors - perfect equivalence). Pearson correlations are often used to summarize pair-wise structural equivalence because the statistic (called "little r") is widely used in social statistics. If the data on ties are truly nominal, or if density is very high or very low, correlations can sometimes be a little troublesome, and matches (see below) should also be examined. Different statistics, however, usually give very much the same answers. Figure 13.4 shows the correlations of the ten Knoke organization's profiles of in and out information ties. We are applying correlation, even though the Knoke data are binary. The UCINET algorithm *Tools>Similarities* will calculate correlations for rows or columns.



```
            1      2      3      4      5      6      7      8      9     10
       ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
 1      1.000  0.447 -0.000  0.775  0.293  0.258  0.467  0.775  1.000  0.500
 2      0.447  1.000 -0.447  0.447  0.655  0.293  0.333  0.745  0.333  0.378
 3     -0.000 -0.447  1.000  0.258 -0.293 -0.149  0.600 -0.333  0.447  0.258
 4      0.775  0.447  0.258  1.000  0.293 -0.258  0.745  0.775  0.775  0.775
 5      0.293  0.655 -0.293  0.293  1.000  0.000  0.218  0.488  0.218  0.378
 6      0.258  0.293 -0.149 -0.258  0.000  1.000 -0.447 -0.149  0.149  0.067
 7      0.467  0.333  0.600  0.745  0.218 -0.447  1.000  0.600  0.745  0.258
 8      0.775  0.745 -0.333  0.775  0.488 -0.149  0.600  1.000  0.600  0.149
 9      1.000  0.333  0.447  0.775  0.218  0.149  0.745  0.600  1.000  0.600
10      0.500  0.378  0.258  0.775  0.378  0.067  0.258  0.149  0.600  1.000
```

Figure 13.4: *Pearson correlations of rows (sending) for Knoke information network*

We can see, for example, that node 1 and node 9 have identical patterns of ties; there is a moderately strong tendency for actor 6 to have ties to actors that actor 7 does not, and vice versa.

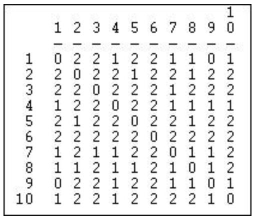
The Pearson correlation measure does not pay attention to the overall prevalence of ties (the mean of the row or column), and it does not pay attention to differences between actors in the variances of their ties. Often this is desirable - to focus only on the pattern, rather than the mean and variance as aspects of similarity between actors.

Often though, we might want our measure of similarity to reflect not only the pattern of ties, but also differences among actors in their overall tie density. *Tools>Similarities* will also calculate the *covariance* matrix. If we want to include differences in variances across actors as aspects of (dis)similarity, as well as means, the *cross-product* ratio calculated in *Tools>Similarities* might be used.

Euclidean, Manhattan, and Squared Distances

An alternative approach to linear correlation (and its relatives) is to measure the "distance" or "dissimilarity" between the tie profiles of each pair of actors. Several "distance" measures are fairly commonly used in network analysis, particularly the Euclidean distance or squared Euclidean distance. These measures are not sensitive to the linearity of association and can be used with either valued or binary data.

Figure 13.5 shows the Euclidean distances among the Knoke organizations calculated using *Tools>Dissimilarities and Distances>Std Vector dissimilarities/distances*.



```
                       1
     1 2 3 4 5 6 7 8 9 0
     - - - - - - - - - -
 1   0 2 2 1 2 2 1 1 0 1
 2   2 0 2 2 1 2 2 1 2 2
 3   2 2 0 2 2 2 1 2 2 2
 4   1 2 2 0 2 2 1 1 1 1
 5   2 1 2 2 0 2 2 1 2 2
 6   2 2 2 2 2 0 2 2 2 2
 7   1 2 1 1 2 2 0 1 1 2
 8   1 1 2 1 1 2 1 0 1 2
 9   0 2 2 1 2 2 1 1 0 1
10   1 2 2 1 2 2 2 2 1 0
```

Figure 13.5: *Euclidean distances in sending for Knoke information network*

The Euclidean distance between two vectors is equal to the square root of the sum of the squared differences between them. That is, the strength of actor A's tie to C is subtracted from the strength of actor B's tie to C, and the difference is squared. This is then repeated across all the other actors (D, E, F, etc.), and summed. The square root of the sum is then taken.

A closely related measure is the "Manhattan" or block distance between the two vectors. This distance is simply the sum of the absolute difference between the actor's ties to each alter, summed across the alters.

Binary Relations

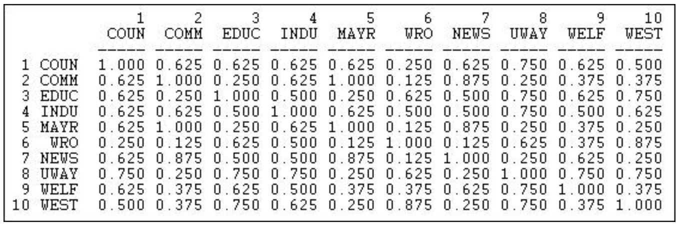
If the information that we have about the ties among our actors is binary, correlation and distance measures can be used, but may not be optimal. For data that are binary, it is more common to look at the vectors of two actor's ties, and see how closely the entries in one "match" the entries in the other.

There are several useful measures of tie profile similarity based on the matching idea that are calculated by *Tools>Similarities*.

Matches: Exact, Jaccard, Hamming

A very simple and often effective approach to measuring the similarity of two tie profiles is to count the number of times that actor A's tie to alter is the same as actor B's tie to alter, and express this as a percentage of the possible total.

Figure 13.6 shows the result for the columns (information receiving) relation of the Knoke bureaucracies.



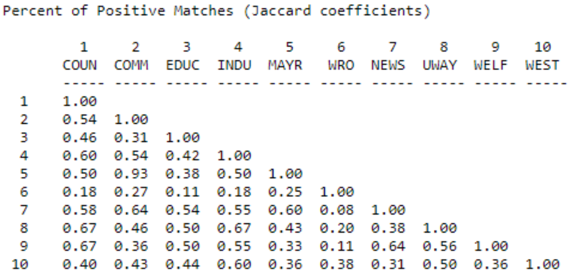
```
             1     2     3     4     5     6     7     8     9    10
          COUN  COMM  EDUC  INDU  MAYR   WRO  NEWS  UWAY  WELF  WEST
         ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
 1 COUN  1.000 0.625 0.625 0.625 0.625 0.250 0.625 0.750 0.625 0.500
 2 COMM  0.625 1.000 0.250 0.625 1.000 0.125 0.875 0.250 0.375 0.375
 3 EDUC  0.625 0.250 1.000 0.500 0.250 0.625 0.500 0.750 0.625 0.750
 4 INDU  0.625 0.625 0.500 1.000 0.625 0.500 0.500 0.750 0.500 0.625
 5 MAYR  0.625 1.000 0.250 0.625 1.000 0.125 0.875 0.250 0.375 0.250
 6  WRO  0.250 0.125 0.625 0.500 0.125 1.000 0.125 0.625 0.375 0.875
 7 NEWS  0.625 0.875 0.500 0.500 0.875 0.125 1.000 0.250 0.625 0.250
 8 UWAY  0.750 0.250 0.750 0.750 0.250 0.625 0.250 1.000 0.750 0.750
 9 WELF  0.625 0.375 0.625 0.500 0.375 0.375 0.625 0.750 1.000 0.375
10 WEST  0.500 0.375 0.750 0.625 0.250 0.875 0.250 0.750 0.375 1.000
```

Figure 13.6: *Proportion of matches for Knoke information receiving*

These results show similarity in a way that is quite easy to interpret. The number 0.625 in the cell 2,1 means that, in comparing actor #1 and #2, they have the same tie (present or absent) to other actors 62.5% of the time. The measure is particularly useful with multi-category nominal measures of ties; it also provides a nice scaling for binary data.

In some networks connections are very sparse. Indeed, if one were looking at ties of personal acquaintance in very large organizations, the data might have very low density. Where density is very low, the "matches", "correlation", and "distance" measures can all show relatively little variation among the actors, and may cause difficulty in discerning structural equivalence sets (of course, in very large, low density networks, there may really be very low levels of structural equivalence).

One approach to solving this problem is to calculate the number of times that both actors report a tie (or the same type of tie) to the same third actors as a percentage of the total number of ties reported. That is, we ignore cases where neither X nor Y are tied to Z, and ask, of the total ties that are present, what percentage are in common. Figure 13.7 shows the Jaccard coefficients for information receiving in the Knoke network, calculated using *Tools>Similarities*, and selecting "Jaccard".

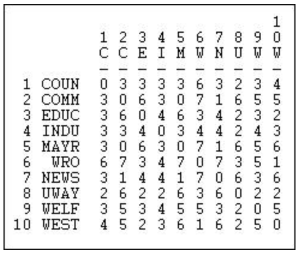


```
Percent of Positive Matches (Jaccard coefficients)

              1     2     3     4     5     6     7     8     9    10
           COUN  COMM  EDUC  INDU  MAYR   WRO  NEWS  UWAY  WELF  WEST
          ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   1       1.00
   2       0.54  1.00
   3       0.46  0.31  1.00
   4       0.60  0.54  0.42  1.00
   5       0.50  0.93  0.38  0.50  1.00
   6       0.18  0.27  0.11  0.18  0.25  1.00
   7       0.58  0.64  0.54  0.55  0.60  0.08  1.00
   8       0.67  0.46  0.50  0.67  0.43  0.20  0.38  1.00
   9       0.67  0.36  0.50  0.55  0.33  0.11  0.64  0.56  1.00
  10       0.40  0.43  0.44  0.60  0.36  0.38  0.31  0.50  0.36  1.00
```

Figure 13.7: *Jaccard coefficients for information receiving profiles in Knoke network*

Again the same basic picture emerges. The uniqueness of actor #6, though, is emphasized. Actor 6 is more unique by this measure because of the relatively small number of total ties that it has - this results in a lower level of similarity when "joint absence" of ties are ignored. Where data are sparse, and where there are very substantial differences in the degrees of points, the positive match coefficient is a good choice for binary or nominal data.

Another interesting "matching" measure is the Hamming distance, shown in Figure 13.8.



```
                           1
          1 2 3 4 5 6 7 8 9 0
          C C E I M W N U W W
          - - - - - - - - - -
 1 COUN   0 3 3 3 3 6 3 2 3 4
 2 COMM   3 0 6 3 0 7 1 6 5 5
 3 EDUC   3 6 0 4 6 3 4 2 3 2
 4 INDU   3 3 4 0 3 4 4 2 4 3
 5 MAYR   3 0 6 3 0 7 1 6 5 6
 6  WRO   6 7 3 4 7 0 7 3 5 1
 7 NEWS   3 1 4 4 1 7 0 6 3 6
 8 UWAY   2 6 2 2 6 3 6 0 2 2
 9 WELF   3 5 3 4 5 5 3 2 0 5
10 WEST   4 5 2 3 6 1 6 2 5 0
```

Figure 13.8: *Hamming distances of information receiving in Knoke network*

The Hamming distance is the number of entries in the vector for one actor that would need to be changed in order to make it identical to the vector of the other actor. These differences could be either adding or dropping a tie, so the Hamming distance treats joint absence as similarity.

With some inventiveness, you can probably think of some other reasonable ways of indexing the degree of structural similarity between actors. You might look at the program "Proximities" by SPSSx, which offers a large collection of measures of similarity. The choice of a measure should be driven by a conceptual notion of "what about" the similarity of two tie profiles is most important for the purposes of a particular analysis. Often, frankly, it makes little difference, but that is hardly sufficient grounds to ignore the question.

13.3: Visualizing Similarity and Distance

In the previous section, we've seen how the degree of similarity or distance between two actors patterns of ties with other actors can be measured and indexed. Once this is done, then what?

It is often useful to examine the similarities or distances to try to locate groupings of actors (that is, larger than a pair) who are similar. By studying the bigger patterns of which groups of actors are similar to which others, we may also gain some insight into "what about" the actor's positions is most critical in making them more similar or more distant.

Two tools that are commonly used for visualizing patterns of relationships among variables are also very helpful in exploring social network data. When we have created a similarity or distance matrix describing all the pairs of actors, we can study the similarity of differences among "cases" relations in the same way that we would study similarities among attributes.

In the next two sections we will show very brief examples of how multi-dimensional scaling and hierarchical cluster analysis can be used to identify patterns in actor-by-actor similarity/distance matrices. Both of these tools are widely used in non-network analysis; there are large and excellent literatures on the many important complexities of using these methods. Our goal here is just to provide just a very basic introduction.

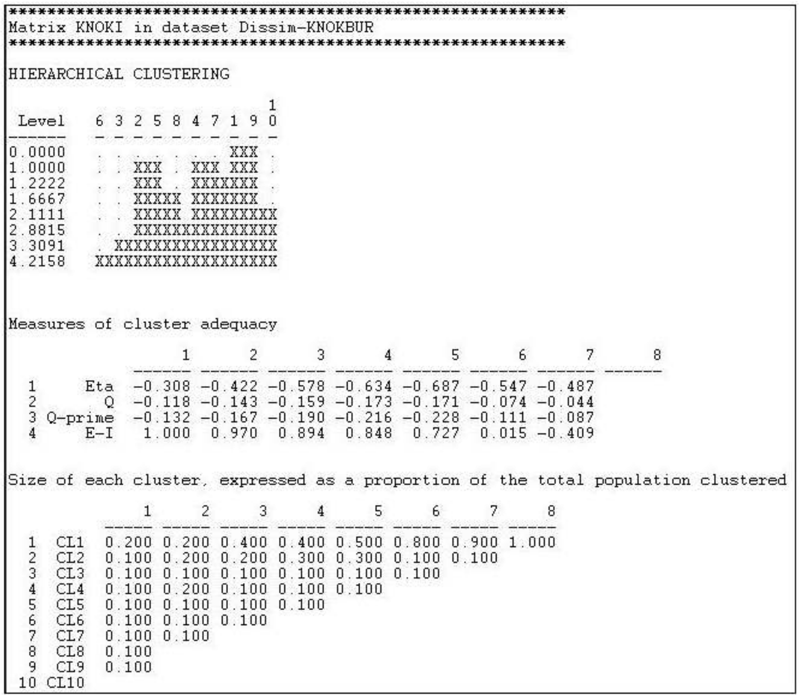
Clustering Tools

Agglomerative hierarchical clustering of nodes on the basis of the similarity of their profiles of ties to other cases provides a "joining tree" or "dendogram" that visualizes the degree of similarity among cases - and can be used to find approximate equivalence classes.

Tools>Cluster>Hierarchical proceeds by initially placing each case in its own cluster. The two most similar cases (those with the highest measured similarity index) are then combined into a class. The similarity of this new class to all others is then computed on the basis of one of three methods. On the basis of the newly computed similarity matrix, the joining/recalculation process is repeated until all cases are "agglomerated" into a single cluster. The "hierarchical" part of the method's name refers to the fact that once a case has been joined into a cluster, it is never re-classified. This results in clusters of increasing size that always enclose smaller clusters.

The "Average" method computes the similarity of the average scores in the newly formed cluster to all other clusters; the "Single-Link" method (a.k.a. "nearest neighbor") computes the similarities on the basis of the similarity of the member of the new cluster that is most similar to each other case not in the cluster. The "Complete-Link" method (a.k.a. "farthest neighbor") computes similarities between the member of the new cluster that is least similar to each other case not in the cluster. The default method is to use the cluster average; single-link methods will tend to give long, stringy joining diagrams; complete-link methods will tend to give highly separated joining diagrams.

The Hamming distance in information sending in the Knoke network was computed as shown in the section above, and the results were store as a file. This file was then input to *Tools>Cluster>Hierarchical*. We specified that the "average" method was to be used, and that the data were "dissimilarities". The results are shown as Figure 13.9.



```
**********************************************************
Matrix KNOKI in dataset Dissim-KNOKBUR
**********************************************************

HIERARCHICAL CLUSTERING

                           1
 Level   6 3 2 5 8 4 7 1 9 0
------   - - - - - - - - - -
0.0000   . . . . . . . XXX .
1.0000   . . XXX . XXX XXX .
1.2222   . . XXX . XXXXXXX .
1.6667   . . XXXXX XXXXXXX .
2.1111   . . XXXXX XXXXXXXXX
2.8815   . . XXXXXXXXXXXXXXX
3.3091   . XXXXXXXXXXXXXXXXX
4.2158   XXXXXXXXXXXXXXXXXXX


Measures of cluster adequacy

                    1      2      3      4      5      6      7      8
               ------ ------ ------ ------ ------ ------ ------ ------
  1     Eta    -0.308 -0.422 -0.578 -0.634 -0.687 -0.547 -0.487
  2       Q    -0.118 -0.143 -0.159 -0.173 -0.171 -0.074 -0.044
  3 Q-prime    -0.132 -0.167 -0.190 -0.216 -0.228 -0.111 -0.087
  4     E-I     1.000  0.970  0.894  0.848  0.727  0.015 -0.409

Size of each cluster, expressed as a proportion of the total population clustered

               1     2     3     4     5     6     7     8
           ----- ----- ----- ----- ----- ----- ----- -----
  1  CL1   0.200 0.200 0.400 0.400 0.500 0.800 0.900 1.000
  2  CL2   0.100 0.200 0.200 0.300 0.300 0.100 0.100
  3  CL3   0.100 0.100 0.100 0.100 0.100 0.100
  4  CL4   0.100 0.200 0.100 0.100 0.100
  5  CL5   0.100 0.100 0.100 0.100
  6  CL6   0.100 0.100 0.100
  7  CL7   0.100 0.100
  8  CL8   0.100
  9  CL9   0.100
 10 CL10
```

Figure 13.9: *Clustering of Hamming distances of information sending in the Knoke network*

The first graphic shows that nodes 1 and 9 were the most similar, and joined first. The graphic, by the way, can be rendered as a more polished dendogram using *Tools>Dendogram>Draw* on data saved from the cluster tool. At the next step, there are three clusters (cases 2 and 5, 4 and 7, and 1 and 9). The joining continues until (at the 8th step) all cases are agglomerated into a single cluster. This gives a clear picture of the similarity of cases, and the groupings or classes of cases. But there are really eight pictures here (one for each step of the joining). Which is the "right" solution?

Again, there is no single answer. Theory and a substantive knowledge of the processes giving rise to the data are the best guide. The second panel "Measures of cluster adequacy" can be of some assistance. There are a number of indexes here, and most will (usually) give similar answers. As we move from the right (higher steps or amounts of agglomeration) to the left (more clusters, less agglomeration) fit improves. The E-I index is often most helpful, as it measures the ratio of the numbers of ties within the clusters to ties between clusters. Generally, the goal is to achieve classes that are highly similar within, and quite distinct without. Here, one might be most tempted by the solution of the 5th step of the process (clusters of 2+5, 4+7, 1+9, and the others being single-item clusters).

To be meaningful, clusters should also contain a reasonable percentage of the cases. The last panel shows information on the relative sizes of the clusters at each stage. With only 10 cases to be clustered in our example, this is not terribly enlightening here.

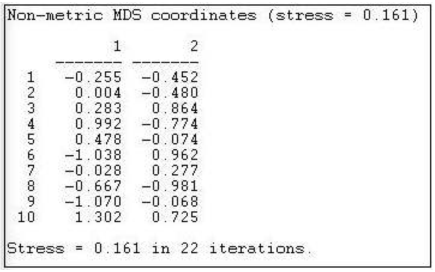
UCINET provides two additional cluster analysis tools that we won't discuss at any length here - but which you may wish to explore. *Tools>Cluster>Optimization* allows the user to select, *a priori*, a number of classes, and then uses the chosen cluster analysis method to optimally fit cases to classes. This is very similar to the structural optimization technique we will discuss below. *Tools>Cluster>Cluster Adequacy* takes a user-supplied classification (a partition, or attribute file), fits the data to it, and reports on the goodness of fit.

Multi-Dimensional Scaling Tools

Usually our goal in equivalence analysis is to identify and visualize "classes" or clusters of cases. In using cluster analysis, we are implicitly assuming that the similarity or distance among cases reflects as single underlying dimension. It is possible, however, that there are multiple "aspects" or "dimensions" underlying the observed similarities of cases. Factor or component analysis could be applied to correlations or covariances among cases. Alternatively, multi-dimensional scaling could be used (non-metric for data that are inherently nominal or ordinal; metric for valued).

MDS represents the patterns of similarity or dissimilarity in the tie profiles among the actors (when applied to adjacency or distances) as a "map" in multi-dimensional space. This map lets us see how "close" actors are, whether they "cluster" in multi-dimensional space, and how much variation there is along each dimension.

Figures 13.10 and 13.11 show the results of applying *Tools>MDS>Non-Metric MDS* to the raw adjacency matrix of the Knoke information network, and selecting a two-dimensional solution.

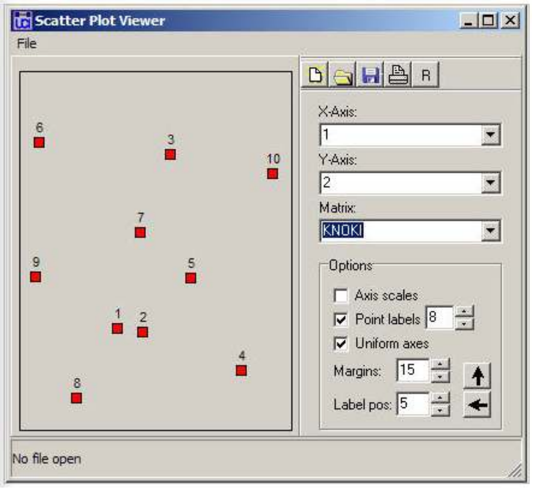


```
Non-metric MDS coordinates (stress = 0.161)

            1        2
      -------- --------
  1    -0.255   -0.452
  2     0.004   -0.480
  3     0.283    0.864
  4     0.992   -0.774
  5     0.478   -0.074
  6    -1.038    0.962
  7    -0.028    0.277
  8    -0.667   -0.981
  9    -1.070   -0.068
 10     1.302    0.725

Stress = 0.161 in 22 iterations.
```

Figure 13.10: *Non-metric MDS two-dimensional coordinates of Knoke information adjacency*

"Stress" is a measure of badness of fit. In using MDS, it is a good idea to look at a range of solutions with more dimensions, so you can assess the extent to which the distances are uni-dimensional. The coordinates show the location of each case (1 through 10) on each of the dimensions. Case one, for example, is in the lower left quadrant, having negative scores on both dimension 1 and dimension 2.

The "meaning" of the dimensions can sometimes be assessed by comparing cases that are at the extreme poles of each dimension. Are the organizations at one pole "public" and those at the other "private"? In analyzing social network data, it is not unusual for the first dimension to be simply the amount of connection or the degree of the nodes.



X-Axis: 1

Y-Axis: 2

Matrix: KNOKI

Options

☐ Axis scales

☑ Point labels 8

☑ Uniform axes

Margins: 15

Label pos: 5

No file open

Figure 13.11: *Two-dimensional map of non-metric MDS of Knoke information adjacency*

Figure 13.11 graphs the nodes according to their coordinates. In this map, we are looking for meaningful tight clusters of points to identify cases that are highly similar on both dimensions. In our example, there is very little such similarity (save, perhaps, nodes 1 and 2).

Clustering and scaling tools can be useful in many kinds of network analysis. Any measure of the relations among nodes can be visualized using these methods - adjacency, strength, correlation, and distance are most commonly examined.

These tools are also quite useful for examining equivalence. Most methods for assessing equivalence generate actor-by-actor measures of closeness or similarity in the tie profiles (using different rules, depending on what type of equivalence we are trying to measure). Cluster and MDS are often quite helpful in making sense of the results.

13.4: Describing Structural Equivalence Sets

Two actors that are structurally equivalent have the same ties to all other actors - they are perfectly substitutable or exchangeable. In "real" data, exact equivalence may be quite rare, and it many be meaningful to measure approximate equivalence. There are several approaches for examining the pattern of similarities in the tie-profiles of actors, and for forming structural equivalence classes.

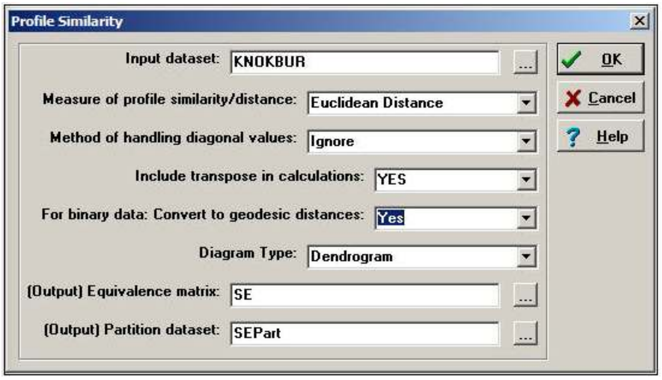
One very useful approach is to apply cluster analysis to attempt to discern how many structural equivalence sets there are, and which actors fall within each set. We will examine two more common approaches - CONCOR, and numerical optimization by tabu search.

What the similarity matrix and cluster analysis do not tell us is what similarities make the actors in each set "the same" and which differences make the actors in one set "different" from the actors in another. A very useful approach to understanding the bases of similarity and difference among sets of structurally equivalent actors is the block model, and a summary based on it called the image matrix. Both of these ideas have been explained elsewhere. We will take a look at how they can help us to understand the results of CONCOR and tabu search.

Clustering Similarities or Distances Profiles

Cluster analysis is a natural method for exploring structural equivalence. Two actors who have the similar patterns of ties to other actors will be joined into a cluster, and hierarchical methods will show a "tree" of successive joining.

Network>Roles & Positions>Structural>Profile can perform a variety of kinds of cluster analysis for assessing structural equivalence. Figure 13.12 shows a typical dialog for this algorithm.



Profile Similarity

Input dataset: KNOKBUR

Measure of profile similarity/distance: Euclidean Distance

Method of handling diagonal values: Ignore

Include transpose in calculations: YES

For binary data: Convert to geodesic distances: Yes

Diagram Type: Dendrogram

(Output) Equivalence matrix: SE

(Output) Partition dataset: SEPart

Figure 13.12: *Dialog of Network>Roles & Positions>Structural>Profile*

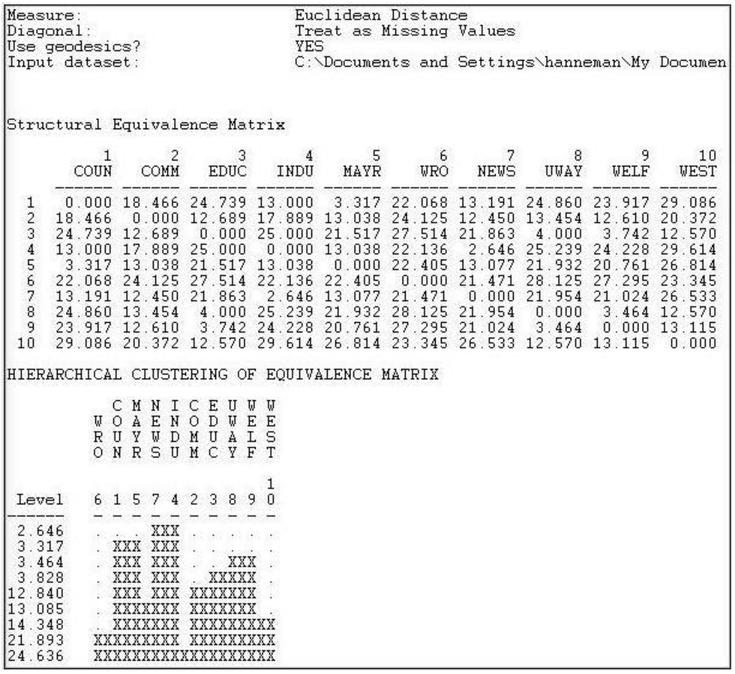
Depending on how the relations between actors have been measured, several common ways of constructing the actor-by-actor similarity or distance matrix are provided (correlations, Euclidean distances, total matches, or Jaccard coefficients). Should you desire a different measure of similarity, you can construct it elsewhere (e.g. *Tools>Similarities*), save the result, and apply cluster analysis directly (i.e. *Tools>Cluster*).

There are some other important choices. One is, what to do with the items in the similarity matrix that index the similarity of an actor to themselves (i.e. the diagonal values)? One choice ("Retain") includes the similarity of a node with itself; another choice ("Ignore") excludes diagonal elements from the calculation of similarity or difference. The default method ("Reciprocal") replaces the diagonal element for both cases with the tie that exists between the cases.

One may "Include transpose" or not. If the data being examined are symmetric (i.e. a simple graph, not a directed one), then the transpose is identical to the matrix, and shouldn't be included. For directed data, the algorithm will, by default, calculate similarities on the rows (out-ties) but not in-ties. If you want to include the full profile of both in and out ties for directed data, you need to include the transpose.

If you are working with a raw adjacency matrix, similarity can be computed on the tie profile (probably using a match or Jaccard approach). Alternatively, the adjacencies can be turned into a valued measure of dissimilarity by calculating geodesic distances (in which case correlations or Euclidean distances might be chosen as a measure of similarity).

Figure 13.13 shows the results of the analysis described in the dialog.



```
Measure:                      Euclidean Distance
Diagonal:                     Treat as Missing Values
Use geodesics?                YES
Input dataset:                C:\Documents and Settings\hanneman\My Documen


Structural Equivalence Matrix

            1      2      3      4      5      6      7      8      9     10
         COUN   COMM   EDUC   INDU   MAYR    WRO   NEWS   UWAY   WELF   WEST
       ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1     0.000 18.466 24.739 13.000  3.317 22.068 13.191 24.860 23.917 29.086
  2    18.466  0.000 12.689 17.889 13.038 24.125 12.450 13.454 12.610 20.372
  3    24.739 12.689  0.000 25.000 21.517 27.514 21.863  4.000  3.742 12.570
  4    13.000 17.889 25.000  0.000 13.038 22.136  2.646 25.239 24.228 29.614
  5     3.317 13.038 21.517 13.038  0.000 22.405 13.077 21.932 20.761 26.814
  6    22.068 24.125 27.514 22.136 22.405  0.000 21.471 28.125 27.295 23.345
  7    13.191 12.450 21.863  2.646 13.077 21.471  0.000 21.954 21.024 26.533
  8    24.860 13.454  4.000 25.239 21.932 28.125 21.954  0.000  3.464 12.570
  9    23.917 12.610  3.742 24.228 20.761 27.295 21.024  3.464  0.000 13.115
 10    29.086 20.372 12.570 29.614 26.814 23.345 26.533 12.570 13.115  0.000
HIERARCHICAL CLUSTERING OF EQUIVALENCE MATRIX

          C M N I C E U W W
        W O A E N O D W E E
        R U Y W D M U A L S
        O N R S U M C Y F T

                          1
 Level  6 1 5 7 4 2 3 8 9 0
------  - - - - - - - - - -
 2.646  . . . XXX . . . . .
 3.317  . XXX XXX . . . . .
 3.464  . XXX XXX . . XXX .
 3.828  . XXX XXX . XXXXX .
12.840  . XXX XXX XXXXXXX .
13.085  . XXXXXXX XXXXXXX .
14.348  . XXXXXXX XXXXXXXXX
21.893  XXXXXXXXX XXXXXXXXX
24.636  XXXXXXXXXXXXXXXXXXX
```

Figure 13.13: *Profile similarity of geodesic distances of rows and columns of Knoke information network*

The first panel shows the structural equivalence matrix - or the degree of similarity among pairs of actors (in this case, dissimilarity, since we chose to analyze Euclidean distances).

The second panel shows a rough character-mapped graphic of the clustering. Here we see that actors 7 and 4 are most similar, a second cluster is formed by actors 1 and 5; a third by actors 8 and 9. This algorithm also provides a more polished presentation of the result as a dendogram in a separate window, as shown in Figure 13.14.

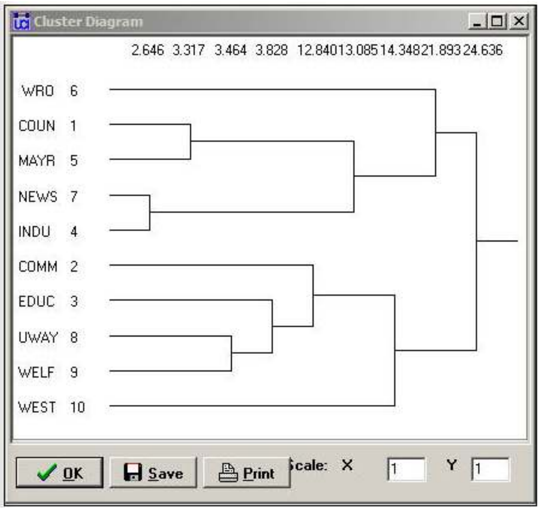


Figure 13.14: *Dendogram of structural equivalence data (see Figure 13.13)*

There are no exact structural equivalence in the example data. That is, there are no two cases that have identical ties to all other cases. The dendogram can be particularly helpful in locating groupings of cases that are sufficiently equivalent to be treated as classes. The measures of clustering adequacy in *Tools>Cluster* can provide additional guidance.

Two other approaches, CONCOR and optimization, follow a somewhat different logic than clustering. In both of these methods, partitions or approximate equivalence classes are set up first (the user selects how many), and the cases are allocated to these classes by numerical techniques designed to maximize similarity within classes.

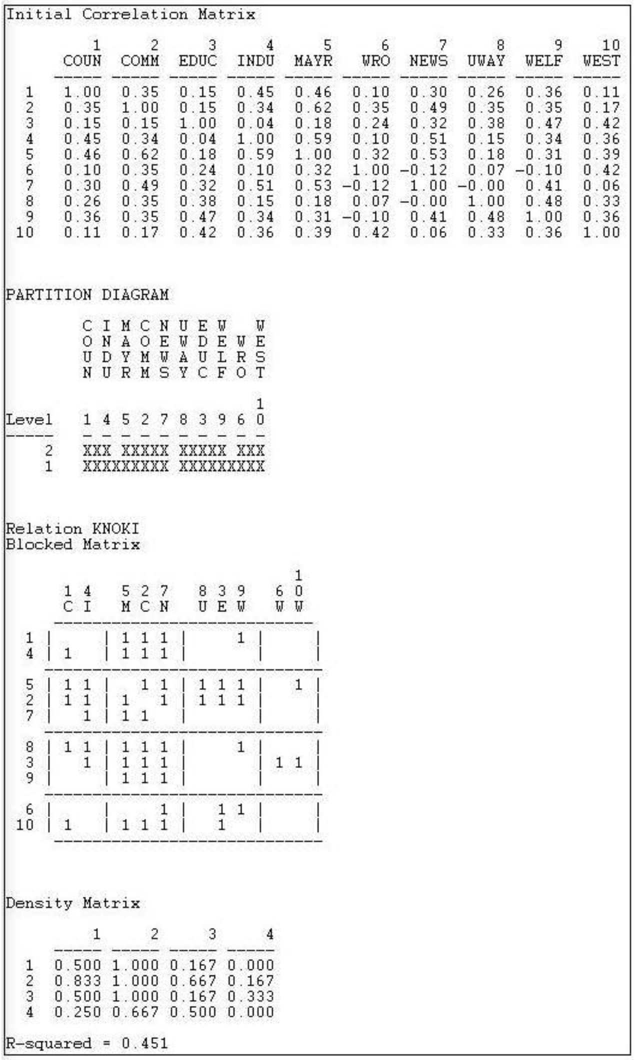
CONCOR

CONCOR is an approach that has been used for quite some time. Although the algorithm of CONCOR is now regarded as a bit peculiar, the technique usually produces meaningful results.

CONCOR begins by correlating each pair of actors (as we did above). Each row of this actor-by-actor correlation matrix is then extracted, and correlated with each other row. In a sense, the approach is asking "how similar is the vector of similarities of actor X to the vector of similarities of actor Y"? This process is repeated over and over. Eventually the elements in this "iterated correlation matrix" converge on a value of either +1 or -1 (if you want to convince yourself, give it a try!).

CONCOR then divides the data into two sets on the basis of these correlations. Then, within each set (if it has more than two actors) the process is repeated. The process continues until all actors are separated (or until we lose interest). The result is a binary branching tree that gives rise to a final partition.

For illustration, we have asked CONCOR to show us the groups that best satisfy this property when we believe that there are four groups in the Knoke information data. We used *Network>Roles & Positions>Structural>CONCOR*, and set the *depth of splits = 2* (that is, divide the data twice). All blocking algorithms require that we have a prior idea about how many groups there are. The results are shown in Figure 13.15.



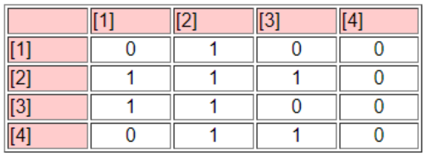
```
Initial Correlation Matrix

            1     2     3     4     5     6     7     8     9    10
         COUN  COMM  EDUC  INDU  MAYR   WRO  NEWS  UWAY  WELF  WEST
        ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   1     1.00  0.35  0.15  0.45  0.46  0.10  0.30  0.26  0.36  0.11
   2     0.35  1.00  0.15  0.34  0.62  0.35  0.49  0.35  0.35  0.17
   3     0.15  0.15  1.00  0.04  0.18  0.24  0.32  0.38  0.47  0.42
   4     0.45  0.34  0.04  1.00  0.59  0.10  0.51  0.15  0.34  0.36
   5     0.46  0.62  0.18  0.59  1.00  0.32  0.53  0.18  0.31  0.39
   6     0.10  0.35  0.24  0.10  0.32  1.00 -0.12  0.07 -0.10  0.42
   7     0.30  0.49  0.32  0.51  0.53 -0.12  1.00 -0.00  0.41  0.06
   8     0.26  0.35  0.38  0.15  0.18  0.07 -0.00  1.00  0.48  0.33
   9     0.36  0.35  0.47  0.34  0.31 -0.10  0.41  0.48  1.00  0.36
  10     0.11  0.17  0.42  0.36  0.39  0.42  0.06  0.33  0.36  1.00


PARTITION DIAGRAM

        C I M C N U E W   W
        O N A O E W D E W E
        U D Y M W A U L R S
        N U R M S Y C F O T

                          1
Level   1 4 5 2 7 8 3 9 6 0
-----   - - - - - - - - - -
    2   XXX XXXXX XXXXX XXX
    1   XXXXXXXXX XXXXXXXXX


Relation KNOKI
Blocked Matrix

                                1
       1 4   5 2 7   8 3 9   6 0
       C I   M C N   U E W   W W
      -------------------------
   1 |     | 1 1 1 |     1 |     |
   4 | 1   | 1 1 1 |       |     |
    ---------------------------
   5 | 1 1 |   1 1 | 1 1 1 |   1 |
   2 | 1 1 | 1   1 | 1 1 1 |     |
   7 |   1 | 1 1   |       |     |
    ---------------------------
   8 | 1 1 | 1 1 1 |     1 |     |
   3 |   1 | 1 1 1 |       | 1 1 |
   9 |     | 1 1 1 |       |     |
    ---------------------------
   6 |     |     1 |   1 1 |     |
  10 | 1   | 1 1 1 |   1   |     |
      -------------------------


Density Matrix

            1     2     3     4
        ----- ----- ----- -----
   1    0.500 1.000 0.167 0.000
   2    0.833 1.000 0.667 0.167
   3    0.500 1.000 0.167 0.333
   4    0.250 0.667 0.500 0.000

R-squared = 0.451
```

Figure 13.15: *CONCOR on Knoke information matrix with two splits*

The first panel shows the correlations of the cases. We included the transpose, so these correlations are based on both sending and receiving of ties. Our data, however, are binary, so the use of the correlation coefficient (and CONCOR) should be treated with caution.

The second panel shows the two splits. In the first division, the two groups {1, 4, 5, 2, 7} and {8, 3, 9, 6, 10} were formed. On the second split these were sub-divided into {1, 4}, {5, 2, 7}, {8, 3, 9}, and {6, 10}.

The third panel (the "Blocked Matrix") shows the permuted original data. The result here could be simplified further by creating a "block image" matrix of the four classes by the four classes, with "1" in high density blocks and "0" in low density blocks - as in Figure 13.16.



	[1]	[2]	[3]	[4]
[1]	0	1	0	0
[2]	1	1	1	0
[3]	1	1	0	0
[4]	0	1	1	0

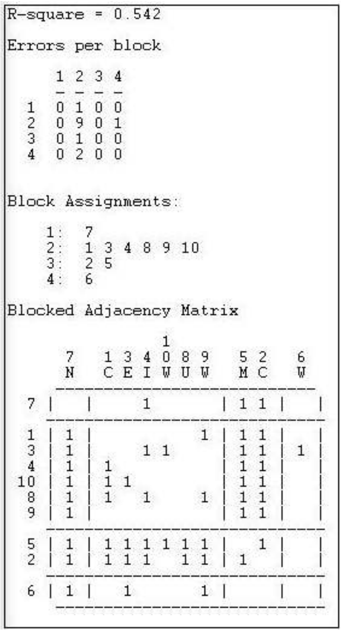
Figure 13.16: *Block image of CONCOR results*

The goodness of fit of a block model can be assessed by correlating the permuted matrix (the block model) against a "perfect" model with the same blocks (i.e. one in which all elements of one blocks are ones, and all elements of zero blocks are zeros). For the CONCOR two-split (four group) model, this r-squared is 0.451. That is, about 1/2 of the variance in the ties in the CONCOR model can be accounted for by a "perfect" structural block model. This might be regarded as OK, but is hardly a wonderful fit (there is no real criterion for what is a good fit).

The block model and its image also provide a description of what it means when we say "the actors in block one are approximately structurally equivalent". Actors in equivalence class one are likely to send ties to all other actors in block two, but no other block. Actors in equivalence class one are likely to receive ties from all actors in blocks 2 and 3. So, we have not only identified the classes, we've also described the form of the relations that makes the cases equivalent.

Optimization by Tabu Search

This method of blocking has been developed more recently, and relies on extensive use of the computer. Tabu search uses a more modern (and computer intensive) algorithm than CONCOR, but is trying to implement the same idea of grouping together actors who are most similar into a block. Tabu search does this by searching for sets of actors who, if placed into a block, produce the smallest sum of within-block variances in the tie profiles. That is, if actors in a block have similar ties, their variance around the block mean profile will be small. So, the partitioning that minimizes the sum of within block variances is minimizing the overall variance in tie profiles. In principle, this method ought to produce results similar (but not necessarily identical) to CONCOR. In practice, this is not always so. Here (Figure 13.17) are the results of *Network>Roles & Positions>Structural>Optimization>Binary* applied to the Knoke information network, and requesting four classes. A variation of the technique for valued data is available as *Network>Roles & Positions>Structural>Optimization>Valued*.



```
R-square = 0.542

Errors per block

     1 2 3 4
     - - - -
  1  0 1 0 0
  2  0 9 0 1
  3  0 1 0 0
  4  0 2 0 0


Block Assignments:

    1:  7
    2:  1 3 4 8 9 10
    3:  2 5
    4:  6

Blocked Adjacency Matrix

                1
      7   1 3 4 0 8 9   5 2   6
      N   C E I W U W   M C   W
     ---------------------------
  7 |   |     1       | 1 1 |   |
    -----------------------------
  1 | 1 |           1 | 1 1 |   |
  3 | 1 |     1 1     | 1 1 | 1 |
  4 | 1 | 1           | 1 1 |   |
 10 | 1 | 1 1         | 1 1 |   |
  8 | 1 | 1   1     1 | 1 1 |   |
  9 | 1 |             | 1 1 |   |
    -----------------------------
  5 | 1 | 1 1 1 1 1 1 |   1 |   |
  2 | 1 | 1 1 1   1 1 | 1   |   |
    -----------------------------
  6 | 1 |   1       1 |     |   |
     ---------------------------
```

Figure 13.17: *Optimized four-block solution for structural equivalence of Knoke information network*

The overall correlation between the actual scores in the blocked matrix, and a "perfect" matrix composed of only ones and zeros is reasonably good (0.544).

The suggested partition into structural equivalence classes is {7}, {1, 3, 4, 10, 8, 9}, {5, 2}, and {6}.

We can now also describe the positions of each of the classes. The first class (actor 7) has dense sending ties to the third (actors 5 and 2); and receives information from all three other classes. The second, and largest, class sends information to the first and the third class, and receives information from the third class. The last class (actor 6), sends to the first class, but receives from none.

This last analysis illustrates most fully the primary goals of an analysis of structural equivalence:

1) How many equivalence classes, or approximate equivalence classes, are there?

2) How good is the fit of this simplification into equivalence classes in summarizing the information about all the nodes?

3) What is the position of each class, as defined by its relations to the other classes?

13.S: Measures of Similarity and Structural Equivalence (Summary)

In this chapter we have discussed the idea of "structural equivalence" of actors, and seen some of the methodologies that are most commonly used to measure structural equivalence, find patterns in empirical data, and describe the sets of "substitutable" actors.

Structural equivalence of two actors is the degree to which the two actors have the same profile of relations across alters (all other actors in the network). Exact structural equivalence is rare in most social structures (one interpretation of exact structural equivalence is that it represents systematic redundancy of actors, which may be functional in some way to the network).

While it is sometimes possible to see patterns of structure equivalence "by eye" from an adjacency matrix or diagram, we almost always use numerical methods. Numerical methods allow us to deal with multiplex data, large numbers of actors, and valued data (as well as the binary type that we have examined here).

The first step in examining structural equivalence is to produce a "similarity" or a "distance" matrix for all pairs of actors. This matrix summarizes the overall similarity (or dissimilarity) of each pair of actors in terms of their ties to alters. While there are many ways of calculating such index numbers, the most common are the Pearson Correlation, the Euclidean Distance, the proportion of matches (for binary data), and the proportion of positive matches (Jaccard coefficient, also for binary data).

A number of methods may be used to identify patterns in the similarity or distance matrix, and to describe those patterns. Cluster analysis groups together the two most similar actors, recalculates similarities, and iterates until all actors are combined. What is produced is a "joining sequence" or map of which actors fall into a hierarchy of increasingly inclusive (and hence less exactly equivalent) groups. Multi-dimensional scaling and factor analysis can be used to identify what aspects of the tie profiles are most critical to making actors similar or different, and can also be used to identify groups. Groupings of structurally equivalent actors can also be identified by the divisive method of iterating the correlation matrix of actors (CONCOR), and by the direct method of permutation and search for perfect zero and one blocks in the adjacency matrix (Optimization by Tabu search).

Once the number of groupings that are useful has been determined, the data can be permuted and blocked, and images calculated. These techniques enable us to get a rather clear picture of how the actors in one set are "approximately equivalent" and why different sets of actors are different. That is, they enable us to describe the meaning of the groups, and the place of group members in the overall network in a general way.

Structural equivalence analysis often produces interesting and revealing findings about the patterns of ties and connections among the individual actors in a network. The structural equivalence concept aims to operationalize the notion that actors may have identical positions in a network - and hence be directly "substitutable" for one another. An alternative interpretation is that actors who are structurally equivalent face nearly the same matrix of constraints and opportunities in their social relationships.

Sociological analysis is not really about individual people. And, structural analysis is primarily concerned with the more general and abstract idea of the roles or positions that define the structure of the group - rather than the locations of specific actors with regard to specific others. For such analysis, we turn to a related set of tools for studying replicate sub-structures ("automorphic equivalence") and social roles ("regular equivalence").

CHAPTER OVERVIEW

14: Automorphic Equivalence

Automorphic equivalence is not as demanding a definition of similarity as structural equivalence, but is more demanding than regular equivalence. There is a hierarchy of the three equivalence concepts: any set of structural equivalences are also automorphic and regular equivalences. Any set of automorphic equivalences are also regular equivalences. Not all regular equivalences are necessarily automorphic or structural; and not all automorphic equivalences are necessarily structural.

14.1: Defining automorphic equivalence

Automorphic equivalence is not as demanding a definition of similarity as structural equivalence, but is more demanding than regular equivalence. There is a hierarchy of the three equivalence concepts: any set of structural equivalences are also automorphic and regular equivalences. Any set of automorphic equivalences are also regular equivalences. Not all regular equivalences are necessarily automorphic or structural; and not all automorphic equivalences are necessarily structural.

Formally "Two vertices u and v of a labeled graph G are automorphically equivalent if all the vertices can be re-labeled to form an isomorphic graph with the labels of u and v interchanged. Two automorphically equivalent vertices share exactly the same label-independent properties." (Borgatti, Everett, and Freeman, 1996: 119).

More intuitively, actors are automorphically equivalent if we can permute the graph in such a way that exchanging the two actors has no effect on the distances among all actors in the graph. If we want to assess whether two actors are automorphically equivalent, we first imagine exchanging their positions in the network. Then, we look and see if, by changing some other actors as well, we can create a graph in which all of the actors are the same distance that they were from one another in the original graph.

In the case of structural equivalence, two actors are equivalent if we can exchange them one-for-one, and not affect any properties of the graph. Automorphically equivalent actors are actors that can be exchanged with no effect on the graph -- given that other actors are also moved. If the concept is still a bit difficult to grasp at this point, don't worry. Read on, and then come back after you've looked at a few examples.

14.2: Uses of the Concept

Structural equivalence focuses our attention on pair-wise comparisons of actors. By trying to find actors who can be swapped for each other, we are really paying attention to the positions of the actors in a particular network. We are trying to find actors who are clones or substitutes.

Automorphic equivalence begins to change the focus of our attention, moving us away from concern with individual's network positions, and toward a more abstracted view of the network. Automorphic equivalence asks if the whole network can be re-arranged, putting different actors at different nodes, but leaving the relational structure or skeleton of the network intact.

Suppose that we had 10 workers in the University Avenue McDonald's restaurant, who report to one manager. The manager, in turn, reports to a franchise owner. The franchise owner also controls the Park Street McDonald's restaurant. It too has a manager and 10 workers. Now, if the owner decided to transfer the manager from University Avenue to the Park Street restaurant (and vice versa), the network has been disrupted. But if the owner transfers both the managers and the workers to the other restaurant, all of the network relations remain intact. Transferring both the workers and the managers is a permutation of the graph that leaves all of the distances among the pairs of actors exactly as it was before the transfer. In a sense, the "staff" of one restaurant is equivalent to the staff of the other, though the individual persons are not substitutable.

The hypothetical example of the restaurants suggests the main utility of the automorphic equivalence concept. Rather than asking what individuals might be exchanged without modifying the social relations described by a graph (structural equivalence), the somewhat more relaxed concept of automorphic equivalence focuses our attention on sets of actors who are substitutable as sub-graphs, in relation to other sub-graphs. In many social structures, there may well be sub-structures that are equivalent to one another (or approximately so). The number, type, and relations among such sub-structures might be quite interesting. Many structures that look very large and complex may actually be composed (at least partially) of multiple identical sub-structures; these sub-structures may be "substitutable" for one another. Indeed, a McDonalds is a McDonalds is a McDonalds...

14.3: Finding Equivalence Sets

With binary data, numerical algorithms are used to search for classes of actors that satisfy the mathematical definitions of automorphic equivalence. Basically, the nodes of a graph are exchanged, and the distances among all pairs of actors in the new graph are compared to the original graph. When the new graph and the old graph have the same distances among nodes, the graphs are isomorphic, and the "swapping" that was done identifies the isomorphic sub-graphs.

One approach to binary data, "all permutations," (*Network>Roles & Positions>Automorphic>All Permutations*) literally compares every possible swapping of nodes to find isomorphic graphs. With even a small graph, there are a very large number of such alternatives, and the computation is extensive. An alternative approach with the same intent ("optimization by tabu search") (*Network>Roles & Positions>Exact>Optimization*) can much more quickly sort nodes into a user-defined number of partitions in such a way as to maximize automorphic equivalence. There is no guarantee, however, that the number of partitions (equivalence classes) chosen is "correct," or that the automorphisms identified are "exact." For larger data sets, and where we are willing to entertain the idea that two sub-structures can be "almost" equivalent, optimization is a very useful method.

When we have measures of the strength, cost, or probability of relations among nodes (i.e. valued data), exact automorphic equivalence is far less likely. It is possible, however, to identify classes of approximately equivalent actors on the basis of their profile of distance to all other actors. The "equivalence of distances" method (*Network>Roles & Positions>Automorphic>MaxSim*) produces measures of the degree of automorphic equivalence for each pair of nodes, which can be examined by clustering and scaling methods to identify approximate classes. This method can also be applied to binary data by first turning binary adjacency into some measure of graph distance (usually, geodesic distance).

Let's look at these in a little more detail, with some examples.

All Permutations (i.e. Brute Force)

The automorphisms in a graph can be identified by the brute force method of examining every possible permutation of the graph. With a small graph, and a fast computer, this is a useful thing to do. Basically, every possible permutation of the graph is examined to see if it has the same tie structure as the original graph. For graphs of more than a few actors, the number of permutations that need to be compared becomes extremely large.

Let's use *Networks>Roles & Positions>Automorphic>All Permutations* to search the Wasserman-Faust network shown in Figure 14.1.

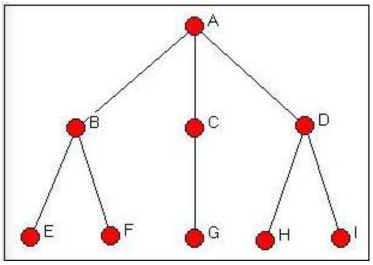
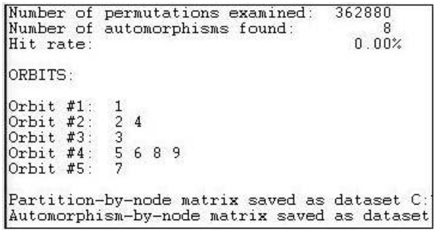


Figure 14.1: Wasserman-Faust network

The results of *Networks>Roles & Positions>Automorphic>All Permutations* for this graph are shown in Figure 14.2.



```
Number of permutations examined:  362880
Number of automorphisms found:         8
Hit rate:                          0.00%

ORBITS:

Orbit #1:  1
Orbit #2:  2 4
Orbit #3:  3
Orbit #4:  5 6 8 9
Orbit #5:  7

Partition-by-node matrix saved as dataset C:
Automorphism-by-node matrix saved as dataset
```

Figure 14.2: *Automorphic equivalences by all permutations for the Wasserman-Faust network*

The algorithm examined over three hundred sixty two thousand possible permutations of the graph. The isomorphism classes that it located are called "orbits." And, the results correspond to our logical analysis (chapter 12). There are five "types" of actors who are embedded at equal distances from other sets of actors: actor A (orbit 1), actor C (orbit 3), and actor G (orbit 7) are unique. Actors B and D form a class of actors who can be exchanged if members of other classes are also exchanged; actors E, F, H, and I (5, 6, 8, and 9) also form a class of exchangeable actors.

Note that automorphism classes identify groups of actors who have the same pattern of distance from other actors, rather than the "sub-structures" themselves (in this case, the two branches of the tree.

Optimization by Tabu Search

For larger graphs, direct search for all equivalencies is impractical both because it is computationally intensive, and because exactly equivalent actors are likely to be rare.

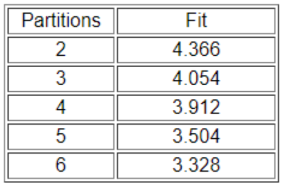
Network>Roles & Positions>Exact>Optimization provides a numerical tool for finding the best approximations to a user-selected number of automorphism classes. In using this method, it is important to explore a range of possible numbers of partitions (unless one has a prior theory about this), to determine how many partitions are useful. Having selected a number of partitions, it is useful to re-run the algorithm a number of times to insure that a global, rather than local minimum has been found.

The method begins by randomly allocating nodes to partitions. A measure of badness of fit is constructed by calculating the sums of squares for each row and each column within each block, and calculating the variance of these sums of squares. These variances are then summed across the blocks to construct a measure of badness of fit. Search continues to find an allocation of actors to partitions that minimizes this badness of fit statistic.

What is being minimized is a function of the dissimilarity of the variance of scores within partitions. That is, the algorithm seeks to group together actors who have similar amounts of variability in their row and column scores within blocks. Actors who have similar variability probably have similar profiles of ties sent and received within, and across blocks -- though they do not necessarily have the same ties to the same other actors, they are likely to have ties of the same distance to actors in other classes.

Let's examine the Knoke bureaucracies information exchange network again, this time looking for automorphisms. In the Knoke information data there are no exact automorphisms. This is not really surprising, given the complexity of the pattern (and particularly if we distinguish in-ties from out-ties) of connections.

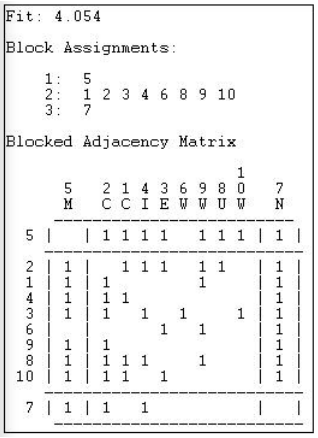
We ran the routine a number of times, requesting partitions into different numbers of classes. Figure 14.3 summarizes the "badness of fit" of the models.



Partitions	Fit
2	4.366
3	4.054
4	3.912
5	3.504
6	3.328

Figure 14.3: *Fit of automorphic equivalence models to Knoke information network*

There is no "right" answer about how many classes there are. There are two trivial answers: those that group all the cases together into one partition and those that separate each case into its own partition. In between, one might want to follow the logic of the "scree" plot from factor analysis to select a meaningful number of partitions. Look first at the results for three partitions (Figure 14.4).



```
Fit: 4.054

Block Assignments:

    1:  5
    2:  1 2 3 4 6 8 9 10
    3:  7

Blocked Adjacency Matrix

                        1
      5   2 1 4 3 6 9 8 0   7
      M   C C I E W W U W   N
     -------------------------
  5 |   | 1 1 1 1   1 1 1 | 1 |
    ---------------------------
  2 | 1 |   1 1 1   1 1   | 1 |
  1 | 1 | 1         1     | 1 |
  4 | 1 | 1 1             | 1 |
  3 | 1 | 1   1   1     1 | 1 |
  6 |   |       1   1     | 1 |
  9 | 1 | 1               | 1 |
  8 | 1 | 1 1 1     1     | 1 |
 10 | 1 | 1 1   1         | 1 |
    ---------------------------
  7 | 1 | 1   1           |   |
     -------------------------
```

Figure 14.4: *3-class automorphic equivalence solution for the Knoke information network*

At this level of approximate equivalence, there are three classes - two individuals and one large group. The newspaper (actor 7) has low rates of sending (row) and high rates of receiving (column); the mayor (actor 5) has high rates of sending and high rates of receiving. With only three classes, the remainder of the actors are grouped into an approximate class with roughly equal (and higher) variability of both sending and receiving.

Because automorphic equivalence actually operates on the profile of distances of actors, it tends to identify groupings of actors who have similar patterns of in and out degree. This goes beyond structural equivalence (which emphasizes ties to exactly the same other actors) to a more general and fuzzier idea that two actors are equivalent if they are similarly embedded. The emphasis shifts from individual position, to the role of the position in the structure of the whole graph.

Equivalence of Distances (MaxSim)

When we have information on the strength, cost, or probability of relations (i.e. valued data), exact automorphic equivalence could be expected to be extremely rare. But, since automorphic equivalence emphasizes the similarity in the profile of distances of actors from others, the idea of approximate equivalence can be applied to valued data. *Network>Roles & Positions>Automorphic>MaxSim* generates a matrix of "similarity" between shape of the distributions of ties of actors that can be grouped by clustering and scaling into approximate classes. The approach can also be applied to binary data, if we first convert the adjacency matrix into a matrix of geodesic nearnesses (which can be treated as a valued measure of the strength of ties).

The algorithm begins with a (reciprocal of) distance or strength of tie matrix. The distances of each actor re-organized into a sorted list from low to high, and the Euclidean distance is used to calculate the dissimilarity between the distance profiles of each pair of actors. The algorithm scores actors who have similar distance profiles as more automorphically equivalent. Again, the focus is on whether actor u has a similar set of distances, regardless of which distances, to actor v. Again, dimensional scaling or clustering of the distances can be used to identify sets of approximately automorphically equivalent actors.

Let's apply this idea to two examples, one simple and abstract, the other more realistic. First, let's look at the "line" network (Figure 14.5).

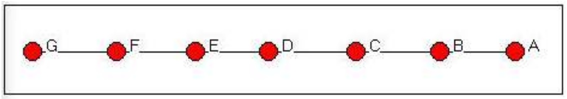
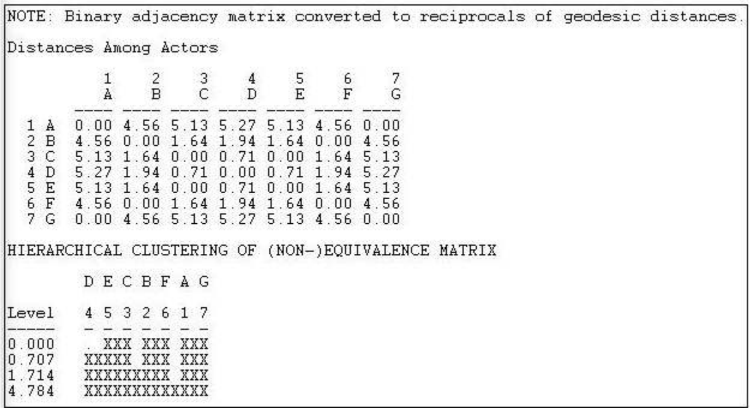


Figure 14.5: *Line network*

Figure 14.6 shows the results of analyzing this network with *Network>Roles & Positions>Automorphic>MaxSim.*



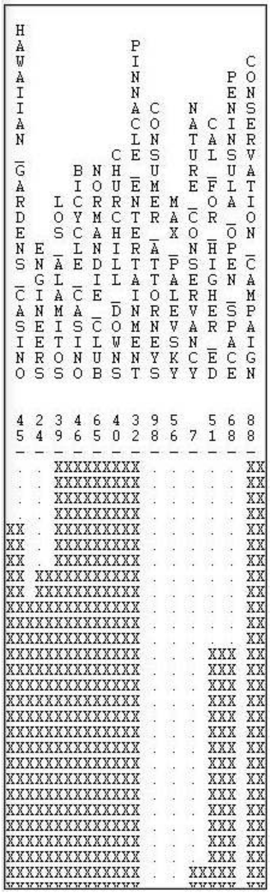
```
NOTE: Binary adjacency matrix converted to reciprocals of geodesic distances.

Distances Among Actors

             1     2     3     4     5     6     7
             A     B     C     D     E     F     G
          ----- ----- ----- ----- ----- ----- -----
   1 A    0.00  4.56  5.13  5.27  5.13  4.56  0.00
   2 B    4.56  0.00  1.64  1.94  1.64  0.00  4.56
   3 C    5.13  1.64  0.00  0.71  0.00  1.64  5.13
   4 D    5.27  1.94  0.71  0.00  0.71  1.94  5.27
   5 E    5.13  1.64  0.00  0.71  0.00  1.64  5.13
   6 F    4.56  0.00  1.64  1.94  1.64  0.00  4.56
   7 G    0.00  4.56  5.13  5.27  5.13  4.56  0.00

HIERARCHICAL CLUSTERING OF (NON-)EQUIVALENCE MATRIX

          D E C B F A G

Level     4 5 3 2 6 1 7
-----     - - - - - - -
0.000     . XXX XXX XXX
0.707     XXXXX XXX XXX
1.714     XXXXXXXXX XXX
4.784     XXXXXXXXXXXXX
```

Figure 14.6: *Automorphic equivalence of geodesic distances in the line network*

The first step is to convert the adjacency matrix into a geodesic distance matrix. Then the reciprocal of the distance is taken, and a vector of the rows entries concatenated with the column entries for each actor is produced. The Euclidean distances between these lists are then created as a measure of the non-automorphic-equivalence, and hierarchical clustering is applied.

We see that actors 3 and 5 (C and E) form a class; actors 2 and 6 (B and F) form a class; actors 1 and 7 (A and G) form a class, and actor 4 (D) is a class. Mathematically, this is a sensible result; exchanges of labels of actors within these sets can occur and still produce an isomorphic distance matrix. The result also makes substantive sense -- and is quite like that for the Wasserman-Faust network.

This approximation method can also be applied where the data are valued. If we look at our data on donors to California political campaigns, we have measures of the strength of ties among the actors that are the number of positions in campaigns they have in common when either contributed. Figure 14.7 shows part of the output of *Network>Roles & Positions>Automorphic>MaxSim.*



```
H
A           P
W           I           C
A           N         P O
I           N         E N
I           A C   N   N S
A           C O   A C I E
N           L N   T A N R
_         C E S   U L S V
G     B N H _ U   R _ U A
A     I O U E M   E F L T
R   L C R R N E M _ O A I
D   O Y M C T R A C R _ O
E   S C A H E _ X O _ O N
N E _ L N I R A _ N H P _
S N A E D L T T P S I E C
_ G L _ I L A T A E G N A
C I A C E _ I O L R H _ M
A N M A _ D N R E V E S P
S E I S C O M N V A R P A
I E T I L W E E S N _ A I
N R O N U N N Y K C E C G
O S S O B S T S Y Y D E N

4 2 3 4 6 4 3 9 5   5 6 8
5 4 9 6 5 0 2 8 6 7 1 8 8
- - - - - - - - - - - - -
. . XXXXXXXXX . . . . . XX
. . XXXXXXXXX . . . . . XX
. . XXXXXXXXX . . . . . XX
. . XXXXXXXXX . . . . . XX
XX . XXXXXXXXX . . . . . XX
XX . XXXXXXXXX . . . . . XX
XX . XXXXXXXXX . . . . . XX
XX XXXXXXXXXXX . . . . . XX
XX XXXXXXXXXXX . . . . . XX
XXXXXXXXXXXXX . . . . . XX
XXXXXXXXXXXXX . . . . . XX
XXXXXXXXXXXXX . . . . . XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . . XXX XX
XXXXXXXXXXXXX . . XXXXX XX
```

Figure 14.7: *Approximate automorphic equivalence of California political donors (truncated)*

This small part of a large piece of output (there are 100 donors in the network) shows that a number of non-Indian casinos and race-tracks cluster together, and separately from some other donors who are primarily concerned with education and ecological issues.

The identification of approximate equivalence classes in valued data can be helpful in locating groups of actors who have a similar location in the structure of the graph as a whole. By emphasizing distance profiles, however, it is possible to finds classes of actors that include nodes that are quite distant from one another, but at a similar distance to all the other actors. That is, actors that have similar positions in the network as a whole.

14.S: Automorphic Equivalence (Summary)

The kind of equivalence expressed by the notion of automorphism falls between structural and regular equivalence, in a sense. Structural equivalence means that individual actors can be substituted one for another. Automorphic equivalence means that sub-structures of graphs can can be substituted for one another. As we will see next, regular equivalence goes further still, and seeks to deal with classes or types of actors--where each member of any class has similar relations with some member of each other.

The notion of structural equivalence corresponds well to analyses focusing on how individuals are embedded in networks -- or network positional analysis. The notion of regular equivalence focuses our attention on classes of actors, or "roles" rather than individuals or groups. Automorphic equivalence analysis falls between these two more conventional foci, and has not received as much attention in empirical research. Still, the search for multiple substitutable sub-structures in graphs (particularly in large and complicated ones) may reveal that the complexity of very large structures is more apparent than real; sometimes very large structures are decomposable (or partially so) into multiple similar smaller ones.

CHAPTER OVERVIEW

15: Regular Equivalence

Regular equivalence is the least restrictive of the three most commonly used definitions of equivalence. It is, however, probably the most important for the sociologist. This is because the concept of regular equivalence, and the methods used to identify and describe regular equivalence sets correspond quite closely to the sociological concept of a "role". The notion of social roles is a centerpiece of most sociological theorizing.

15.1: Defining Regular Equivalence

Regular equivalence is the least restrictive of the three most commonly used definitions of equivalence. It is, however, probably the most important for the sociologist. This is because the concept of regular equivalence, and the methods used to identify and describe regular equivalence sets correspond quite closely to the sociological concept of a "role". The notion of social roles is a centerpiece of most sociological theorizing.

Formally, "Two actors are regularly equivalent if they are equally related to equivalent others." (Borgatti, Everett, and Freeman, 1996: 128). That is, regular equivalence sets are composed of actors who have similar relations to members of other regular equivalence sets. The concept does not refer to specific other actors, or to presence in similar sub-graphs; actors are regularly equivalent if they have similar ties to any members of other sets.

The concept is actually more easy to grasp intuitively than formally. Susan is the daughter of Inga. Deborah is the daughter of Sally. Susan and Deborah form a regular equivalence set because each has a tie to a member of the other set. Inga and Sally form a set because each has a tie to a member of the other set. In regular equivalence, we don't care which daughter goes with which mother; what is identified by regular equivalence is the presence of two sets (which we might label "mothers" and "daughters"), each defined by its relation to the other set. Mothers are mothers because they have daughters; daughters are daughters because they have mothers.

15.2: Uses of the Concept

Most approaches to social positions define them relationally. For Marx, capitalists can only exist if there are workers, and *vice versa*. The two "roles" are defined by the relation between them (i.e. capitalists expropriate surplus value from the labor power of workers). Husbands and wives; men and women; minorities and majorities; lower caste and higher caste; and most other roles are defined relationally.

The regular equivalence approach is important because it provides a method for identifying "roles" from the patterns of ties present in a network. Rather than relying on attributes of actors to define social roles and to understand how social roles give rise to patterns of interaction, regular equivalence analysis seeks to identify social roles by identifying regularities in the patterns of network ties - whether or not the occupants of the roles have names for their positions.

Regular equivalence analysis of a network then can be used to locate and define the nature of roles by their patterns of ties. The relationship between the roles that are apparent from regular equivalence analysis and the actor's perceptions or naming of their roles can be problematic. What actors label others with role names, and the expectations that they have toward them as a result (i.e. the expectations or norms that go with roles) may pattern - but not wholly determine actual patterns of interaction. Actual patterns of interaction, in turn, are the regularities out of which roles and norms emerge.

These ideas: interaction giving rise to culture and norms, and norms and roles constraining interaction, are at the core of the micro-sociological perspective. The identification and definition of "roles" by the regular equivalence analysis of network data is possibly the most important intellectual development of social network analysis.

15.3: Finding Equivalence Sets

The formal definition says that two actors are regularly equivalent if they have similar patterns of ties to equivalent others. Consider two men. Each has children (though they have different numbers of children, and, obviously have different children). Each has a wife (though again, usually different persons fill this role with respect to each man). Each wife, in turn, also has children and a husband (that is, they have ties with one or more members of each of those sets). Each child has ties to one or more members of the set of "husbands" and "wives".

In identifying which actors are "husbands" we do not care about ties between members of this set (actually, we would expect this block to be a zero block, but we really don't care). What is important is that each "husband" have at least one tie to a person in the "wife" category and at least one person in the "child" category. That is, husbands are equivalent to each other because each has similar ties to some member of the sets of wives and children.

But there would seem to be a problem with this fairly simple definition. If the definition of each position depends on its relations with other positions, where do we start?

There are a number of algorithms that are helpful in identifying regular equivalence sets. UCINET provides some methods that are particularly helpful for locating approximately regularly equivalent actors in valued, multi-relational and directed graphs. Some simpler methods for binary data can be illustrated directly.

Consider, again, the Wasserman-Faust example network. Imagine, however, that this is a picture of order-giving in a simple hierarchy. That is, all ties are directed from the top of the diagram in Figure 15.1 downward. We will find a regular equivalence characterization of this graph.

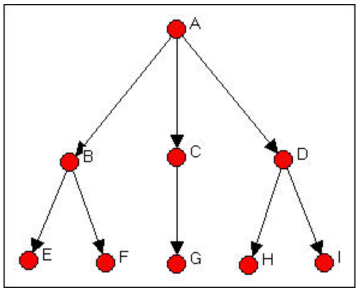


Figure 15.1: *Directed tie version of the Wasserman-Faust network*

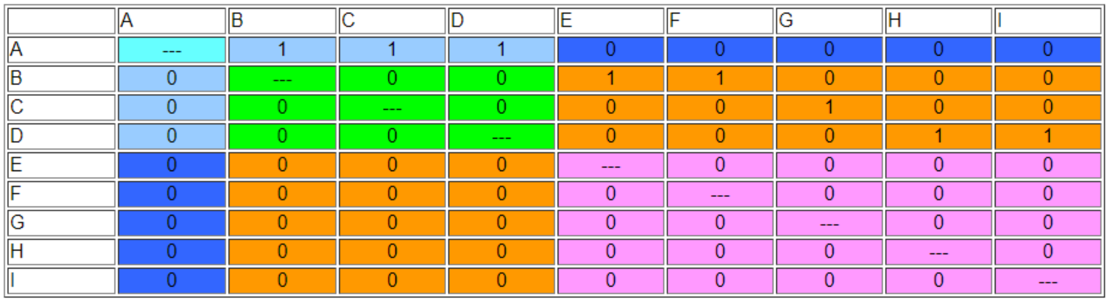
For a first step, characterize each node as either a "source" (an actor that sends ties, but does not receive them), a "repeater" (an actor that both repeats and sends), or a "sink" (an actor that receives ties, but does not send them). The source is A; repeaters are B, C, and D; and sinks are E, F, G, H, and I. There is a fourth logical possibility. An "isolate" is a node that neither sends nor receives ties. Isolates form a regular equivalence set in any network, and should be excluded from the regular equivalence analysis of the connected sub-graph.

Since there is only one actor in the set of senders, we cannot identify any further complexity in this "role".

Consider the three "repeaters" B, C, and D. In the neighborhood of (that is, adjacent to) actor B are both "sources" and "sinks". The same is true for "repeaters" C and D, even though the three actors may have different numbers of sources and sinks, and these may be different (or the same) specific sources and sinks. We cannot define the "role" of the set {B, C, D} any further, because we have exhausted their neighborhoods. That is, the sources to whom our repeaters are connected cannot be further differentiated into multiple types (because there is only one source); the sinks to whom our repeaters send cannot be further differentiated, because they have no further connections themselves.

Now consider our "sinks" (i.e. actors E, F, G, H, and I). Each is connected to a source (although the sources may be different). We have already determined, in the current case, that all of these sources (actors B, C, and D) are regularly equivalent. So, E through I are equivalently connected to equivalent others. We are done with our partitioning.

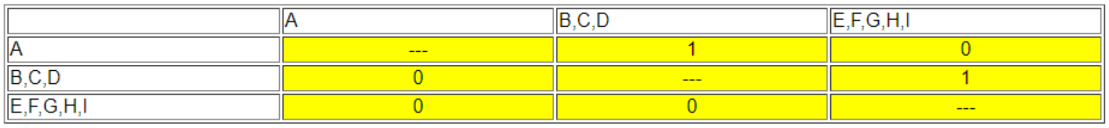
The result of {A} {B, C, D} {E, F, G, H, I} satisfies the condition that each actor in each partition have the same pattern of connections to actors in other partitions. The permuted adjacency matrix is shown in Figure 15.2.



	A	B	C	D	E	F	G	H	I
A	---	1	1	1	0	0	0	0	0
B	0	---	0	0	1	1	0	0	0
C	0	0	---	0	0	0	1	0	0
D	0	0	0	---	0	0	0	1	1
E	0	0	0	0	---	0	0	0	0
F	0	0	0	0	0	---	0	0	0
G	0	0	0	0	0	0	---	0	0
H	0	0	0	0	0	0	0	---	0
I	0	0	0	0	0	0	0	0	---

Figure 15.2: *Permuted Wasserman-Faust network to show regular equivalence classes*

It is useful to block this matrix and show its image. Here, however, we will use some special rules for determining zero and 1 blocks. If a block is all zeros, it will be a zero block. If each actor in a partition has a tie to any actor in another, then we will define the joint block as a 1-block. Bear with me a moment. The image using this rule is shown in Figure 15.3.



	A	B,C,D	E,F,G,H,I
A	---	1	0
B,C,D	0	---	1
E,F,G,H,I	0	0	---

Figure 15.3: *Block image of regular equivalence classes in directed Wasserman-Faust network*

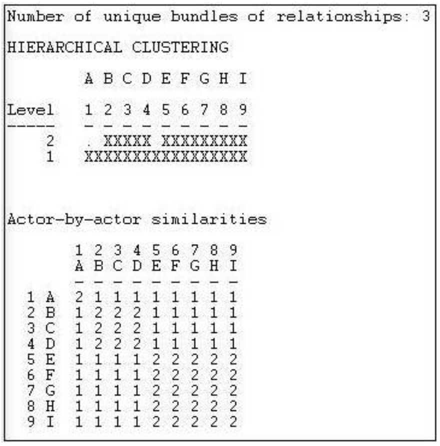
{A} sends to one or more of {BCD} but to none of {EFGHI}. {BCD} does not send to {A}, but each of {BCD} sends to at least one of {EFGHI}. None of {EFGHI} send to any of {A}, or of {BCD}. The image, in fact, displays the characteristic pattern of a strict hierarchy: ones on the first off-diagonal vector and zeros elsewhere. The rule of defining a 1-block when each actor in one partition has a relationship with any actor in the other partition is a way of operationalizing the notion that the actors in the first set are equivalent if they are connected to equivalent actors (i.e. actors in the other partition), without requiring (or prohibiting) that they be tied to the same other actors, or the same number of actors in another partition.

For directed binary graphs, the neighborhood search method we applied here usually works quite well. For binary graphs that are not directed, usually the geodesic distance among actors is computed and used instead of raw adjacency. For graphs with valued relations (strength, cost, probability), a method for identifying approximate regular equivalence was developed by White and Reitz. These several alternatives are illustrated below.

Categorical REGE for Directed Binary Data (Wasserman-Faust Directed Network)

The neighborhood search method illustrated above (with the directed Wasserman-Faust network) is the algorithm performed by *Network>Roles & Positions>Maximal Regular>CATREGE*. This approach is ideal for networks where relations are measured at the nominal level, and are directed. Our example will be of a binary graph; the algorithm, however, can also deal with multi-valued nominal data (e.g. "1" = friend, "2" = kin, "3" = coworker, etc.).

Applying *Network>Roles & Positions>Maximal Regular>CATREGE* to the Wasserman-Faust directed network gives the results shown in Figure 15.4.



```
Number of unique bundles of relationships: 3

HIERARCHICAL CLUSTERING

          A B C D E F G H I

Level     1 2 3 4 5 6 7 8 9
-----     - - - - - - - - -
    2     . XXXXX XXXXXXXXX
    1     XXXXXXXXXXXXXXXXX


Actor-by-actor similarities

         1 2 3 4 5 6 7 8 9
         A B C D E F G H I
         - - - - - - - - -
  1 A    2 1 1 1 1 1 1 1 1
  2 B    1 2 2 2 1 1 1 1 1
  3 C    1 2 2 2 1 1 1 1 1
  4 D    1 2 2 2 1 1 1 1 1
  5 E    1 1 1 1 2 2 2 2 2
  6 F    1 1 1 1 2 2 2 2 2
  7 G    1 1 1 1 2 2 2 2 2
  8 H    1 1 1 1 2 2 2 2 2
  9 I    1 1 1 1 2 2 2 2 2
```

Figure 15.4: *Categorical REGE analysis of Wasserman-Faust directed network*

This result is the same as the one that we did "by hand" earlier in the chapter. A hierarchical clustering diagram can be useful if the equivalences found are inexact, or numerous, and a further simplification is needed. Here, we see at level 2 of the clustering that there are three groups {A}, {B, C, D}, and {E, F, G, H, I}. An image matrix is also produced (but not "reduced" to 3 by 3).

The results can also be usefully visualized with a dendogram, as in Figure 15.5.

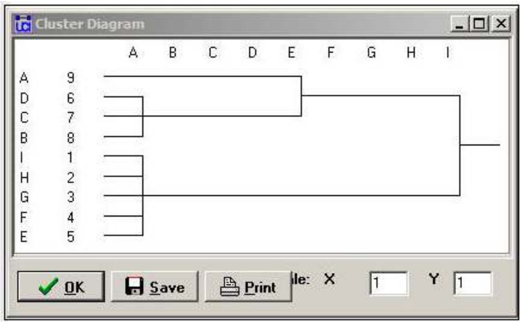


Figure 15.5: *Dendogram of categorical REGE (Figure 15.4)*

We know, from our analysis, that there really are exactly three regular equivalence classes. Should we want to use only two, however, the dendogram suggests that grouping A with B, C, and D would be the most reasonable choice.

Once a regular equivalence blocking has been achieved, it is usually a good idea to produce a permuted and blocked version of the original data so that you can see the tie profiles of each of the classes. One way to do this is to save the permutation vector from *Network>Roles & Positions>Maximal Regular>CATREGE*, and use it to permute the original data (*Data>Permute*).

Categorical REGE for Geodesic Distances (Padgett's Marriage Data)

The Padgett data on marriage alliances among leading Florentine families are of low to moderate density. There are considerable differences among the positions of the families, as can be seen from the graph in Figure 15.6. The data are binary, and not directed. This causes a problem for regular equivalence analysis, because all actors (except isolates) are "equivalent" as "transmitters".

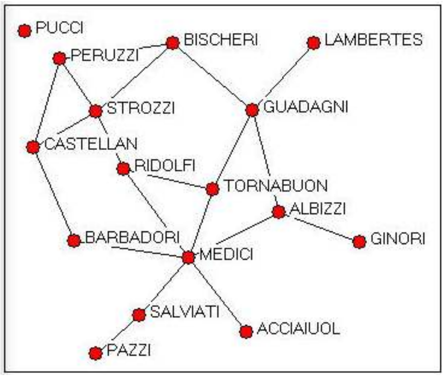
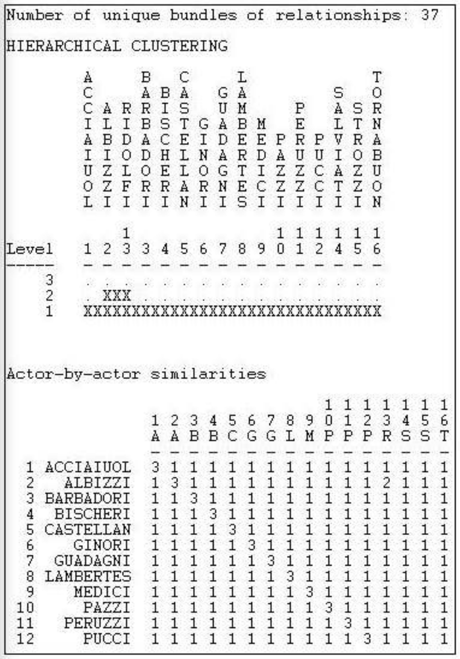


Figure 15.6: *Padgett Florentine marriage alliances*

The categorical REGE algorithm (*Network>Roles & Positions>Maximal Regular>CATREGE*) can be used to identify regularly equivalent actors by treating the elements of the geodesic distance matrix as describing "types" of ties - that is different geodesic distances are treated as "qualitatively" rather than "quantitatively" different. Two nodes are more equivalent if each has an actor in their neighborhood of the same "type". In this case, that means they are similar if they each have an actor that is at the same geodesic distance from themselves. With many data sets, the levels of similarity of neighborhoods can turn out to be quite high - and it may be difficult to differentiate the positions of the actors on "regular" equivalence grounds.

Figure 15.7 shows the results of regular equivalence analysis where geodesic distances have been used to represent multiple qualitative types of relations among actors.



```
Number of unique bundles of relationships: 37

HIERARCHICAL CLUSTERING

        A     B   C     L             T
        C     A B A   G A           S O
        C A R R I S   U M     P   A S R
        I L I B S T G A B M   E   L T N
        A B D A C E I D E E P R P V R A
        I I O D H L N A R D A U U I O B
        U Z L O E L O G T I Z Z C A Z U
        O Z F R R A R N E C Z Z C T Z O
        L I I I I N I I S I I I I I I N

            1             1 1 1 1 1 1
Level   1 2 3 3 4 5 6 7 8 9 0 1 2 4 5 6
-----   - - - - - - - - - - - - - - - -
    3   . . . . . . . . . . . . . . . .
    2   . XXX . . . . . . . . . . . . .
    1   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


Actor-by-actor similarities

                                  1 1 1 1 1 1 1
                1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
                A A B B C G G L M P P P R S S T
                - - - - - - - - - - - - - - - -
 1 ACCIAIUOL    3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 2   ALBIZZI    1 3 1 1 1 1 1 1 1 1 1 1 2 1 1 1
 3 BARBADORI    1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1
 4  BISCHERI    1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1
 5 CASTELLAN    1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1
 6    GINORI    1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1
 7  GUADAGNI    1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1
 8 LAMBERTES    1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1
 9    MEDICI    1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
10     PAZZI    1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
11   PERUZZI    1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1
12     PUCCI    1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1
```

Figure 15.7: *Categorical multi-value analysis (geodesic distance) of Padgett marriage alliances*

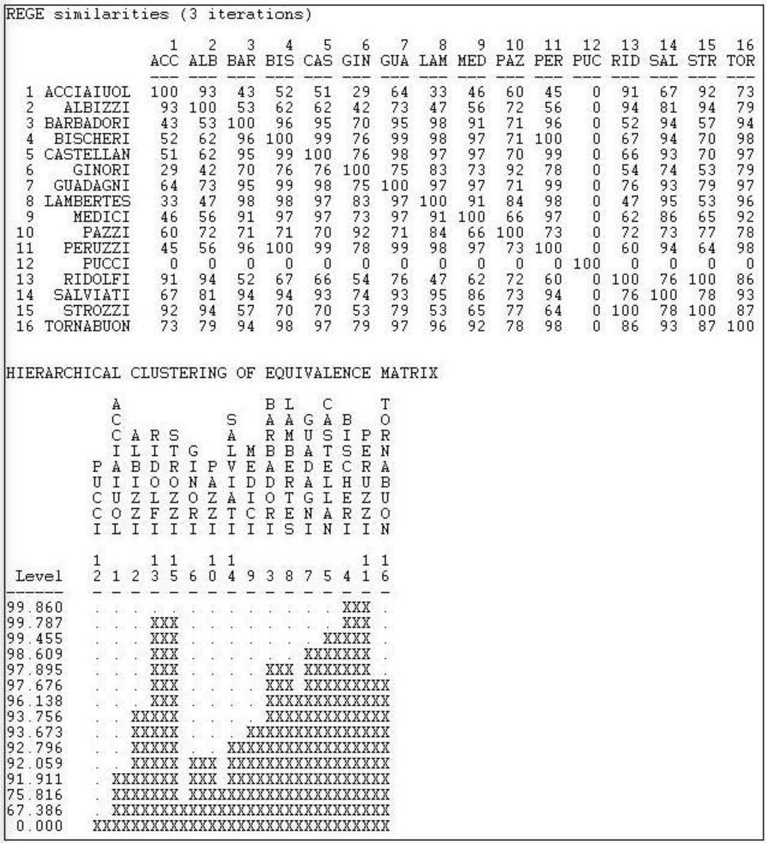
Since the data are highly connected and geodesic distances are short, we are not able to discriminate highly distinctive regular classes in these data. Two families (Albizzi and Ridolfi) do emerge as more similar than others, but generally the differences among profiles is small.

The use of REGE with undirected data, even substituting geodesic distances for binary values, can produce rather unexpected results. It may be more useful to combine a number of different ties to produce continuous values. The main problem, however, is that with undirected data, most cases will appear to be very similar to one another (in the "regular" sense), and no algorithm can really "fix" this. If geodesic distances can be used to represent differences in the types of ties (and this is a conceptual question), and if the actors do have some variability in their distances, this method can produce meaningful results. But, in my opinion, it should be used cautiously, if at all, with undirected data.

Continuous REGE for Geodesic Distances (Padgett's Marriage Data)

An alternative approach to the undirected Padgett data is to treat the different levels of geodesic distances as measures of (the inverse of) strength of ties. Two nodes are said to be more equivalent if they have an actor of similar distance in their neighborhood (similar in the quantitative sense of "5" is more similar to "4" than "6" is). By default, the algorithm extends the search to neighborhoods of distance 3 (though less or more can be selected).

Figure 15.8 shows the results of applying *Network>Roles & Positions>Maximal Regular>REGE* to the Padgett data, using "3 iterations" (that is, three-step neighborhoods).



```
REGE similarities (3 iterations)

                 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
               ACC ALB BAR BIS CAS GIN GUA LAM MED PAZ PER PUC RID SAL STR TOR
               --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
  1 ACCIAIUOL  100  93  43  52  51  29  64  33  46  60  45   0  91  67  92  73
  2   ALBIZZI   93 100  53  62  62  42  73  47  56  72  56   0  94  81  94  79
  3 BARBADORI   43  53 100  96  95  70  95  98  91  71  96   0  52  94  57  94
  4  BISCHERI   52  62  96 100  99  76  99  98  97  71 100   0  67  94  70  98
  5 CASTELLAN   51  62  95  99 100  76  98  97  97  70  99   0  66  93  70  97
  6    GINORI   29  42  70  76  76 100  75  83  73  92  78   0  54  74  53  79
  7  GUADAGNI   64  73  95  99  98  75 100  97  97  71  99   0  76  93  79  97
  8 LAMBERTES   33  47  98  98  97  83  97 100  91  84  98   0  47  95  53  96
  9    MEDICI   46  56  91  97  97  73  97  91 100  66  97   0  62  86  65  92
 10     PAZZI   60  72  71  71  70  92  71  84  66 100  73   0  72  73  77  78
 11   PERUZZI   45  56  96 100  99  78  99  98  97  73 100   0  60  94  64  98
 12     PUCCI    0   0   0   0   0   0   0   0   0   0   0 100   0   0   0   0
 13   RIDOLFI   91  94  52  67  66  54  76  47  62  72  60   0 100  76 100  86
 14  SALVIATI   67  81  94  94  93  74  93  95  86  73  94   0  76 100  78  93
 15   STROZZI   92  94  57  70  70  53  79  53  65  77  64   0 100  78 100  87
 16 TORNABUON   73  79  94  98  97  79  97  96  92  78  98   0  86  93  87 100

HIERARCHICAL CLUSTERING OF EQUIVALENCE MATRIX

           A               B L   C     T
           C           S   A A G A B   O
           C A R S     A   R M U S I P R
           I L I T G   L M B B A T S E N
         P A B D R I P V E A E D E C R A
         U I I O O N A I D D R A L H U B
         C U Z L Z O Z A I O T G L E Z U
         C O Z F Z R Z T C R E N A R Z O
         I L I I I I I I I I S I N I I N

         1     1 1   1 1           1 1
 Level   2 1 2 3 5 6 0 4 9 3 8 7 5 4 1 6
------   - - - - - - - - - - - - - - - -
99.860   . . . . . . . . . . . . . XXX .
99.787   . . . XXX . . . . . . . . XXX .
99.455   . . . XXX . . . . . . . XXXXX .
98.609   . . . XXX . . . . . . XXXXXXX .
97.895   . . . XXX . . . . XXX XXXXXXX .
97.676   . . . XXX . . . . XXX XXXXXXXXX
96.138   . . . XXX . . . . XXXXXXXXXXXXX
93.756   . . XXXXX . . . . XXXXXXXXXXXXX
93.673   . . XXXXX . . . XXXXXXXXXXXXXXX
92.796   . . XXXXX . . XXXXXXXXXXXXXXXXX
92.059   . . XXXXX XXX XXXXXXXXXXXXXXXXX
91.911   . XXXXXXX XXX XXXXXXXXXXXXXXXXX
75.816   . XXXXXXX XXXXXXXXXXXXXXXXXXXXX
67.386   . XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 0.000   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

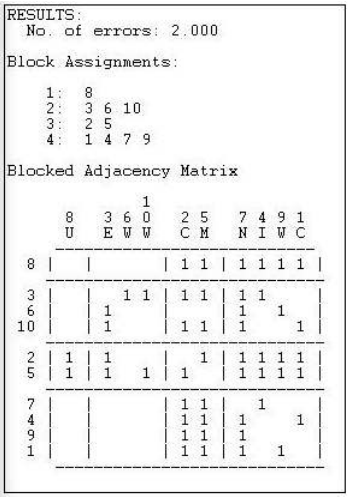
Figure 15.8: *Continuous REGE of Padgett marriage alliance data*

The first panel of the output displays the approximate pair-wise regular similarities as a matrix. Note that the isolated family (Pucci) is treated as a separate class. Also note that these results are finding rather different features of the data than did the categorical treatment. The continuous REGE algorithm applied to the undirected data is probably a better choice than the categorical approach. The result still shows very high regular equivalence among the actors, and the solution is only modestly similar to that of the categorical approach.

The Knoke Bureaucracies Information Exchange Network Analyzed by Tabu Search

At the end of our analysis in the section "Finding Equivalence Sets" above, we produced a "permuted and blocked" version of our data. In doing this, we used a few rules that, in fact, identify what regular equivalence relations "look like". To repeat the main points: we don't care about the ties among members of a regular class; ties between members of a regular class and another class are either all zero, or such that each member of one class has a tie to at least one member of the other class.

This "picture" of what regular classes look like can be used to search for them using numerical methods. The *Network>Roles & Positions>Maximal Regular>Optimization* algorithm seeks to sort nodes into (a user selected number of) categories that come as close to satisfying the "image" of regular equivalence as possible. Figure 15.9 shows the results of applying this algorithm to the Knoke information network.



```
RESULTS:
  No. of errors: 2.000

Block Assignments:

    1:  8
    2:  3 6 10
    3:  2 5
    4:  1 4 7 9

Blocked Adjacency Matrix

              1
      8   3 6 0   2 5   7 4 9 1
      U   E W W   C M   N I W C
     ----------------------------
  8 |   |       | 1 1 | 1 1 1 1 |
    ------------------------------
  3 |   |   1 1 | 1 1 | 1 1     |
  6 |   | 1     |     | 1   1   |
 10 |   | 1     | 1 1 | 1     1 |
    ------------------------------
  2 | 1 | 1     |   1 | 1 1 1 1 |
  5 | 1 | 1   1 | 1   | 1 1 1 1 |
    ------------------------------
  7 |   |       | 1 1 |   1     |
  4 |   |       | 1 1 | 1     1 |
  9 |   |       | 1 1 | 1       |
  1 |   |       | 1 1 | 1   1   |
     -----------------------------
```

Figure 15.9: *Four regular equivalence classes for the Knoke information network by optimum search*

The method produces a fit statistic (number of errors), and solutions for different numbers of partitions should be compared.

The blocked adjacency matrix for the four group solution is, however, quite convincing. Of the 12 block of interest (the blocks on the diagonal are not usually treated as relevant to "role" analysis) 11 satisfy the rules for zero or one blocks perfectly. Only the block connecting sending from {3, 6, 10} to the block {2, 5} fails to satisfy the image of regular equivalence (because actor 6 has no sending ties to either actor 2 or 5).

The solution is also an interesting one substantively. The third set {2, 5} for example, are pure "repeaters" sending and receiving from all other roles. The set {3, 6, 10} send to only two other types (not all three other types) and receive from only one other type. And so on.

The tabu search method can be very useful, and usually produces quite nice results. It is an iterative search algorithm, however, and can find local solutions. Many networks have more than one valid partitioning by regular equivalence, and there is no guarantee that the algorithm will always find the same solution. It should be run a number of times with different starting configurations.

15.S: Regular Equivalence (Summary)

The regular equivalence concept is a very important one for sociologists using social network methods, because it accords well with the notion of a "social role". Two actors are regularly equivalent if they are equally related to equivalent (but not necessarily the same, or same number of) equivalent others. Regular equivalences can be exact or approximate. Unlike the structural and automorphic equivalence definitions, there may be many valid ways of classifying actors into regular equivalence sets for a given graph - and more than one may be meaningful.

There are a number of algorithmic approaches for performing regular equivalence analysis. All are based on searching the neighborhoods of actors and profiling these neighborhoods by the presence of actors of other "types". To the extent that actors have similar "types" of actors at similar distances in their neighborhoods, they are regularly equivalent. This seemingly loose definition can be translated quite precisely into zero and one block rules for making image matrices of proposed regular equivalence blockings. The "goodness" of these images is perhaps the best test of a proposed regular equivalent partitioning. And, the images themselves are the best description of the nature of each "role" in terms of its expected pattern of ties with other roles.

We have only touched the surface of regular equivalence analysis, and the analysis of roles in networks. One major extension that makes role analysis far richer is the inclusion of multiple kinds of ties (that is, stacked or pooled matrices of ties). Another extension is "role algebra" which seeks to identify "underlying" or "generator" or "master" relations from the patterns of ties in multiple tie networks (rather than simply stacking them up or adding them together).

CHAPTER OVERVIEW

16: Multiplex Networks

In this chapter we will look at some of the tools that social network analysts have used to grapple with the complexity of analyzing simultaneous multiple relations among actors. We'll begin by examining some basic data structures for multiplex data, and how they can be visualized. To be useful in analysis, however, the information about multiple relations among a set of actors must somehow be represented in summary form. There are two general approaches: reduction and combination. The "reduction" approach seeks to combine information about multiple relations among the same set of actors into a single relation that indexes the quantity of ties. All of these issues are dealt with in the section on multiplex data basics.

16.1: Introduction to Multiplex Networks

Most of the tools of social network analysis deal with structures defined by patterns in a single kind of relationship among actors: friendship, kinship, economic exchange, warfare, etc. Social relations among actors, however, are usually more complex, in that actors are connected in multiple ways simultaneously. In face-to-face groups of persons, the actors may have emotional connections, exchange relations, kinship ties, and other connections all at the same time. Organizations exchange personnel, money, information, and form groups and alliances. Relations among nation-states are characterized by numerous forms of cultural, economic, and political exchange.

Sociologists tend to assume, until proven otherwise, that actors' behavior is strongly shaped by the complex interaction of many simultaneous constraints and opportunities arising from how the individual is embedded in multiple kinds of relationships. The characteristics and behavior of whole populations, as well, may depend on multiple dimensions of integration/cleavage. Solidarity may be established by economic exchange, shared information, kinship, and other ties operating simultaneously.

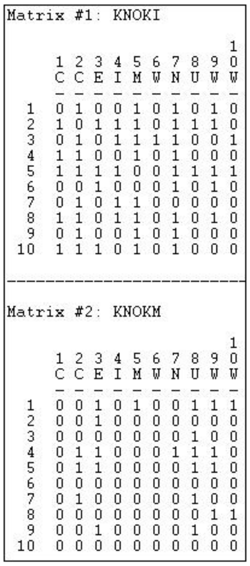
In this chapter we will look at some of the tools that social network analysts have used to grapple with the complexity of analyzing simultaneous multiple relations among actors. We'll begin by examining some basic data structures for multiplex data, and how they can be visualized. To be useful in analysis, however, the information about multiple relations among a set of actors must somehow be represented in summary form. There are two general approaches: reduction and combination. The "reduction" approach seeks to combine information about multiple relations among the same set of actors into a single relation that indexes the quantity of ties. All of these issues are dealt with in the section on multiplex data basics.

The "combination" approach also seeks to create a single index of the multiplex relations, but attempts to represent the quality of ties. Summarizing the information about multiple kinds of ties among actors as a single qualitative typology is discussed in the section on "role algebra". We won't actually explore the complexities of role algebra analysis, but we will provide a brief introduction to this way of approaching multi-relational complexity.

16.2: Multiplex Data Basics

Multiplex data are data that describe multiple relations among the same set of actors. The measures of the relations can be directed or not; and the relations can be recorded as binary, multi-valued nominal, or valued (ordinal or interval).

The most common structure for multiplex data is a set of actor-by-actor matrices (or "slices"), one for each relation. Figure 16.1 shows the output of *Data>Display* for the Knoke social welfare organizations dataset, which contains information on two (binary, directed) relations: information exchange (KNOKI), and money exchange (KNOKM).



```
Matrix #1: KNOKI

                                   1
          1  2  3  4  5  6  7  8  9  0
          C  C  E  I  M  W  N  U  W  W
          -  -  -  -  -  -  -  -  -  -
    1     0  1  0  0  1  0  1  0  1  0
    2     1  0  1  1  1  0  1  1  1  0
    3     0  1  0  1  1  1  1  0  0  1
    4     1  1  0  0  1  0  1  0  0  0
    5     1  1  1  1  0  0  1  1  1  1
    6     0  0  1  0  0  0  1  0  1  0
    7     0  1  0  1  1  0  0  0  0  0
    8     1  1  0  1  1  0  1  0  1  0
    9     0  1  0  0  1  0  1  0  0  0
   10     1  1  1  0  1  0  1  0  0  0

-------------------------------------

Matrix #2: KNOKM

                                   1
          1  2  3  4  5  6  7  8  9  0
          C  C  E  I  M  W  N  U  W  W
          -  -  -  -  -  -  -  -  -  -
    1     0  0  1  0  1  0  0  1  1  1
    2     0  0  1  0  0  0  0  0  0  0
    3     0  0  0  0  0  0  0  1  0  0
    4     0  1  1  0  0  0  1  1  1  0
    5     0  1  1  0  0  0  0  1  1  0
    6     0  0  0  0  0  0  0  0  0  0
    7     0  1  0  0  0  0  0  1  0  0
    8     0  0  0  0  0  0  0  0  1  1
    9     0  0  1  0  0  0  0  1  0  0
   10     0  0  0  0  0  0  0  0  0  0
```

Figure 16.1: *Data>Display of Knoke multi-relational data structure*

The two relations are stored as separate matrices, but within the same file. Many of the analysis tools in UCINET will process each matrix or "slice" of a multiple-matrix data file like the Knoke example. *Data>Unpack* can be used to remove individual matrices from a multiple matrix file; *Data>Join* can be used to create a multiple-matrix dataset from separate single-matrix data files.

The multiple-matrix approach is most general, and allows us to record as many different relations as we wish by using separate matrices. Some matrices may be symmetric and others not; some may be binary, and others valued. A number of the tools that we will discuss shortly, however, will require that the data in the multiple matrices be of the same type (symmetric/asymmetric, binary/valued). So, often it will be necessary to do transformations on individual matrices before "reduction" and "combination" strategies can be applied.

A closely related multiplex data structure is the "Cognitive social structure" or CSS. A CSS records the perceptions of a number of actors of the relations among a set of nodes. For example, we might ask each of Bob, Carol, Ted, and Alice to tell us who among them was friends with whom. The result would be four matrices of the same form (4 actors by 4 actors), reporting the same relation (who's friends with whom), but differing according to who is doing the reporting and perceiving.

CSS data have exactly the same form as standard actor-by-actor slices. And some of the tools used for indexing CSS data are the same. Because of the unique nature of CSS data - which focuses on complex perception of a single structure, instead of a single perception of a complex structure - some additional tools may be applied (more, below).

A third, and rather different, data structure is the multi-valued matrix. Suppose that the relations among actors were nominal (that is, qualitative, or "present-absent") but there were multiple kinds of relations each pair of actors might have - forming a nominal polyotomy. That is, each pair of actors had one (and only one) of several kinds of relations. For one example, relations among a set of actors might (in some populations) be coded as either "nuclear family co-member" or "coworkers" or "extended family member" or "co-religionist" or "none". For another example, we could combine multiple relations to create qualitative types: 1 = kin only, 2 = coworker only, 3 = both kin and coworker, and 4 = neither kin nor coworker.

Nominal, but multi-valued, data combine information about multiplex relations into a single matrix. The values, however, don't represent strength, cost, or probability of a tie, but rather distinguish the qualitative type of tie that exists between each pair of actors. Recording data this way is efficient, and some algorithms in UCINET (e.g. Categorical REGE) can work directly with it.

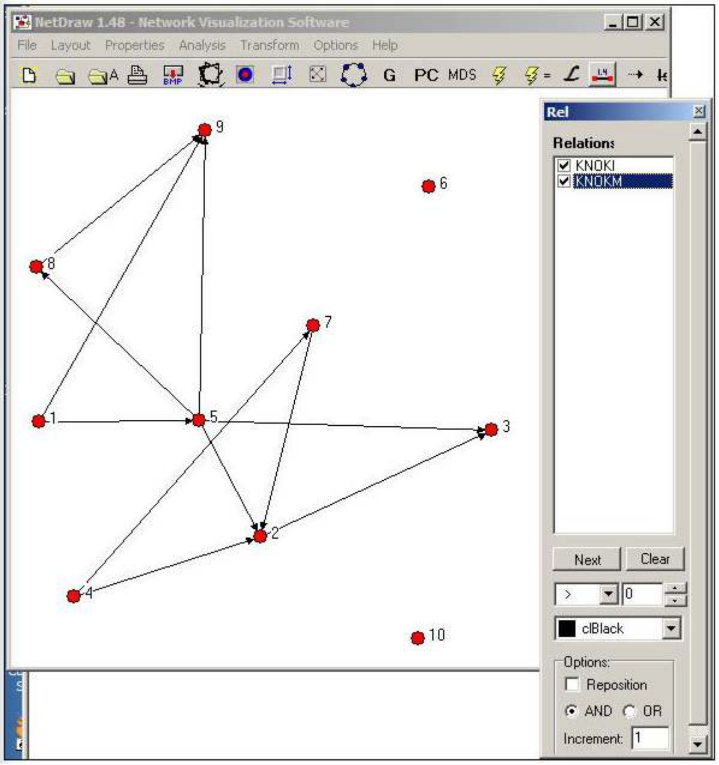
Often, though, data about multiplex relations that has been stored in a single multi-valued matrix will need to be transformed before we can perform many network operations on it.

Visualizing Multiplex Relations

For relatively small networks, drawing graphs is the best way of "seeing" structure. The only new problem is how to represent multiple relations among actors. One approach is to use multiple lines (with different colors or styles) and overlay one relation on another. Alternatively, one can "bundle" the relations into qualitative types and represent them with a single graph using lines of different colors or styles (e.g. kin tie = red; work tie = blue; kin and work tie = green).

Netdraw has some useful tools for visualizing multiple relations among the same set of actors. If the data have been stored as multiple matrices within the same file, when that file is opened (*Netdraw>File>Open>UCINET dataset>Network*) a *Ties* dialog box will allow you to select which matrix to view (as well as to set cut-off values for visualizing valued data). This is useful for flipping back and forth between relations, with the nodes remaining in the same locations. Suppose, for example, we had stored ten matrices in a file, reflecting snapshots of relations in a network as it evolved over some period of time. Using the *Ties* dialog, we can "flip the pages" to see the network evolve.

An even more useful tool is found in *Netdraw>Properties>Lines>Multi-relation Selection*. A drawing of the Knoke network with this dialog box visible is shown in Figure 16.2.



☑ KNOKI
☑ KNOKM

> 0

☐ Reposition
● AND ○ OR
Increment: 1

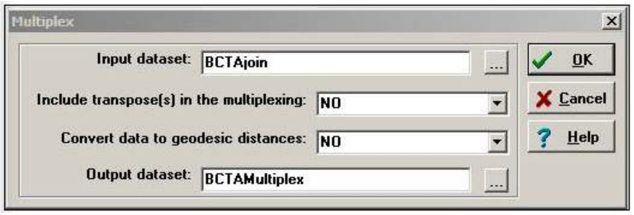
Figure 16.2: *NetDraw graph of Knoke information and money exchange networks*

The Relations dialog box allows you to select which relations you would like to view, and whether to view the union ("or") or intersection ("and") of the ties. In our example, we've asked to see the pattern of ties among organizations that send both information and money to others.

Combining Multiple Relations

For most analyses, the information about the multiple relations among actors will need to be combined into a single summary measure. One common approach is to combine the multiple relations into an index that reflects the quality (or type) of multiplex relation.

Transform>Multiplex can be used to summarize multiple relations among actors into a qualitative multi-valued index. Suppose that we had measured two relations among Bob, Carol, Ted, and Alice. The first is a directed friendship nomination, and the second is an undirected spousal relation. These two four-by-four binary matrices have been packed into a single data file called BCTAjoin. The dialog for Transform>Multiplex is shown as Figure 16.3.



Input dataset: BCTAjoin

Include transpose(s) in the multiplexing: NO

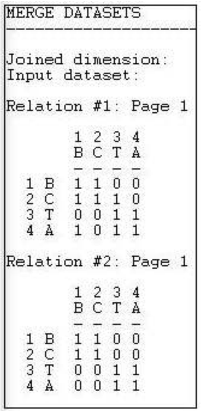
Convert data to geodesic distances: NO

Output dataset: BCTAMultiplex

Figure 16.3: *Transform>Multiplex dialog*

There are two choices here. *Convert data to geodesic distances* allows us to first convert each relation into a valued metric from the binary. We've chosen not to do this. Another choice is whether or not to *Include transpose(s) in the multiplexing*. For asymmetric data, selecting *yes* will cause the rows and the columns of the input matrix to be treated as separate relations in forming the qualitative combinations. Again, we've chosen not to do this (though it is a reasonable idea in many real cases).

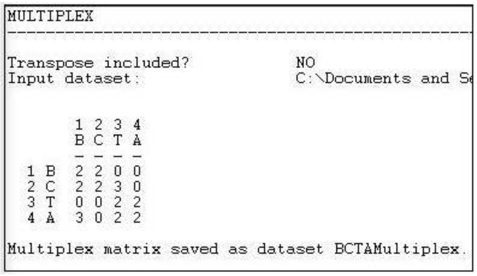
Figure 16.4 shows the input file, which is composed of two "stacked" or "sliced" matrices representing friendship and spousal ties.



```
MERGE DATASETS
--------------------------

Joined dimension:
Input dataset:

Relation #1: Page 1

       1 2 3 4
       B C T A
       - - - -
 1 B   1 1 0 0
 2 C   1 1 1 0
 3 T   0 0 1 1
 4 A   1 0 1 1

Relation #2: Page 1

       1 2 3 4
       B C T A
       - - - -
 1 B   1 1 0 0
 2 C   1 1 0 0
 3 T   0 0 1 1
 4 A   0 0 1 1
```

Figure 16.4: *Transform>Multiplex input*

Figure 16.5 shows the resulting "typology" of kinds of relations among the actors, which has been generated as a multi-valued nominal index.



```
MULTIPLEX
--------------------------------------------------

Transpose included?           NO
Input dataset:                C:\Documents and Se


       1 2 3 4
       B C T A
       - - - -
 1 B   2 2 0 0
 2 C   2 2 3 0
 3 T   0 0 2 2
 4 A   3 0 2 2

Multiplex matrix saved as dataset BCTAMultiplex.
```

Figure 16.5: *Transform>Multiplex output*

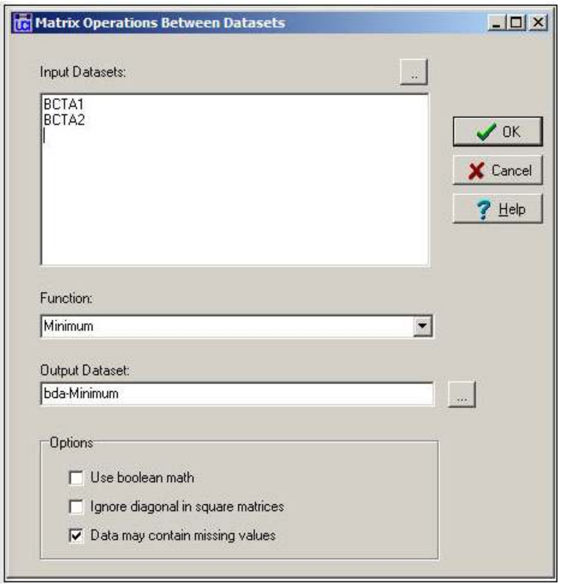
Where there is no tie in either matrix, the type "0" has been assigned. Where there is both a friendship and a spousal tie, the number "2" has been assigned; where there is a friendship tie, but no spousal tie, the number "3" has been assigned. There could have been an additional type (spousal tie, but no friendship) which would have been assigned a different number.

Combining multiple relations in this way yields a qualitative typology of the kinds of relations that exist among actors. An index of this type might be of considerable interest in describing the prevalence of the types in a population, and in selecting sub-graphs for closer analysis.

The operation Transform>Multigraph does the reverse of what Transform>Multiplex does. That is, if we begin with a multi-valued single matrix (as in Figure 16.5), this operation will split the data and create a multiple matrix data file with one matrix for each "type" of relation. In the case of our example, *Transform>Multigraph* would generate two new matrices (one describing the "2" relation, and one describing the "3" relation).

In dealing with multiple relations among actors, we might also want to create a quantitative index that combines the relations. For example, we might suppose that if actors are tied by 4 different relations they share a "stronger" tie than if they share only 3 relations. But, there are many possible ways of creating indexes that capture different aspects or dimensions of the multiple relations among actors. Two tool-kits in UCINET support combining multiple matrices with a wide variety of built-in functions for capturing different aspect of the multi-relational data.

Transform>Matrix Operations>Matrix Operations>Between Datasets>Statistical Summaries provides some basic tools for creating a single valued matrix from multiple matrices. Figure 16.6 shows the dialog for this tool.



Matrix Operations Between Datasets

Input Datasets:
BCTA1
BCTA2

Function: Minimum

Output Dataset: bda-Minimum

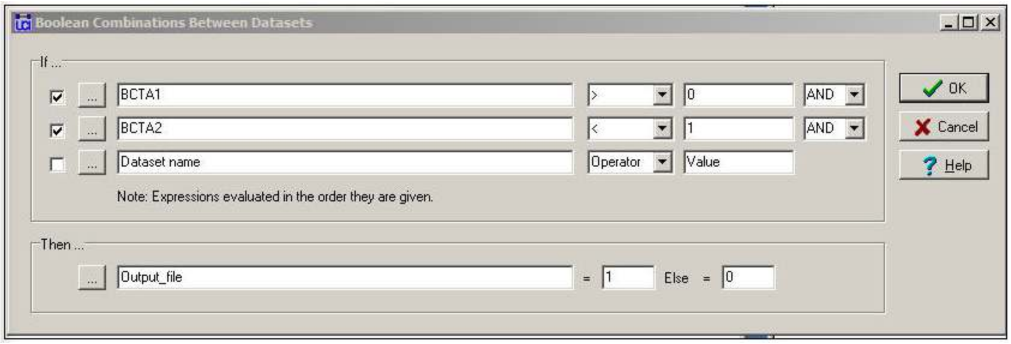
Options
☐ Use boolean math
☐ Ignore diagonal in square matrices
☑ Data may contain missing values

Figure 16.6: *Dialog for between dataset matrix operations - statistical summaries*

In the example, we've selected the two separate single-relation matrices for Bob, Carol, Ted, and Alice, and asked to create a new (single matrix) dataset called bda-Minimum. By selecting the *Minimum* function, we've chosen a rule that says: look at relations across the matrices, and summarize each pair-wise relation as the weakest one. For binary data, this is the same as the logical operation "and".

Also available in this dialog are *Sum* (which adds the values, element-wise, across matrices); *Average* (which computes the mean, element-wise across matrices); *Maximum* (which selects the largest value, element-wise); and *Element-wise Multiplication* (which multiplies the elements across matrices). This is a pretty useful tool kit, and captures most of the ways in which quantitative indexes might be created (weakest tie, strongest tie, average tie, interaction of ties).

We might want to combine the information on multiple relations into a quantitative index by using logical operations instead of numeric. Figure 16.7 shows the dialog for *Transform>Matrix Operations>Matrix Operations>Between Datasets>Boolean Combinations*.



Boolean Combinations Between Datasets

If ...

☑	BCTA1	>	0	AND
☑	BCTA2	<	1	AND
☐	Dataset name	Operator	Value	

Note: Expressions evaluated in the order they are given.

Then ...

Output_file = 1 Else = 0

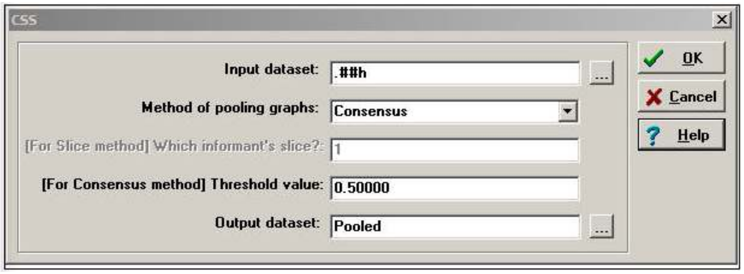
Figure 16.7: *Dialog for between dataset matrix operations - Boolean combinations*

In this dialog, we've said: if there is a friendship tie and there is no spousal tie, then code the output relation as "1". Otherwise, code the output relation as "0". This is not a very sensible thing to do, but it illustrates the point that this tool can be used to perform basic logical operations to create valued (or binary) indexes that combine the information on multiple relations.

Combining Multiple Views

Suppose that I asked every member of the faculty of my department to fill out a questionnaire reporting on their perceptions of who likes whom among the faculty. We would be collecting "cognitive social structure" data; that is, reports from actors embedded in a network about the whole network. There is a very interesting research literature that explores the relationship between actor's positions in networks, and their perceptions of the network. For example, do actors have a bias toward perceiving their own positions as more "central" than other actors' perceptions of their centrality?

A cognitive social structure (CSS) dataset contains multiple actor-by-actor matrices. Each matrix reports on the full set of a single relation among all the actors, as perceived by a particular respondent. While we could use many of the tools discussed in the previous section to combine or reduce data like these into indexes, there are some special tools that apply to cognitive data. Figure 16.8 shows the dialog of *Data>CSS*, which provides access to some specialized tools for cognitive network research.



Input dataset: .##h

Method of pooling graphs: Consensus

[For Slice method] Which informant's slice?: 1

[For Consensus method] Threshold value: 0.50000

Output dataset: Pooled

Figure 16.8: *Dialog for Data>CSS*

The key element here is the choice of Method for pooling graphs. In creating a single summary of the relations, we could select the perceptions of a single actor; or, we might want to focus on the perceptions of the pair of actors involved in each particular relationship; or we might want to combine the information of all of the actors in the network.

Slice selects the perception of one particular actor to represent the network (the dialog then asks, "which informant?"). If we had a particular expert informant we might choose his/her view of the network as a summary. Or, we could extract multiple different actors into different files. We might also extract actors based on some attribute (e.g. gender) and extract their graphs, then pool them by some other method.

Row LAS uses the data from each actor's row to be the row entry in the output matrix. That is, actor A's perceptions of his/her row values are used for row A in the output matrix; actor B's perceptions of his/her row values are used for row B in the output matrix. This uses each actor as the "informant" about their own out-ties.

Column LAS uses each actor's column to be the column entry in the output matrix. That is, each actor is being used as the "informant" regarding their own in-ties.

Intersection LAS constructs the output matrix by examining the entries of the particular pair of actors involved. For example, in the output matrix we would have an element that described the relation between Bob and Ted. We have data on how Bob, Ted, Carol, and Alice each perceive the relation of Bob and Ted. The LAS method focuses on only the two involved nodes (Bob and Ted) and ignores the others. The intersection method gives a "1" to the tie if both Bob and Ted say there is a tie, and a "0" otherwise.

Union LAS assigns a "1" to the pair-wise relation if either actor (i.e. either Bob or Ted) says there is a tie.

Median LAS selects the median of the two values for the B,T relation that are reported by B and by T. This is useful if the relation being examined is valued, rather than binary.

Consensus uses the perceptions of all actors to create the summary index. The perceptions of Bob, Carol, Ted, and Alice are summed, and if the sum is greater than a user-specified cut-off value, "1" is assigned, else "0".

Average calculates the numerical average of all actors' perceptions of each pair-wise tie.

Sum calculates the sum of all actors' perceptions for each pair-wise tie.

The range of choices here suggests a fertile research area in how actors embedded in relations perceive those relations. The variety of indexing methods also suggests a number of interesting questions about, and methods for dealing with, the reliability of network data when it is collected from embedded respondents.

16.3: Role Algebra for Mulitplex Data

Let's suppose that we were looking at a single matrix on who was friends with whom. An obvious way of characterizing what we see is to classify each pair as "friends" or "not friends". But now, let's extend our analysis one step further (or look at paths of length 2). Now each pair of actors could be characterized as friend, not friend, friend of friend, friend of not-friend, not-friend of friend, or not-friend of not-friend. If we wanted to consider paths of length three...well, you get the idea.

The notion of a "role algebra" is to understand the relations between actors as realizations of the logically possible "compounds" of relations of selected path lengths. Most often in network analyses, we focus on path of length one (two actors are connected or not). But, sometimes it is useful to characterize a graph as containing more complex kinds of relations (friend of friend, not-friend of friend, etc.). Lists of these kinds of relations can be obtained by taking Boolean products of matrices (i.e. 0 * 0 = 0, 0 * 1 = 0, 1 * 0 = 0, and 1 * 1 = 1). When applied to single matrix, we raise a matrix to a power (multiply it by itself) and take the Boolean product; the result generates a matrix that tells us if there is a relation between each pair of nodes that is of a path length equal to the power. That is, to find whether each pair of actors is connected by the relation "friend of a friend" we take the Boolean product of the friendship matrix squared.

This (elegant, but rather mysterious) method of finding "compound relations" can be applied to multiplex data as a way of identifying the kinds of relations that exist in a multiplex graph. The *Transform>Semigroup* algorithm can be used to identify these more complex qualitative kinds of relations among nodes.

It is easier for most people to understand this with an example, than in the abstract. So let's do a somewhat extended examination of the Knoke data for both information and money ties.

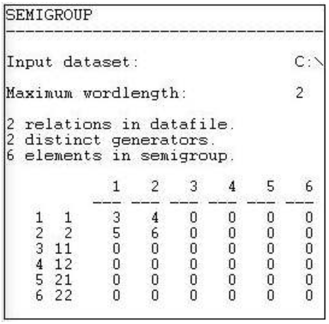
If we consider just direct relations, there are two: organizations can be tied by information; organizations can be tied by money. What if we consider relations at two steps (what are called "word lengths" in role algebra)? In addition to the original two relations, there are now four more:

- When we multiply the information matrix by its transpose and take Boolean products, we are identifying linkages like "sends information to a node that send information to..."
- When we multiply the money matrix by its transpose and take Boolean products, we are identifying the linkage: "sends money to a node that send money to..."
- When we multiply the information matrix by the money matrix, we are identifying the relationship: "sends information to a node that sends money to..."
- When we multiply the money matrix by the information matrix, we are identifying the relationship: "sends money to a node that sends information to..."

These four new (two-step) relations among nodes are "words" of length two, or "compounds".

It is possible, of course, to continue to compound to still greater lengths. In most sociological analyses with only two types of ties, longer lengths are rarely substantively meaningful. With more kinds of ties, however, the number of types of compound relationships can become quite large quite quickly.

The tool *Transform>Semigroup* computes all of the logically possible compounded types of relations up to a word length (i.e. network distance) that the user specifies. It produces a log file that contains a "map" of the types of relations, as we see in Figure 16.9. It also produces, in a separate file, adjacency matrices for each of the types of relationships (Figures 16.10 and 16.11).

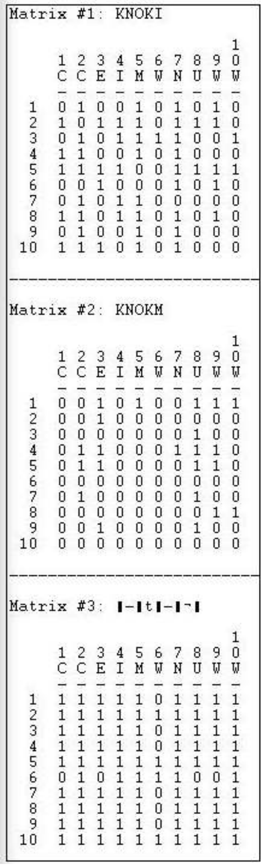


```
SEMIGROUP
----------------------------------

Input dataset:                  C:\

Maximum wordlength:             2

2 relations in datafile.
2 distinct generators.
6 elements in semigroup.

                1    2    3    4    5    6
              ---  ---  ---  ---  ---  ---
    1   1       3    4    0    0    0    0
    2   2       5    6    0    0    0    0
    3  11       0    0    0    0    0    0
    4  12       0    0    0    0    0    0
    5  21       0    0    0    0    0    0
    6  22       0    0    0    0    0    0
```

Figure 16.9: *Semigroups of word-length 2 for Knoke information and money networks*

The output tells us that there were two relations (information and money). These were the "generators" that were used to create the types. Six possible compound relations were generated for the word-length 2 (identified down the left-hand side). Relations 1 and 2 are information and money individually - the original matrices. Relation 3 is a compound of information with itself; relation four is the compound of information with money, etc. The numbers (3, 4, 5, 6) are simply guides to which matrix in the output file refers to which relation.

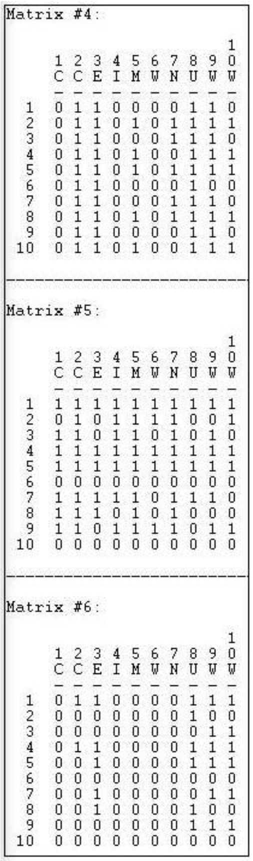
From these new "types" of relations (which are compounds within and between the two types of ties) we can generate new adjacency matrices that show which pairs of actors are joined by each particular type of relation. These are presented as a series of adjacency matrices, as shown in Figure 16.10 and continued in Figure 16.11.



```
Matrix #1: KNOKI

                                     1
         1  2  3  4  5  6  7  8  9  0
         C  C  E  I  M  W  N  U  W  W
         -  -  -  -  -  -  -  -  -  -
    1    0  1  0  0  1  0  1  0  1  0
    2    1  0  1  1  1  0  1  1  1  0
    3    0  1  0  1  1  1  1  0  0  1
    4    1  1  0  0  1  0  1  0  0  0
    5    1  1  1  1  0  0  1  1  1  1
    6    0  0  1  0  0  0  1  0  1  0
    7    0  1  0  1  1  0  0  0  0  0
    8    1  1  0  1  1  0  1  0  1  0
    9    0  1  0  0  1  0  1  0  0  0
   10    1  1  1  0  1  0  1  0  0  0

--------------------------------------

Matrix #2: KNOKM

                                     1
         1  2  3  4  5  6  7  8  9  0
         C  C  E  I  M  W  N  U  W  W
         -  -  -  -  -  -  -  -  -  -
    1    0  0  1  0  1  0  0  1  1  1
    2    0  0  1  0  0  0  0  0  0  0
    3    0  0  0  0  0  0  0  1  0  0
    4    0  1  1  0  0  0  1  1  1  0
    5    0  1  1  0  0  0  0  1  1  0
    6    0  0  0  0  0  0  0  0  0  0
    7    0  1  0  0  0  0  0  1  0  0
    8    0  0  0  0  0  0  0  0  1  1
    9    0  0  1  0  0  0  0  1  0  0
   10    0  0  0  0  0  0  0  0  0  0

--------------------------------------

Matrix #3: ▌-▌t▌-▌¬▌

                                     1
         1  2  3  4  5  6  7  8  9  0
         C  C  E  I  M  W  N  U  W  W
         -  -  -  -  -  -  -  -  -  -
    1    1  1  1  1  1  0  1  1  1  1
    2    1  1  1  1  1  1  1  1  1  1
    3    1  1  1  1  1  0  1  1  1  1
    4    1  1  1  1  1  0  1  1  1  1
    5    1  1  1  1  1  1  1  1  1  1
    6    0  1  0  1  1  1  1  0  0  1
    7    1  1  1  1  1  0  1  1  1  1
    8    1  1  1  1  1  0  1  1  1  1
    9    1  1  1  1  1  0  1  1  1  1
   10    1  1  1  1  1  1  1  1  1  1
```

Figure 16.10: *Relations tables for Figure 16.9 (part 1)*

Matrix 1 is simply the original information matrix; matrix 2 is the original money matrix. Matrix 3 is the compound of information with information - which actors are tied by a relationship "Ego sends information to someone who sends information to Alter"?



```
Matrix #4:

                              1
       1  2  3  4  5  6  7  8  9  0
       C  C  E  I  M  W  N  U  W  W
       -  -  -  -  -  -  -  -  -  -
  1    0  1  1  0  0  0  0  1  1  0
  2    0  1  1  0  1  0  1  1  1  1
  3    0  1  1  0  0  0  1  1  1  0
  4    0  1  1  0  1  0  0  1  1  1
  5    0  1  1  0  1  0  1  1  1  1
  6    0  1  1  0  0  0  0  1  0  0
  7    0  1  1  0  0  0  1  1  1  0
  8    0  1  1  0  1  0  1  1  1  1
  9    0  1  1  0  0  0  0  1  1  0
 10    0  1  1  0  1  0  0  1  1  1

--------------------------------------

Matrix #5:

                              1
       1  2  3  4  5  6  7  8  9  0
       C  C  E  I  M  W  N  U  W  W
       -  -  -  -  -  -  -  -  -  -
  1    1  1  1  1  1  1  1  1  1  1
  2    0  1  0  1  1  1  1  0  0  1
  3    1  1  0  1  1  0  1  0  1  0
  4    1  1  1  1  1  1  1  1  1  1
  5    1  1  1  1  1  1  1  1  1  1
  6    0  0  0  0  0  0  0  0  0  0
  7    1  1  1  1  1  0  1  1  1  0
  8    1  1  1  0  1  0  1  0  0  0
  9    1  1  0  1  1  1  1  0  1  1
 10    0  0  0  0  0  0  0  0  0  0

--------------------------------------

Matrix #6:

                              1
       1  2  3  4  5  6  7  8  9  0
       C  C  E  I  M  W  N  U  W  W
       -  -  -  -  -  -  -  -  -  -
  1    0  1  1  0  0  0  0  1  1  1
  2    0  0  0  0  0  0  0  1  0  0
  3    0  0  0  0  0  0  0  0  1  1
  4    0  1  1  0  0  0  0  1  1  1
  5    0  0  1  0  0  0  0  1  1  1
  6    0  0  0  0  0  0  0  0  0  0
  7    0  0  1  0  0  0  0  0  1  1
  8    0  0  1  0  0  0  0  1  0  0
  9    0  0  0  0  0  0  0  1  1  1
 10    0  0  0  0  0  0  0  0  0  0
```

Figure 16.11: *Relations tables for Figure 16.9 (part 2)*

Matrix 4 is the compound of money with itself, or "Ego sends money to someone who sends money to Alter".

Matrices 5 and 6 are, in some ways, the most interesting. While exchanging information for information and money for money are obvious ways in which a network can be integrated, it is also possible that actors can be integrated by relations that involve both "apples" and "oranges". That is, I may send money, and receive information; I may send information, and receive money.

Role algebra has proven to be of particular value in the study of kinship relations, where across-generation (parent/child) ties are recorded in one matrix and within-generation relations are recorded in another. The various compounds (e.g. "child of child", "child of brother") fairly easily capture the meaningful terms in kinship relations.

16.S: Multiplex Networks (Summary)

The actors in the kinds of networks that social scientists study are very frequently connected by more than one type of tie, simultaneously. That is, the relationship between any two actors may be multiplex. In this chapter, we've introduced a few of the tools that are commonly used to help to make sense of the complex patterns of embedding that can emerge when there is more than one kind of tie operating simultaneously.

Multiplex data are usually stored in a data structure of node-by-node matrices that are "stacked" as "slices" in a single file. Usually, these structures contain slices that measure different relations (e.g. money, information). However, the same data structure can be effectively used to store and work with multiple slices that show the state of the same network at multiple points in time, or the same network as perceived by different observers embedded in it (Cognitive social structures, or CSS). A compact way of storing information about multiple kinds of relations among actors in a single matrix, the multi-valued matrix, uses a number to reflect the qualitative type of relation that exists between two actors (e.g. none, money only, information only, information and money; or mutually exclusive "multiple choice" types like: kin, neighbor, coworker).

With relatively small networks, and relatively small numbers of relations, graphs can be prepared that show the unions nd intersections of multiple kinds of relations, or "animate" change over time in network structure.

Usually the information about multiple kinds of relations among actors is indexed by reducing the multiple ties into a single quantitative value that represents a summary across the separate relations (e.g. average tie strength, maximum, minimum). Alternatively, the information about different kinds of ties may be combined into more complex typologies using logical relations and "role algebra". A special set of tools for dealing with the unique features of CSS data was also discussed.

Many social network studies avoid the complexity of multiplex data by focusing on a single relation, or by dealing with multiple relations separately. There is a good bit of virtue in this, for multiplex analysis can be quite demanding (at least there are many plausible ways of approaching any multi-relational problem). Still, in some cases, engaging the full complexity of multiplex data has pair huge returns. Our understanding of kinship structures, and our understanding of the positions of nation-states in the world system have been greatly enhanced by indexing actors' relational positions based on multiple and simultaneous ties.

CHAPTER OVERVIEW

17: Two-Mode Networks

Two-mode data offer some very interesting analytic possibilities for gaining greater understanding of "macro-micro" relations. In the Davis data, for example, we can see how the choices of the individual women "make" the meaning of the parties by choosing to attend or not. We can also see how the parties, as macro structures, may affect the choices of the individual women. With a little creativity, you can begin to see examples of these kinds of two-mode, or macro-micro social structures everywhere. The social world is one of "nesting" in which individuals (and larger structures) are embedded in larger structures (and larger structures are embedded in still larger ones). Indeed, the analysis of the tension between "structure and agency" or "macro and micro" is one of the core themes in sociological theory and analysis. In this chapter we will take a look at some of the tools that have been applied (and, in some cases, developed) by social network analysts for examining two-mode data. We begin with a discussion of data structures, proceed to visualization, and then turn our attention to techniques for identifying quantitative and qualitative patterns in two-mode data.

17.1: Introduction to Two-Mode Networks

For a classic study of the American south (Deep South, University of Chicago Press, 1941), Davis and his colleagues collected data on which of 18 women were present at each of the 14 events of the "social season" in a community. By examining patterns of which women are present (or absent) at which events, it is possible to infer an underlying pattern of social ties, factions, and groupings among the women. At the same time, by examining which women were present at the 14 events, it is possible to infer underlying patterns in the similarity of the events.

The Davis study is an example of what Rom Breiger (1974) called "The duality of persons and groups". Breiger is calling attention to the dual focus of social network analysis on how individuals, by their agency, create social structures while, at the same time, social structures develop an institutionalized reality that constrains and shapes the behavior of the individuals embedded in them.

The data used for social network analysis, most commonly, measure relations at the micro level, and use analysis techniques to infer the presence of social structure at the macro level. For example, we examine the ties of individuals (micro) for patterns that allow us to infer macro structure (i.e. cliques).

The Davis data is a bit different. It describes ties between two sets of nodes at two different levels of analysis. The ties that Davis identifies are between actors (the women) and events (the parties of the social season). Data like these involve two levels of analysis (or two "modes"). Often, such data are termed "affiliation" data because they describe which actors are affiliated with (present, or members of) which macro structures.

Two-mode data offer some very interesting analytic possibilities for gaining greater understanding of "macro-micro" relations. In the Davis data, for example, we can see how the choices of the individual women "make" the meaning of the parties by choosing to attend or not. We can also see how the parties, as macro structures, may affect the choices of the individual women.

With a little creativity, you can begin to see examples of these kinds of two-mode, or macro-micro social structures everywhere. The social world is one of "nesting" in which individuals (and larger structures) are embedded in larger structures (and larger structures are embedded in still larger ones). Indeed, the analysis of the tension between "structure and agency" or "macro and micro" is one of the core themes in sociological theory and analysis.

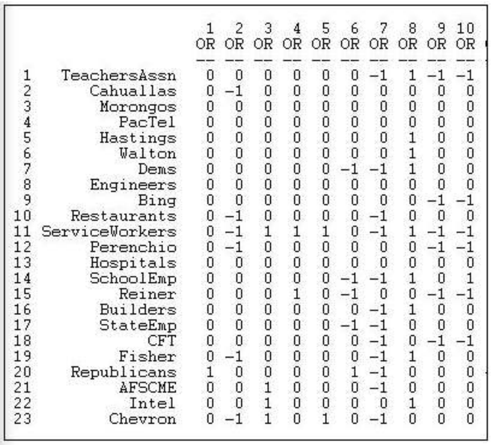
In this chapter we will take a look at some of the tools that have been applied (and, in some cases, developed) by social network analysts for examining two-mode data. We begin with a discussion of data structures, proceed to visualization, and then turn our attention to techniques for identifying quantitative and qualitative patterns in two-mode data.

For most of the examples in this chapter we will use a new 2-mode dataset from a problem that I happen to be working on in parallel with this chapter. The data describe the contributions of a small number of large donors (those who gave a total of at least $1,000,000) to campaigns supporting and opposing ballot initiatives in California during the period 2000 to 2004. We've included 44 of the initiatives. The dataset has two modes: donors and initiatives.

We will use two different forms of the data - one valued and one binary. The valued data describe the relations between donors and initiatives using a simple ordinal scale. An actor is coded as -1 if they gave a contribution opposing a particular initiative, 0 if they did not contribute, and +1 if they contributed in support of the initiative. The binary data describe whether a donor did (+1) or did not (0) contribute in the campaign on each initiative.

17.2: Bipartite Data Structures

The most common way of storing 2-mode data is a rectangular data matrix of actors (rows) by events (columns). Figure 17.1 shoes a portion of the valued dataset we will use here (*Data>Display*).

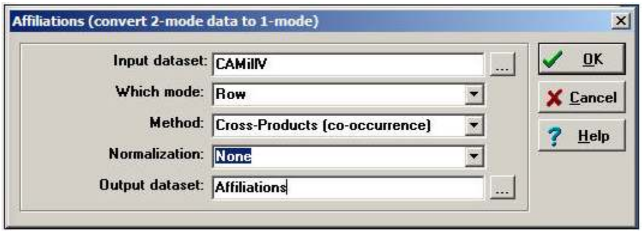


```
                         1  2  3  4  5  6  7  8  9 10
                        OR OR OR OR OR OR OR OR OR OR
                        -- -- -- -- -- -- -- -- -- --
 1     TeachersAssn      0  0  0  0  0  0 -1  1 -1 -1
 2        Cahuallas      0 -1  0  0  0  0  0  0  0  0
 3         Morongos      0  0  0  0  0  0  0  0  0  0
 4           PacTel      0  0  0  0  0  0  0  0  0  0
 5         Hastings      0  0  0  0  0  0  0  1  0  0
 6           Walton      0  0  0  0  0  0  0  1  0  0
 7             Dems      0  0  0  0  0 -1 -1  1  0  0
 8        Engineers      0  0  0  0  0  0  0  0  0  0
 9             Bing      0  0  0  0  0  0  0  0 -1 -1
10      Restaurants      0 -1  0  0  0  0 -1  0  0  0
11   ServiceWorkers      0 -1  1  1  1  0 -1  1 -1 -1
12        Perenchio      0 -1  0  0  0  0  0  0 -1 -1
13        Hospitals      0  0  0  0  0  0  0  0  0  0
14        SchoolEmp      0  0  0  0  0 -1 -1  1  0  1
15           Reiner      0  0  0  1  0 -1  0  0 -1 -1
16         Builders      0  0  0  0  0  0 -1  1  0  0
17         StateEmp      0  0  0  0  0 -1 -1  0  0  0
18              CFT      0  0  0  0  0  0 -1  0 -1 -1
19           Fisher      0 -1  0  0  0  0 -1  1  0  0
20      Republicans      1  0  0  0  0  1 -1  0  0  0
21           AFSCME      0  0  1  0  0  0 -1  0  0  0
22            Intel      0  0  1  0  0  0  0  1  0  0
23          Chevron      0 -1  1  0  1  0 -1  0  0  0
```

Figure 17.1: *Rectangular data array of California political donations data*

The California Teachers Association, for example, gave donations in opposition to the 7th, 9th, and 10th ballot initiatives, and a donation supporting the 8th.

A very common and very useful approach to two-mode data is to convert it into two one-mode datasets, and examine relations within each mode separately. For example, we could create a dataset of actor-by-actor ties, measuring the strength of the tie between each pair of actors by the number of times that they contributed on the same side of initiatives, summed across the 40-some initiatives. We could also create a one-mode dataset of initiative-by-initiative ties, coding the strength of the relation as the number of donors that each pair of initiatives had in common. The *Data>Affiliations* tool can be used to create one-mode datasets from a two-mode rectangular data array. Figure 17.2 displays a typical dialog box.



Affiliations (convert 2-mode data to 1-mode)

Input dataset: CAMilV

Which mode: Row

Method: Cross-Products (co-occurrence)

Normalization: None

Output dataset: Affiliations

Figure 17.2: *Dialog of Data>Affiliations to create actor-by-actor relations of California donors*

There are several choices here.

We have selected the *row* mode (actors) for this example. To create an initiative-by-initiative one-mode dataset, we would have selected *column*.

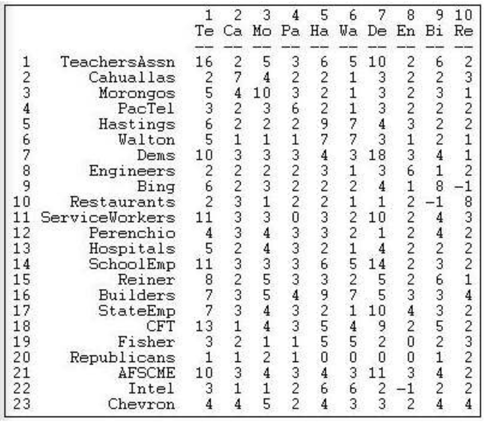
There are two alternative methods:

The cross-product method takes each entry of the row for actor A, and multiplies it times the same entry for actor B, and then sums the result. Usually, this method is used for binary data because the result is a count of co-occurrence. With binary data, each product is 1 only if both actors were "present" at the event, and the sum across events yields the number of events in common - a valued measure of strength.

Our example is a little more complicated because we've applied the cross-product method to valued data. Here, if neither actor donated to an initiative (0 * 0 = 0), or if one donated and the other did not (0 * -1 or 0 * +1 = 0), there is no tie. If both donated in the same direction (-1 * -1 or +1 * +1 = 1), there is a positive tie. If both donated, but in opposite directions (+1 * -1 = -1), there is a negative tie. The sum of the cross-products is a valued count of the preponderance of positive or negative ties.

The minimums method examines the entries for the two actors at each event, and selects the minimum value. For binary data, the result is the same as the cross-product method (if both, or either actor is zero, the minimum is zero; only if both are one is the minimum one). For valued data, the minimums method is essentially saying: the tie between the two actors is equal to the weaker of the ties of the two actors to the event. This approach is commonly used when the original data are measured as valued.

Figure 17.3 shows the result of applying the cross-products method to our valued data.



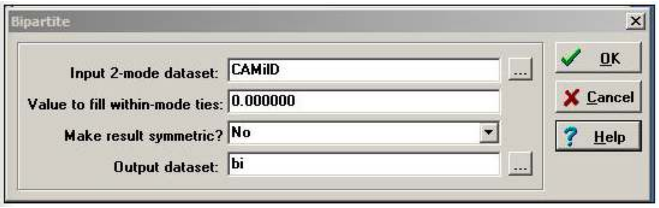
```
                    1   2   3   4   5   6   7   8   9  10
                   Te  Ca  Mo  Pa  Ha  Wa  De  En  Bi  Re
                   --  --  --  --  --  --  --  --  --  --
 1   TeachersAssn  16   2   5   3   6   5  10   2   6   2
 2      Cahuallas   2   7   4   2   2   1   3   2   2   3
 3       Morongos   5   4  10   3   2   1   3   2   3   1
 4         PacTel   3   2   3   6   2   1   3   2   2   2
 5       Hastings   6   2   2   2   9   7   4   3   2   2
 6         Walton   5   1   1   1   7   7   3   1   2   1
 7           Dems  10   3   3   3   4   3  18   3   4   1
 8      Engineers   2   2   2   2   3   1   3   6   1   2
 9           Bing   6   2   3   2   2   2   4   1   8  -1
10    Restaurants   2   3   1   2   2   1   1   2  -1   8
11 ServiceWorkers  11   3   3   0   3   2  10   2   4   3
12      Perenchio   4   3   4   3   3   2   1   2   4   2
13      Hospitals   5   2   4   3   2   1   4   2   2   2
14      SchoolEmp  11   3   3   3   6   5  14   2   3   2
15         Reiner   8   2   5   3   3   2   5   2   6   1
16       Builders   7   3   5   4   9   7   5   3   3   4
17       StateEmp   7   3   4   3   2   1  10   4   3   2
18            CFT  13   1   4   3   5   4   9   2   5   2
19         Fisher   3   2   1   1   5   5   2   0   2   3
20    Republicans   1   1   2   1   0   0   0   0   1   2
21         AFSCME  10   3   4   3   4   3  11   3   4   2
22          Intel   3   1   1   2   6   6   2  -1   2   2
23        Chevron   4   4   5   2   4   3   3   2   4   4
```

Figure 17.3: *Actor-by-actor tie strengths (Figure 17.2)*

The teachers association participated in 16 campaigns (the cross-product of the row with itself counts the number of events). The association took the same position on issues as the Democratic party (actor 7) ten more times than taking opposite (or no) position. The restaurant association (node 10) took an opposite position to Mr. Bing (node 9) more frequently than supporting (or no) position. Using this algorithm, we've captured much, but not all of the information in the original data. A score of -1, for example, could be the result of two actors taking opposite positions on a single issue; or, it could mean that the two actors both took positions on several issues - and, in sum, they disagreed one more time than they agreed.

The resulting one-mode matrices of actors-by-actors and events-by-events are now valued matrices indicating the strength of the tie based on co-occurrence. Any of the methods for one-mode analysis can now be applied to these matrices to study either micro structure or macro structure.

Two-mode data are sometimes stored in a second way, called the "bipartite" matrix. A bipartite matrix is formed by adding the rows as additional columns, and columns as additional rows. For example, a bipartite matrix of our donors data would have 68 rows (the 23 actors followed by the 45 initiatives) and 68 columns (the 23 actors followed by the 45 initiatives). The two actor-by-event blocks of the matrix are identical to the original matrix; the two new blocks (actors by actors and events by events) are usually coded as zeros. The *Transform>Bipartite* tool converts two-mode rectangular matrices to two-mode bipartite matrices. Figure 17.4 shows a typical dialog.



Input 2-mode dataset: CAMilD

Value to fill within-mode ties: 0.000000

Make result symmetric? No

Output dataset: bi

Figure 17.4: *Dialog of Transform>Bipartite for California political donations data*

The *value to fill within-mode ties* is usually zero, so that actors are connected only by co-presence at events, and events are connected only by having actors in common.

Once data have been put in the form of a square bipartite matrix, many of the algorithms discussed elsewhere in this text for one-mode data can be applied. Considerable caution is needed in interpretation, because the network that is being analyzed is a very unusual one in which the relations are ties between nodes at different levels of analysis. In a sense, actors and events are being treated as social objects at a single level of analysis, and properties like centrality and connection can be explored. This type of analysis is relatively rare, but does have some interesting creative possibilities.

More commonly, we seek to keep the actors and events "separate" but "connected" and to seek patterns in how actors tie events together, and how events tie actors together. We will examine a few techniques for this task later in this chapter. A good first step in any network analysis though is to visualize the data.

17.3: Visualizing Two-Mode Data

There are no new technical issues in using graphs to visualize 2-mode data. Both actors and events are treated as nodes, and lines are used to show the connections of actors to events (there will be no lines from actors to actors directly, or from events to events).

In UCINET, the tool *NetDraw>File>Open>UCINET dataset>2-Mode Network* produces a useful graph for small networks. Figure 17.5 shows one rendering of the California donors data in its valued form.

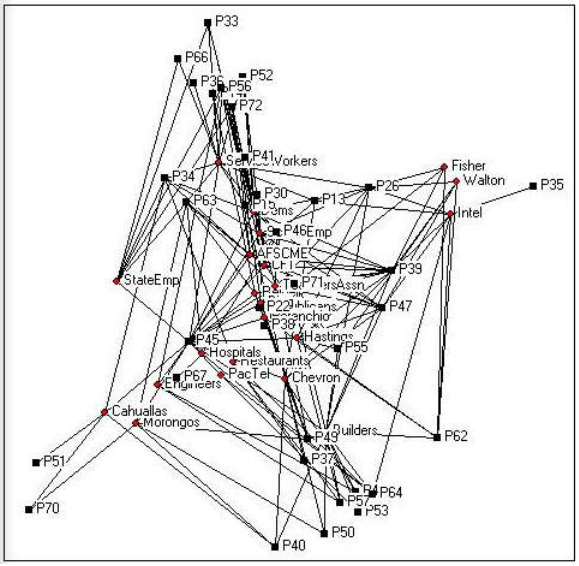


Figure 17.5: *Two-mode valued network of California donors and initiatives.*

Since the graphic has 68 nodes (actors plus initiatives) it is a bit cluttered. We've deleted isolates (initiatives that don't have donors in common and donor that don't have initiatives in common), located the points in space using Gower MDS, resized the nodes and node labels, and eliminated the arrow heads.

We can get some insights from this kind of visualization of a two-mode network (particularly when some kind of scaling method is used to locate the points in space). Actors that are close together (e.g. the Cahualla and Morongo Indians in the lower left corner) are connected because they have similar profiles of events. In this particular case, the two tribes were jointly involved in initiatives about gambling (P70) and environment (P40). Similarly, certain ballot propositions are "similar" in that they have donors in common. And, particular donors are located in the same parts of the space as certain initiatives - defining which issues (events) tend to go along with which actors.

It is exactly this kind of "going together-ness" or "correspondence" of the locations of actors and events that the numeric methods discussed below are intended to index. That is, the numeric methods are efforts to capture the clustering of actors brought together by events; events brought together by the co-presence of actors; and the resulting "bundles" of actors/events.

17.4: Quantitative Analysis

When we are working with a large number of variables that describe aspects of some phenomenon (e.g. items on a test as multiple measures of the underlying trait of "mastery of subject matter"), we often focus our attention on what these multiple measures have "in common". Using information about the co-variation among the multiple measures, we can infer an underlying dimension or factor; once we've done that, we can locate our observations along this dimension. The approach of locating, or scoring, individual cases in terms of their scores on factors of the common variance among multiple indicators is the goal of factor and components analysis (and some other less common scaling techniques).

If we think about our two-mode problem, we could apply this "scaling" logic to either actors or to events. That is, we could "scale" or index the similarity of the actors in terms of their participation in events - but weight the events according to common variance among them. Similarly, we could "scale" the events in terms of the patterns of co-participation of actors - but weight the actors according to their frequency of co-occurrence. Techniques like *Tools>MDS* and factor or principal components analysis could be used to "scale" either actors or events.

It is also possible to apply these kinds of scaling logics to actor-by-event data. UCINET includes two closely-related factor analytic techniques (*Tools>2-Mode Scaling>SVD* and *Tools>2-Mode Scaling Factor Analysis*) that examine the variance in common among both actors and events simultaneously. UCINET also includes *Tools>2-Mode Scaling>Correspondence* which applies the same logic to binary data. Once the underlying dimensions of the joint variance have been identified, we can then "map" both actors and events into the same "space". This allows us to see which actors are similar in terms of their participation in events (that have been weighted to reflect common patterns), which events are similar in terms of what actors participate in them (weighted to reflect common patterns), and which actors and events are located "close" to one another.

It is sometimes possible to interpret they underlying factors or dimensions to gain insights into why actors and events go together in the ways that they do. More generally, clusters of actors and events that are similarly located may form meaningful "types" or "domains" of social action.

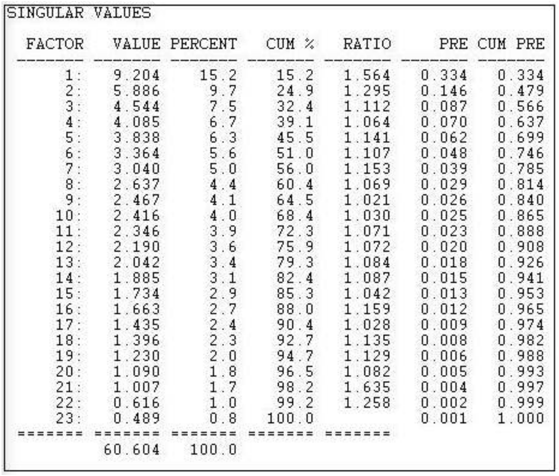
Below, we will very briefly apply these tools to the data on large donors to California initiatives in the 2000-2004 period. Our goal is to illustrate the logic of 2-mode scaling. The discussion here is very short on technical treatments of the (important) differences among the techniques.

Two-Mode SVD Analysis

Singular value decomposition (SVD) is one method of identifying the factors underlying two-mode (valued) data. The method of extracting factors (singular values) differs somewhat from conventional factor and component analysis, so it is a good idea to examine both SVD and 2-mode factoring results.

To illustrate SVD, we have input a matrix of 23 major donors (those who gave a combined total of more than $1,000,000 to five or more campaigns) by 44 California ballot initiatives. Each actor is scored as -1 if they contributed in opposition to the initiative, +1 if they contributed in favor of the initiative, or 0 if they did not contribute. The resulting matrix is valued data that can be examined with SVD and factor analysis; however, the low number of contributors to many initiatives, and the very restricted variance of the scale are not ideal.

Figure 17.6 shows the "singular values" extracted from the rectangular donor-by-initiative matrix using *Tools>2-Mode Scaling>SVD*.

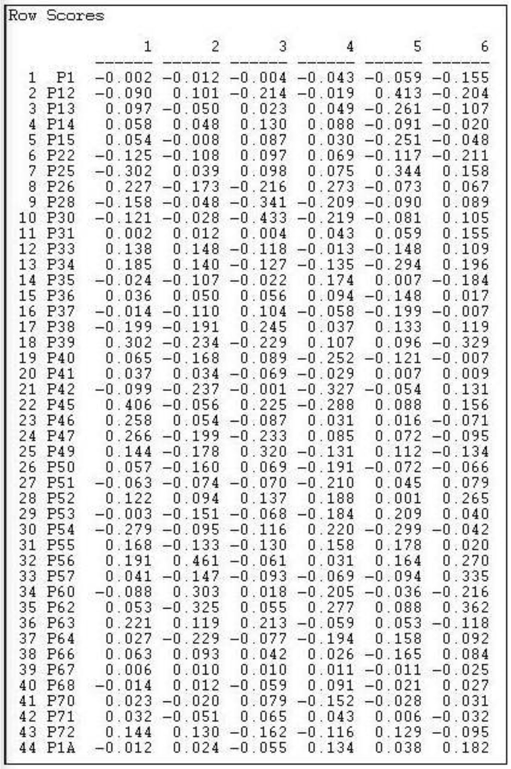


```
SINGULAR VALUES

 FACTOR    VALUE PERCENT   CUM %   RATIO      PRE CUM PRE
------- ------- ------- ------- ------- ------- -------
     1:   9.204    15.2    15.2   1.564   0.334   0.334
     2:   5.886     9.7    24.9   1.295   0.146   0.479
     3:   4.544     7.5    32.4   1.112   0.087   0.566
     4:   4.085     6.7    39.1   1.064   0.070   0.637
     5:   3.838     6.3    45.5   1.141   0.062   0.699
     6:   3.364     5.6    51.0   1.107   0.048   0.746
     7:   3.040     5.0    56.0   1.153   0.039   0.785
     8:   2.637     4.4    60.4   1.069   0.029   0.814
     9:   2.467     4.1    64.5   1.021   0.026   0.840
    10:   2.416     4.0    68.4   1.030   0.025   0.865
    11:   2.346     3.9    72.3   1.071   0.023   0.888
    12:   2.190     3.6    75.9   1.072   0.020   0.908
    13:   2.042     3.4    79.3   1.084   0.018   0.926
    14:   1.885     3.1    82.4   1.087   0.015   0.941
    15:   1.734     2.9    85.3   1.042   0.013   0.953
    16:   1.663     2.7    88.0   1.159   0.012   0.965
    17:   1.435     2.4    90.4   1.028   0.009   0.974
    18:   1.396     2.3    92.7   1.135   0.008   0.982
    19:   1.230     2.0    94.7   1.129   0.006   0.988
    20:   1.090     1.8    96.5   1.082   0.005   0.993
    21:   1.007     1.7    98.2   1.635   0.004   0.997
    22:   0.616     1.0    99.2   1.258   0.002   0.999
    23:   0.489     0.8   100.0           0.001   1.000
======= ======= ======= ======= =======
         60.604   100.0
```

Figure 17.6: *Two-mode scaling of California donors and initiatives by Single Value Decomposition: Singular values*

The "singular values" are analogous to "eigenvalues" in the more common factor and components scaling techniques. The result here shows that the joint "space" of the variance among donors and initiatives is not well captured by a simple characterization. If we could easily make sense of the patterns with ideas like "left/right" and "financial/moral" as underlying dimensions, there would be only a few singular values that explained substantial portions of the joint variance. This result tells us that the ways that actors and events "go together" is not clean, simple, and easy - in this case.

With this important caveat in mind, we can examine how the events and donors are "scaled" or located on the underlying dimensions. First, the ballot initiatives. Figure 17.7 shows the location, or scale scores of each of the ballot propositions on the first six underlying dimensions of this highly multidimensional space.

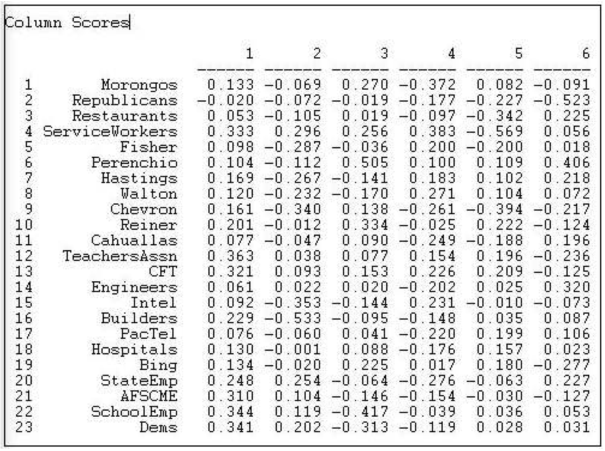


```
Row Scores

                1      2      3      4      5      6
           ------ ------ ------ ------ ------ ------
  1  P1    -0.002 -0.012 -0.004 -0.043 -0.059 -0.155
  2 P12    -0.090  0.101 -0.214 -0.019  0.413 -0.204
  3 P13     0.097 -0.050  0.023  0.049 -0.261 -0.107
  4 P14     0.058  0.048  0.130  0.088 -0.091 -0.020
  5 P15     0.054 -0.008  0.087  0.030 -0.251 -0.048
  6 P22    -0.125 -0.108  0.097  0.069 -0.117 -0.211
  7 P25    -0.302  0.039  0.098  0.075  0.344  0.158
  8 P26     0.227 -0.173 -0.216  0.273 -0.073  0.067
  9 P28    -0.158 -0.048 -0.341 -0.209 -0.090  0.089
 10 P30    -0.121 -0.028 -0.433 -0.219 -0.081  0.105
 11 P31     0.002  0.012  0.004  0.043  0.059  0.155
 12 P33     0.138  0.148 -0.118 -0.013 -0.148  0.109
 13 P34     0.185  0.140 -0.127 -0.135 -0.294  0.196
 14 P35    -0.024 -0.107 -0.022  0.174  0.007 -0.184
 15 P36     0.036  0.050  0.056  0.094 -0.148  0.017
 16 P37    -0.014 -0.110  0.104 -0.058 -0.199 -0.007
 17 P38    -0.199 -0.191  0.245  0.037  0.133  0.119
 18 P39     0.302 -0.234 -0.229  0.107  0.096 -0.329
 19 P40     0.065 -0.168  0.089 -0.252 -0.121 -0.007
 20 P41     0.037  0.034 -0.069 -0.029  0.007  0.009
 21 P42    -0.099 -0.237 -0.001 -0.327 -0.054  0.131
 22 P45     0.406 -0.056  0.225 -0.288  0.088  0.156
 23 P46     0.258  0.054 -0.087  0.031  0.016 -0.071
 24 P47     0.266 -0.199 -0.233  0.085  0.072 -0.095
 25 P49     0.144 -0.178  0.320 -0.131  0.112 -0.134
 26 P50     0.057 -0.160  0.069 -0.191 -0.072 -0.066
 27 P51    -0.063 -0.074 -0.070 -0.210  0.045  0.079
 28 P52     0.122  0.094  0.137  0.188  0.001  0.265
 29 P53    -0.003 -0.151 -0.068 -0.184  0.209  0.040
 30 P54    -0.279 -0.095 -0.116  0.220 -0.299 -0.042
 31 P55     0.168 -0.133 -0.130  0.158  0.178  0.020
 32 P56     0.191  0.461 -0.061  0.031  0.164  0.270
 33 P57     0.041 -0.147 -0.093 -0.069 -0.094  0.335
 34 P60    -0.088  0.303  0.018 -0.205 -0.036 -0.216
 35 P62     0.053 -0.325  0.055  0.277  0.088  0.362
 36 P63     0.221  0.119  0.213 -0.059  0.053 -0.118
 37 P64     0.027 -0.229 -0.077 -0.194  0.158  0.092
 38 P66     0.063  0.093  0.042  0.026 -0.165  0.084
 39 P67     0.006  0.010  0.010  0.011 -0.011 -0.025
 40 P68    -0.014  0.012 -0.059  0.091 -0.021  0.027
 41 P70     0.023 -0.020  0.079 -0.152 -0.028  0.031
 42 P71     0.032 -0.051  0.065  0.043  0.006 -0.032
 43 P72     0.144  0.130 -0.162 -0.116  0.129 -0.095
 44 P1A    -0.012  0.024 -0.055  0.134  0.038  0.182
```

Figure 17.7: *SVD of California donors and initiatives: Scaling of initiatives*

It turns out that the first dimension tends to locate initiatives supporting public expenditure for education and social welfare toward one pole, and initiatives supporting limitation of legislative power toward the other - though interpretations like this are entirely subjective. The second and higher dimensions seem to suggest that initiatives can also be seen as differing from one another in other ways.

At the same time, the results let us locate or scale the donors along the same underlying dimensions. These loadings are shown in Figure 17.8.

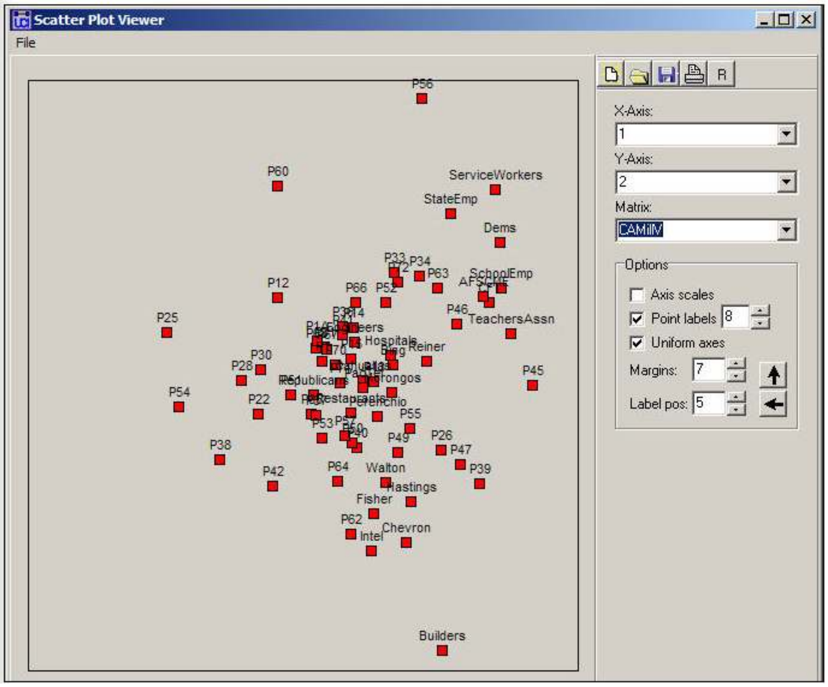


```
Column Scores

                           1      2      3      4      5      6
                      ------ ------ ------ ------ ------ ------
 1          Morongos   0.133 -0.069  0.270 -0.372  0.082 -0.091
 2       Republicans  -0.020 -0.072 -0.019 -0.177 -0.227 -0.523
 3       Restaurants   0.053 -0.105  0.019 -0.097 -0.342  0.225
 4    ServiceWorkers   0.333  0.296  0.256  0.383 -0.569  0.056
 5            Fisher   0.098 -0.287 -0.036  0.200 -0.200  0.018
 6         Perenchio   0.104 -0.112  0.505  0.100  0.109  0.406
 7          Hastings   0.169 -0.267 -0.141  0.183  0.102  0.218
 8            Walton   0.120 -0.232 -0.170  0.271  0.104  0.072
 9           Chevron   0.161 -0.340  0.138 -0.261 -0.394 -0.217
10            Reiner   0.201 -0.012  0.334 -0.025  0.222 -0.124
11         Cahuallas   0.077 -0.047  0.090 -0.249 -0.188  0.196
12      TeachersAssn   0.363  0.038  0.077  0.154  0.196 -0.236
13               CFT   0.321  0.093  0.153  0.226  0.209 -0.125
14         Engineers   0.061  0.022  0.020 -0.202  0.025  0.320
15             Intel   0.092 -0.353 -0.144  0.231 -0.010 -0.073
16          Builders   0.229 -0.533 -0.095 -0.148  0.035  0.087
17            PacTel   0.076 -0.060  0.041 -0.220  0.199  0.106
18         Hospitals   0.130 -0.001  0.088 -0.176  0.157  0.023
19              Bing   0.134 -0.020  0.225  0.017  0.180 -0.277
20          StateEmp   0.248  0.254 -0.064 -0.276 -0.063  0.227
21            AFSCME   0.310  0.104 -0.146 -0.154 -0.030 -0.127
22         SchoolEmp   0.344  0.119 -0.417 -0.039  0.036  0.053
23              Dems   0.341  0.202 -0.313 -0.119  0.028  0.031
```

Figure 17.8: *SVD of California donors and initiatives: Scaling of donors*

Toward the positive end of dimension one (which we earlier interpreted as favoring public expenditure) we find the Democratic party, public employees, and teachers unions; at the opposite pole, we find Republicans and some business and professional groups.

It is often useful to visualize the locations of the actors and events in a scatterplot defined by scale scores on the various dimensions. The map in Figure 17.9 shows the results for the first two dimensions of this space.



X-Axis: 1

Y-Axis: 2

Matrix: CAMilV

□ Axis scales

☑ Point labels 8

☑ Uniform axes

Margins: 7

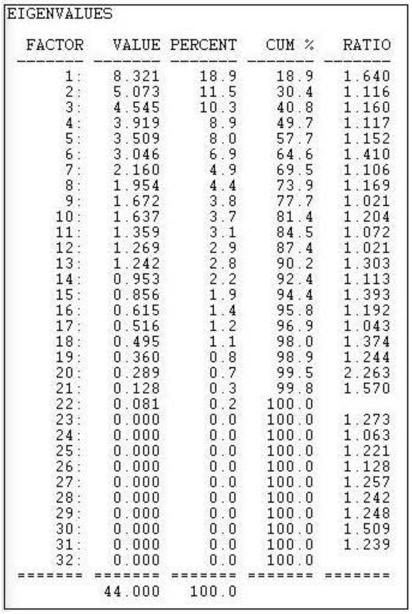
Label pos: 5

Figure 17.9: *SVD of California donors and initiatives: Two-dimensional map*

We note that the first dimension (left-right in the figure) seems to have its poles "anchored" by differences among the initiatives; the second dimension (top-bottom) seems to be defined more by differences among groups (with the exception of proposition 56). The result does not cleanly and clearly locate particular events and particular actors along strong linear dimensions. It does, however, produce some interesting clusters that show groups of actors along with the issues that are central to their patterns of participation. The Democrats and unions cluster (upper right) along with a number of particular propositions in which they were highly active (e.g. 46, 63). Corporate, building, and venture capitalist cluster (more loosely) in the lower right, along with core issues that formed their primary agenda in the initiative process (e.g. prop 62).

Two-Mode Factor Analysis

Factor analysis provides an alternative method to SVD to accomplish the same goals: identifying underlying dimensions of the joint space of actor-by-event variance, and locating or scaling actors and events in that space. The method used by factor analysis to identify the dimensions differs from SVD. Figure 17.10 shows the eigenvalues (by principle components) calculated by *Tools>2-Mode Scaling>Factor Analysis*.

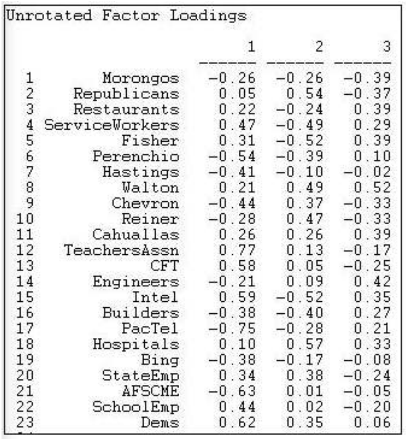


```
EIGENVALUES

 FACTOR    VALUE PERCENT   CUM %   RATIO
 ------- ------- ------- ------- -------
      1:   8.321    18.9    18.9   1.640
      2:   5.073    11.5    30.4   1.116
      3:   4.545    10.3    40.8   1.160
      4:   3.919     8.9    49.7   1.117
      5:   3.509     8.0    57.7   1.152
      6:   3.046     6.9    64.6   1.410
      7:   2.160     4.9    69.5   1.106
      8:   1.954     4.4    73.9   1.169
      9:   1.672     3.8    77.7   1.021
     10:   1.637     3.7    81.4   1.204
     11:   1.359     3.1    84.5   1.072
     12:   1.269     2.9    87.4   1.021
     13:   1.242     2.8    90.2   1.303
     14:   0.953     2.2    92.4   1.113
     15:   0.856     1.9    94.4   1.393
     16:   0.615     1.4    95.8   1.192
     17:   0.516     1.2    96.9   1.043
     18:   0.495     1.1    98.0   1.374
     19:   0.360     0.8    98.9   1.244
     20:   0.289     0.7    99.5   2.263
     21:   0.128     0.3    99.8   1.570
     22:   0.081     0.2   100.0
     23:   0.000     0.0   100.0   1.273
     24:   0.000     0.0   100.0   1.063
     25:   0.000     0.0   100.0   1.221
     26:   0.000     0.0   100.0   1.128
     27:   0.000     0.0   100.0   1.257
     28:   0.000     0.0   100.0   1.242
     29:   0.000     0.0   100.0   1.248
     30:   0.000     0.0   100.0   1.509
     31:   0.000     0.0   100.0   1.239
     32:   0.000     0.0   100.0
 ======= ======= ======= ======= =======
          44.000   100.0
```

Figure 17.10: *Eigenvalues of two-mode factoring of California donors and initiatives*

This solution, although different from SVD, also suggests considerable dimensional complexity in the joint variance of actors and events. That is, simple characterizations of the underlying dimensions (e.g. "left/right") do not provide very accurate predictions about the locations of individual actors or events. The factor analysis method does produce somewhat lower complexity than SVD.

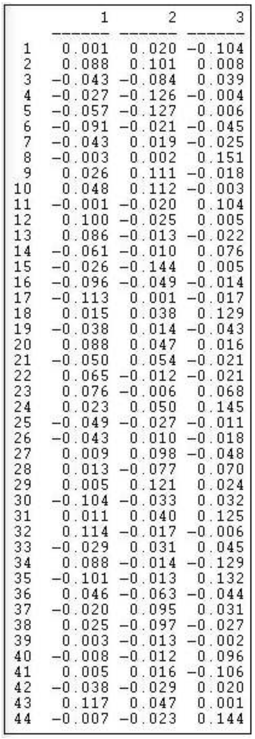
With the caveat of pretty poor fit of a low-dimensional solution in mind, let's examine the scaling of actors on the first three factors (Figure 17.11)



```
Unrotated Factor Loadings

                            1      2      3
                       ------ ------ ------
 1        Morongos      -0.26  -0.26  -0.39
 2     Republicans       0.05   0.54  -0.37
 3     Restaurants       0.22  -0.24   0.39
 4 ServiceWorkers        0.47  -0.49   0.29
 5          Fisher       0.31  -0.52   0.39
 6      Perenchio       -0.54  -0.39   0.10
 7       Hastings       -0.41  -0.10  -0.02
 8         Walton        0.21   0.49   0.52
 9        Chevron       -0.44   0.37  -0.33
10         Reiner       -0.28   0.47  -0.33
11      Cahuallas        0.26   0.26   0.39
12   TeachersAssn        0.77   0.13  -0.17
13            CFT        0.58   0.05  -0.25
14      Engineers       -0.21   0.09   0.42
15          Intel        0.59  -0.52   0.35
16       Builders       -0.38  -0.40   0.27
17         PacTel       -0.75  -0.28   0.21
18      Hospitals        0.10   0.57   0.33
19           Bing       -0.38  -0.17  -0.08
20       StateEmp        0.34   0.38  -0.24
21         AFSCME       -0.63   0.01  -0.05
22      SchoolEmp        0.44   0.02  -0.20
23           Dems        0.62   0.35   0.06
```

Figure 17.11: *Loadings of donors*

The first factor, by this method, produces a similar pattern to SVD. At one pole are Democrats and unions, at the other lie many capitalist groups. There are, however, some notable differences (e.g. AFSCME). Figure 17.12 shows the loadings of the events.



```
             1      2      3
        ------ ------ ------
  1      0.001  0.020 -0.104
  2      0.088  0.101  0.008
  3     -0.043 -0.084  0.039
  4     -0.027 -0.126 -0.004
  5     -0.057 -0.127  0.006
  6     -0.091 -0.021 -0.045
  7     -0.043  0.019 -0.025
  8     -0.003  0.002  0.151
  9      0.026  0.111 -0.018
 10      0.048  0.112 -0.003
 11     -0.001 -0.020  0.104
 12      0.100 -0.025  0.005
 13      0.086 -0.013 -0.022
 14     -0.061 -0.010  0.076
 15     -0.026 -0.144  0.005
 16     -0.096 -0.049 -0.014
 17     -0.113  0.001 -0.017
 18      0.015  0.038  0.129
 19     -0.038  0.014 -0.043
 20      0.088  0.047  0.016
 21     -0.050  0.054 -0.021
 22      0.065 -0.012 -0.021
 23      0.076 -0.006  0.068
 24      0.023  0.050  0.145
 25     -0.049 -0.027 -0.011
 26     -0.043  0.010 -0.018
 27      0.009  0.098 -0.048
 28      0.013 -0.077  0.070
 29      0.005  0.121  0.024
 30     -0.104 -0.033  0.032
 31      0.011  0.040  0.125
 32      0.114 -0.017 -0.006
 33     -0.029  0.031  0.045
 34      0.088 -0.014 -0.129
 35     -0.101 -0.013  0.132
 36      0.046 -0.063 -0.044
 37     -0.020  0.095  0.031
 38      0.025 -0.097 -0.027
 39      0.003 -0.013 -0.002
 40     -0.008 -0.012  0.096
 41      0.005  0.016 -0.106
 42     -0.038 -0.029  0.020
 43      0.117  0.047  0.001
 44     -0.007 -0.023  0.144
```

Figure 17.12: *Loadings of events*

The patterns here also have some similarity to the SVD results, but do differ considerably in the specifics. To visualize the patterns, the loadings of actors and events on the dimensions could be extracted from output data files, and graphed using a scatterplot.

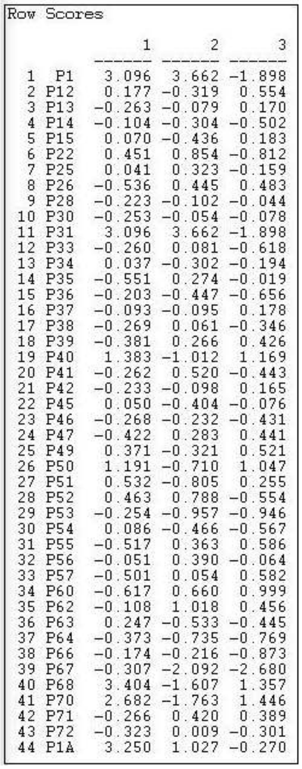
Two-Mode Correspondence Analysis

For binary data, the use of factor analysis and SVD is not recommended. Factoring methods operate on the variance/covariance or correlation matrices among actors and events. When the connections of actors to events is measured at the binary level (which is very often the case in network analysis) correlations may seriously understate covariance and make patterns difficult to discern.

As an alternative for binary actor-by-event scaling, the method of correspondence analysis (*Tools>2-Mode Scaling>Correspondence*) can be used. Correspondence analysis (rather like Latent Class Analysis) operates on multivariate binary cross-tabulations, and its distributional assumptions are better suited to binary data.

To illustrate the application of correspondence analysis, we've dichotomized the political donor and initiatives data by assigning a value of 1 if an actor gave a donation either in favor or against an initiative, and assigning a zero if they did not participate in the campaign of a particular initiative. If we wanted our analysis to pay attention to partisanship, rather than simple participation, we could have created two datasets - one based on opposition or not, one based on support or not - and done two separate correspondence analyses.

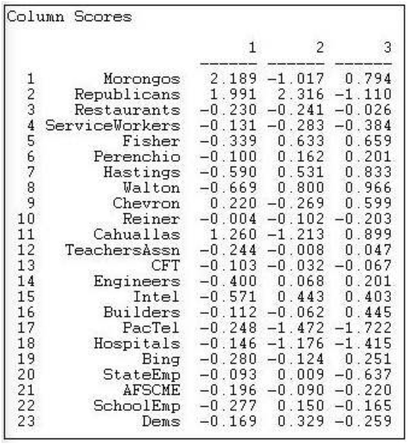
Figure 17.13 shows the location of events (initiatives) along three dimensions of the joint actor-event space identified by the correspondence analysis method.



```
Row Scores

                1      2      3
           ------ ------ ------
  1  P1     3.096  3.662 -1.898
  2 P12     0.177 -0.319  0.554
  3 P13    -0.263 -0.079  0.170
  4 P14    -0.104 -0.304 -0.502
  5 P15     0.070 -0.436  0.183
  6 P22     0.451  0.854 -0.812
  7 P25     0.041  0.323 -0.159
  8 P26    -0.536  0.445  0.483
  9 P28    -0.223 -0.102 -0.044
 10 P30    -0.253 -0.054 -0.078
 11 P31     3.096  3.662 -1.898
 12 P33    -0.260  0.081 -0.618
 13 P34     0.037 -0.302 -0.194
 14 P35    -0.551  0.274 -0.019
 15 P36    -0.203 -0.447 -0.656
 16 P37    -0.093 -0.095  0.178
 17 P38    -0.269  0.061 -0.346
 18 P39    -0.381  0.266  0.426
 19 P40     1.383 -1.012  1.169
 20 P41    -0.262  0.520 -0.443
 21 P42    -0.233 -0.098  0.165
 22 P45     0.050 -0.404 -0.076
 23 P46    -0.268 -0.232 -0.431
 24 P47    -0.422  0.283  0.441
 25 P49     0.371 -0.321  0.521
 26 P50     1.191 -0.710  1.047
 27 P51     0.532 -0.805  0.255
 28 P52     0.463  0.788 -0.554
 29 P53    -0.254 -0.957 -0.946
 30 P54     0.086 -0.466 -0.567
 31 P55    -0.517  0.363  0.586
 32 P56    -0.051  0.390 -0.064
 33 P57    -0.501  0.054  0.582
 34 P60    -0.617  0.660  0.999
 35 P62    -0.108  1.018  0.456
 36 P63     0.247 -0.533 -0.445
 37 P64    -0.373 -0.735 -0.769
 38 P66    -0.174 -0.216 -0.873
 39 P67    -0.307 -2.092 -2.680
 40 P68     3.404 -1.607  1.357
 41 P70     2.682 -1.763  1.446
 42 P71    -0.266  0.420  0.389
 43 P72    -0.323  0.009 -0.301
 44 P1A     3.250  1.027 -0.270
```

Figure 17.13: *Event coordinates for co-participation of donors in California initiative campaigns*

Since these data do not reflect partisanship, only participation, we would not expect the findings to parallel those discussed in the sections above. And, they don't. We do see, however, that this method also can be used to locate the initiatives along multiple underlying dimensions that capture variance in both actors and events. Figure 17.14 shows the scaling of the actors.



```
Column Scores

                             1      2      3
                        ------ ------ ------
 1          Morongos     2.189 -1.017  0.794
 2       Republicans     1.991  2.316 -1.110
 3       Restaurants    -0.230 -0.241 -0.026
 4 ServiceWorkers       -0.131 -0.283 -0.384
 5            Fisher    -0.339  0.633  0.659
 6        Perenchio     -0.100  0.162  0.201
 7          Hastings    -0.590  0.531  0.833
 8            Walton    -0.669  0.800  0.966
 9           Chevron     0.220 -0.269  0.599
10            Reiner    -0.004 -0.102 -0.203
11         Cahuallas     1.260 -1.213  0.899
12      TeachersAssn    -0.244 -0.008  0.047
13               CFT    -0.103 -0.032 -0.067
14         Engineers    -0.400  0.068  0.201
15             Intel    -0.571  0.443  0.403
16          Builders    -0.112 -0.062  0.445
17            PacTel    -0.248 -1.472 -1.722
18         Hospitals    -0.146 -1.176 -1.415
19              Bing    -0.280 -0.124  0.251
20          StateEmp    -0.093  0.009 -0.637
21            AFSCME    -0.196 -0.090 -0.220
22         SchoolEmp    -0.277  0.150 -0.165
23              Dems    -0.169  0.329 -0.259
```

Figure 17.14: *Actor coordinates for co-participation of donors in California initiative campaigns*

The first dimension here does have some similarity to the Democrat/union versus capitalist poles. Here, however, this difference means that the two groupings tend to participate in different groups of initiatives, rather than confronting one another in the same campaigns.

Visualization is often the best approach to finding meaningful patterns (in the absence of a strong theory). Figure 17.15 shows the plot of the actors and events in the first two dimensions of the joint correspondence analysis space.

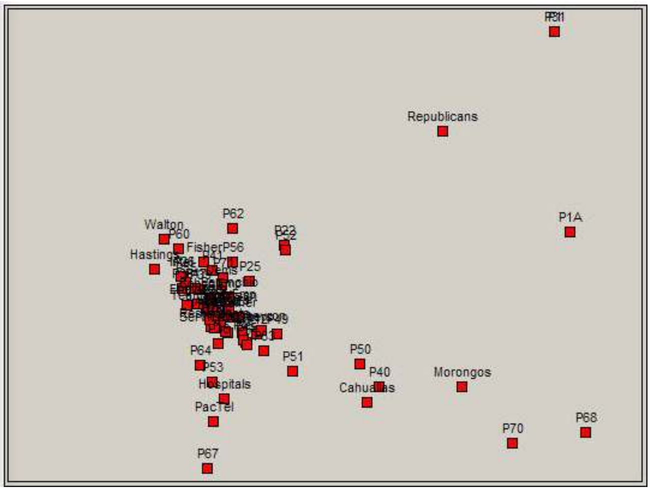


Figure 17.15: *Correspondence analysis two-dimensional map*

The lower right quadrant here contains a meaningful cluster of actors and events, and illustrates how the results of correspondence analysis can be interpreted. In the lower right we have some propositions regarding Indian casino gambling (68 and 70) and two propositions regarding ecological/conservation issues (40 and 50). Two of the major Native American Nations (the Cahualla and Morongo bands of Mission Indians) are mapped together. The result is showing that there is a cluster of issues that "co-occur" with a cluster of donors - actors defining events, and events defining actors.

17.5: Qualitative Analysis

Often all that we know about actors and events is simple co-presence. That is, either an actor was, or wasn't, present, and our incidence matrix is binary. In cases like this, the scaling methods in the last section can be applied, but one should be very cautious about the results. This is because the various dimensional methods operate on similarity/distance matrices, and measures like correlations (as used in two-mode factor analysis) can be misleading with binary data. Even correspondence analysis, which is more friendly to binary data, can be troublesome when data are sparse.

An alternative approach is block modeling. Block modeling works directly on the binary incidence matrix by trying to permute rows and columns to fit, as closely as possible, idealized images. This approach doesn't involve any of the distributional assumptions that are made in scaling analysis.

In principle, one could fit any sort of block model to actor-by-event incidence data. We will examine two models that ask meaningful (alternative) questions about the patterns of linkage between actors and events. Both of these models can be directly calculated in UCINET. Alternative block models, of course, could be fit to incidence data using more general block-modeling algorithms.

Two-Mode Core-Periphery Analysis

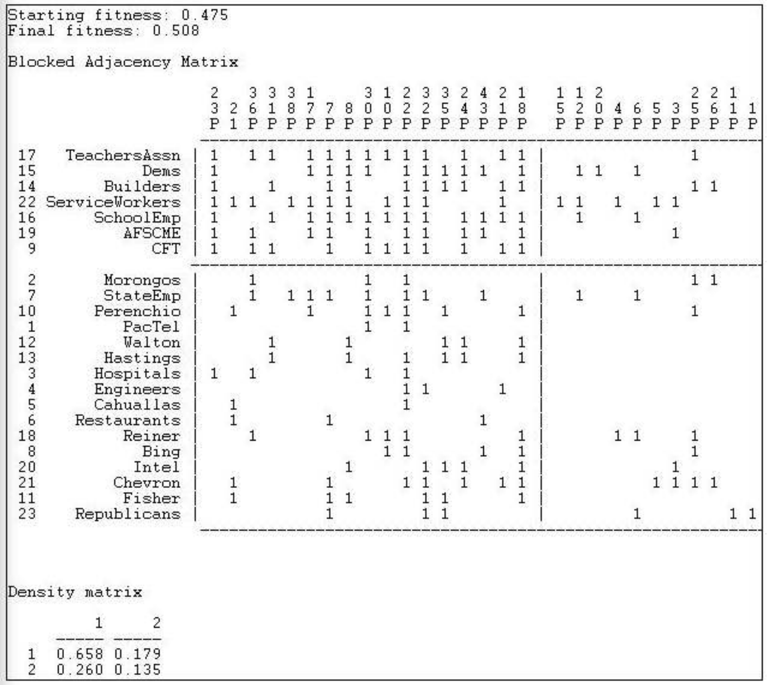
The core-periphery structure is an ideal typical pattern that divides both the rows and the columns into two classes. One of the blocks on the main diagonal (the core) is a high-density block; the other block on the main diagonal (the periphery) is a low-density block. The core-periphery model is indifferent to the density of ties in the off-diagonal blocks.

When we apply the core-periphery model to actor-by-actor data (see *Network>Core/Periphery*), the model seeks to identify a set of actors who have high density of ties among themselves (the core) by sharing many events in common, and another set of actors who have very low density of ties among themselves (the periphery) by having few events in common. Actors in the core are able to coordinate their actions, those in the periphery are not. As a consequence, actors in the core are at a structural advantage in exchange relations with actors in the periphery.

When we apply the core-periphery model to actor-by-event data (*Network>2-Mode>Categorical Core/Periphery*) we are seeking the same idealized "image" of a high and a low density block along the main diagonal. But, now the meaning is rather different.

The "core" consists of a partition of actors that are closely connected to each of the events in an event partition; and simultaneously a partition of events that are closely connected to the actors in the core partition. So, the "core" is a cluster of frequently co-occurring actors and events. The "periphery" consists of a partition of actors who are not co-incident to the same events; and a partition of events that are disjoint because they have no actors in common.

Network>2-Mode>Categorical Core/Periphery uses numerical methods to search for the partition of actors and of events that comes as close as possible to the idealized image. Figure 17.16 shows a portion of the results of applying this method to participation (not partisanship) in the California donors and initiatives data.



```
Starting fitness: 0.475
Final fitness: 0.508

Blocked Adjacency Matrix

                     2   3 3 3 1     3 1 2 3 3 2 4 2 1   1 1 2         2 2 1
                     3 2 6 1 8 7 7 8 0 0 2 2 5 4 3 1 8   5 2 0 4 6 5 3 5 6 1 1
                     P 1 P P P P P P P P P P P P P P P   P P P P P P P P P P P
                    ----------------------------------------------------------
 17   TeachersAssn | 1   1 1   1 1 1 1 1 1 1   1   1 1 |               1
 15           Dems | 1         1 1 1 1   1 1 1 1 1   1 |   1 1   1
 14       Builders | 1     1     1 1     1 1 1 1   1 1 |               1 1
 22 ServiceWorkers | 1 1 1   1 1 1 1   1 1 1       1   | 1 1   1   1 1
 16      SchoolEmp | 1     1   1 1 1 1 1 1 1   1 1 1 1 |   1     1
 19         AFSCME | 1   1     1 1   1   1 1   1 1   1 |             1
  9            CFT | 1   1 1     1   1 1 1 1   1   1 1 |
                   ------------------------------------------------------------
  2       Morongos |     1           1   1             |               1 1
  7       StateEmp |     1   1 1 1   1   1 1     1     |   1     1
 10      Perenchio |   1       1     1 1 1   1       1 |               1
  1         PacTel |                 1   1             |
 12         Walton |       1       1         1 1     1 |
 13       Hastings |       1       1     1   1 1     1 |
  3      Hospitals | 1   1           1   1             |
  4      Engineers |                     1 1       1   |
  5      Cahuallas |   1                 1             |
  6    Restaurants |   1         1               1     |
 18         Reiner |     1           1 1 1           1 |       1 1     1
  8           Bing |                   1 1       1   1 |               1
 20          Intel |               1       1 1 1     1 |             1
 21        Chevron |   1         1       1 1   1   1 1 |           1 1 1 1
 11         Fisher |   1         1 1       1 1       1 |
 23    Republicans |             1         1 1         |         1         1 1
                    ----------------------------------------------------------


Density matrix

          1      2
       ------ ------
  1    0.658 0.179
  2    0.260 0.135
```

Figure 17.16: *Categorical core-periphery model of California $1M donors and ballot initiatives (truncated)*

The numerical search method used by *Network>2-Mode>Categorical Core/Periphery* is a genetic algorithm, and the measure of goodness of fit is stated in terms of a "fitness" score (0 means bad fit, 1 means excellent fit). You can also judge the goodness of the result by examining the density matrix at the end of the output. If the block model was completely successful, the 1,1 block should have a density of one, and the 2,2 block should have a density of zero. While far from perfect, the model here is good enough to be taken seriously.

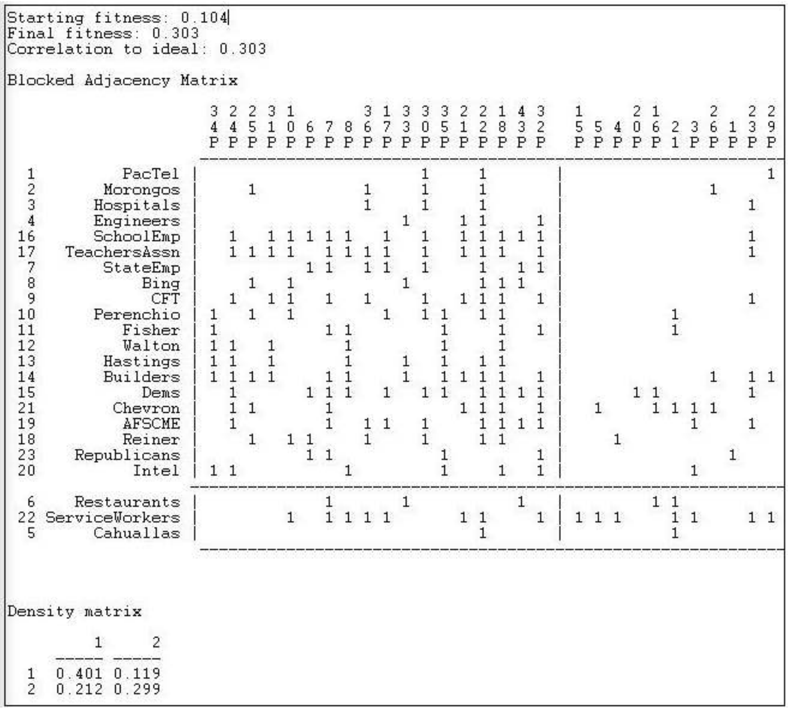
The blocked matrix shows a "core" composed of the Democratic Party, a number of major unions, and the building industry association who are all very likely to participate in a considerable number of initiatives (proposition 23 through proposition 18). The remainder of the actors are grouped into the periphery as both participating less frequently, and having few issues in common. A considerable number of issues are also grouped as "peripheral" in the sense that they attract few donors, and these donors have little in common. We also see (upper right) that core actors do participate to some degree (0.179) in peripheral issues. In the lower left, we see that peripheral actors participate somewhat more heavily (0.260) in core issues.

Two-Mode Factions Analysis

An alternative block model is that of "factions". Factions are groupings that have high density within the group, and low density of ties between groups. *Network>Subgroups>Factions* fits this block model to one-mode data (for any user-specified number of factions). *Network>2-Mode>2-Mode Factions* fits the same type of model to two-mode data (but for only two factions).

When we apply the factions model to one-mode actor data, we are trying to identify two clusters of actors who are closely tied to one another by attending all of the same events, but very loosely connected to members of other factions and the events that tie them together. If we were to apply the idea of factions to events in a one-mode analysis, we would be seeking to identify events that were closely tied by having exactly the same participants.

Network>2-Mode>2-Mode Factions applies the same approach to the rectangular actor-by-event matrix. In doing this,we are trying to locate joint groupings of actors and events that are as mutually exclusive as possible. In principle, there could be more than two such factions. Figure 17.17 shows the results of the two-mode factions block model to the participation of top donors in political initiatives.



```
Starting fitness: 0.104
Final fitness: 0.303
Correlation to ideal: 0.303

Blocked Adjacency Matrix

                     3 2 2 3 1       3 1 3 3 3 2 2 1 4 3   1     2 1     2   2 2
                     4 4 5 1 0 6 7 8 6 7 3 0 5 1 2 8 3 2   5 5 4 0 6 2 3 6 1 3 9
                     P P P P P P P P P P P P P P P P P P   P P P P P 1 P P P P P
                    -------------------------------------------------------------
  1         PacTel |                       1     1       |                     1
  2       Morongos |     1           1     1     1       |               1
  3      Hospitals |                 1     1     1       |                   1
  4      Engineers |                     1     1 1     1 |
 16      SchoolEmp |   1   1 1 1 1 1   1   1   1 1 1 1 1 |                   1
 17   TeachersAssn |   1 1 1 1   1 1 1 1   1   1 1 1   1 |                   1
  7       StateEmp |           1 1   1 1   1     1   1 1 |
  8           Bing |     1   1           1       1 1 1   |
  9            CFT |   1   1 1   1   1     1   1 1 1   1 |                   1
 10      Perenchio | 1   1   1         1   1 1   1 1     |           1
 11         Fisher | 1           1 1         1     1   1 |           1
 12         Walton | 1 1   1       1         1     1     |
 13       Hastings | 1 1   1       1     1   1   1 1     |
 14       Builders | 1 1 1 1     1 1     1   1 1 1 1   1 |               1   1 1
 15           Dems |   1       1 1 1   1   1 1   1 1 1 1 |       1 1         1
 21        Chevron |   1 1       1             1 1 1   1 |   1     1 1 1 1
 19         AFSCME |   1         1   1 1   1     1 1 1 1 |             1     1
 18         Reiner |     1   1 1     1     1     1 1     |     1
 23    Republicans |           1 1           1         1 |                 1
 20          Intel | 1 1           1         1     1   1 |             1
                   --------------------------------------------------------------
  6    Restaurants |             1       1           1   |         1 1
 22 ServiceWorkers |         1   1 1 1 1       1 1     1 | 1 1 1     1 1     1 1
  5      Cahuallas |                             1       |           1
                    -------------------------------------------------------------


Density matrix

         1     2
     ----- -----
  1  0.401 0.119
  2  0.212 0.299
```

Figure 17.17: *Two-mode factions model of California $1M donors and ballot initiatives (truncated)*

Two measures of goodness-of-fit are available. First we have our "fitness" score, which is the correlation between the observed scores (0 or 1) and the scores that "should" be present in each block. The densities in the blocks also informs us about goodness of fit. For a factions analysis, an ideal pattern would be dense 1-blocks along the diagonal (many ties within groups) and zero-blocks off the diagonal (ties between groups).

The fit of the two factions model is not as impressive as the fit of the core-periphery model. This suggests that an "image" of California politics as one of two separate and largely disjoint issue-actor spaces is not as useful as an image of a high intensity core of actors and issues coupled with an otherwise disjoint set of issues and participants.

The blocking itself also is not very appealing, placing most of the actors in one faction (with modest density of 0.401). The second faction is small, and has a density (0.299) that is not very different from the off-diagonal blocks. As before, the blocking of actors by events is grouping together sets of actors and events that define one another.

17.S: Two-Mode Networks (Summary)

One of the major continuing themes of social network analysis is the way in which individual actors "make" larger social structures by their patterns of interaction while, at the same time, institutional patterns shape the choices made by the individuals who are embedded within structures.

Two-mode data (often referred to as "actor-by-event" or "affiliation" in social network analysis) offer some interesting possibilities for gaining insights into macro-micro or agent-structure relations. With two-mode data, we can examine how macro-structures (events) pattern the interactions among agents (or not); we can also examine how the actors define and create macro structures by their patterns of affiliation with them. In addition, we can attempt to describe patterns of relations between actors and structures simultaneously.

In this chapter we briefly examined some of the typical ways in which two-mode data arise in social network analysis, and the data structures that are used to record and manipulate two-mode data. We also briefly examined the utility of two-mode graphs (bipartite graphs) in visualizing the "social space" defined by both actors and events.

Our primary attention, though, was on methods for trying to identify patterns in two-mode data that might better help us describe and understand why actors and events "fit together" in the ways they do.

One class of methods derives from factor analysis and related approaches. These methods (best applied to valued data) seek to identify underlying "dimensions" of the actor-event space, and then map both actors and events in this space. These approaches can be particularly helpful in seeking the "hidden logic" or "latent structure" of more abstract dimensions that may underlie the interactions of many specific actors across many specific events. They can also be useful to identify groups of actors and the events that "go together" when viewed through the lens of latent abstract dimensions.

Another class of methods is based on block modeling. The goal of these methods is to assess how well the observed patterns of actor-event affiliations fit some prior notions of the nature of the "joint space" (i.e. "core-periphery" or "factions"). To the extent that the actor-event affiliations can be usefully thought of in these ways, block models also then allow us to classify types or groups of actors along with the events that are characteristic of them.

Two-mode analysis of social networks need not be limited to individual persons and their participation in voluntary activities (as in the cases of our examples, and the original Davis study discussed at the beginning of this chapter). The tools of two-mode analysis could be applied to CSS (cognitive social structure) data to see if perceivers can be classified according to similarity in their perceptions of networks, simultaneously with classifying network images in terms of the similarity of those doing the perceiving. Units at any level of analysis (organizations and industries, nation states and civilizations, etc.) might be usefully viewed as two-mode problems.

CHAPTER OVERVIEW

18: Some Statistical Tools

In this chapter we will look at some of the most basic and common approaches to applying statistical analysis to the attributes of actors embedded in networks, the relations among these actors, and the similarities between multiple relational networks connecting the same actors.

18.1: Introduction to Statistical Tools

Network analysis in the social sciences developed from a conjuncture of anthropologist's observations about relations in face-to-face groups and mathematical graph theory. A very large part of social network methodology, consequently, deals with relatively small networks, networks where we have confidence in the reliability of our observations about the relations among the actors. Most of the tools of social network analysis involve the use of mathematical functions to describe networks and their sub-structures.

In more recent work, however, some of the focus of social network research has moved away from these roots. Increasingly, the social networks that are being studied may contain many nodes; and, sometimes our observations about these very large networks are based not on censuses, but on samples of nodes. Network researchers have come to recognize that the relations that they study may be constantly evolving, and that the relations observed at one point in time may not the entirely typical because the pattern of relations is not "in equilibrium." They have also recognized that sometimes our observations are fallible -- we fail to record a relation that actually exists, or mis-measure the strength of a tie.

All of these concerns (large networks, sampling, concern about the reliability of observations) have led social network researchers to begin to apply the techniques of descriptive and inferential statistics in their work. Statistics provide useful tools for summarizing large amounts of information, and for treating observations as stochastic, rather than deterministic outcomes of social processes.

Descriptive statistics have proven to be of great value because they provide convenient tools to summarize key facts about the distributions of actors, attributes, and relations; statistical tools can describe not only the shape of one distribution, but also joint distributions, or "statistical association." So, statistical tools have been particularly helpful in describing, predicting, and testing hypotheses about the relations between network properties.

Inferential statistics have also proven to have very useful applications to social network analysis. At a most general level, the question of "inference" is: how much confidence can I have that the pattern I see in the data I've collected is actually typical of some larger population, or that the apparent pattern is not really just a random occurrence?

In this chapter we will look at some of the ways in which quite basic statistical tools have been applied in social network analysis. These are only the starting point. The development of more powerful statistical tools especially tuned for the needs of social network analysis is one of the most rapidly developing "cutting edges" of the field.

18.2: Describing One Network

Most social scientists have a reasonable working knowledge of basic univariate and bivariate descriptive and inferential statistics. Many of these tools find immediate application in working with social network data. There are, however, two quite important distinctive features of applying these tools to network data.

First, and most important, **social network analysis is about relations among actors**, not about relations between variables. Most social scientists have learned their statistics with applications to the study of the distribution of the scores of actors (cases) on variables, and the relations between these distributions. We learn about the mean of a set of scores on the variable "income." We learn about the Pearson zero-order product moment correlation coefficient for indexing linear association between the distribution of actor's incomes and actor's educational attainment.

The application of statistics to social networks is also about describing distributions and relations among distributions. But, rather than describing distributions of attributes of actors (or "variables"), we are concerned with describing the distributions of relations among actors. In applying statistics to network data, we are concerned the issues like the average strength of the relations between actors; we are concerned with questions like "is the strength of ties between actors in a network correlated with the centrality of the actors in the network?" Most of the descriptive statistical tools are the same for attribute analysis and for relational analysis -- but the subject matter is quite different!

Second, many of tools of standard inferential statistics that we learned from the study of the distributions of attributes do not apply directly to network data. Most of the standard formulas for calculating estimated standard errors, computing test statistics, and assessing the probability of null hypotheses that we learned in basic statistics don't work with network data (and, if used, can give us "false positive" answers more often than "false negative"). This is because the "observations" or scores in network data are not "independent" samplings from populations. In attribute analysis, it is often very reasonable to assume that Fred's income and Fred's education are a "trial" that is independent of Sue's income and Sue's education. We can treat Fred and Sue as independent replications.

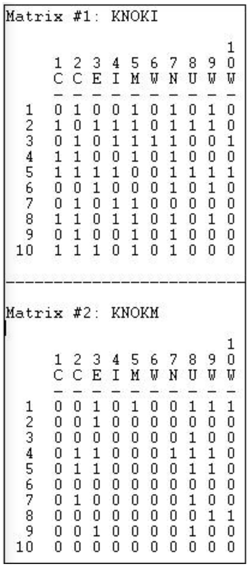
In network analysis, we focus on relations, not attributes. So, one observation might well be Fred's tie with Sue; another observation might be Fred's tie with George; still another might be Sue's tie with George. These are not "independent" replications. Fred is involved in two observations (as are Sue an George), it is probably not reasonable to suppose that these relations are "independent" because they both involve George.

The standard formulas for computing standard errors and inferential tests on attributes generally assume independent observations. Applying them when the observations are not independent can be very misleading. Instead, alternative numerical approaches to estimating standard errors for network statistics are used. These "boot-strapping" (and permutations) approaches calculate sampling distributions of statistics directly from the observed networks by using random assignment across hundreds or thousands of trials under the assumption that null hypotheses are true.

These general points will become clearer as we examine some real cases. So, let's begin with the simplest univariate descriptive and inferential statistics, and then move on to somewhat more complicated problems.

Univariate Descriptive Statistics

For most of the examples in this chapter, we'll focus again on the Knoke data set that describes the two relations of the exchange of information and the exchange of money among ten organizations operating in the social welfare field. Figure 18.1 lists these data.



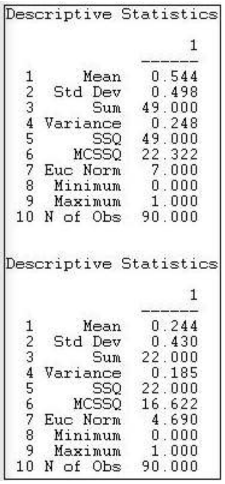
```
Matrix #1: KNOKI

                                  1
       1  2  3  4  5  6  7  8  9  0
       C  C  E  I  M  W  N  U  W  W
       -  -  -  -  -  -  -  -  -  -
   1   0  1  0  0  1  0  1  0  1  0
   2   1  0  1  1  1  0  1  1  1  0
   3   0  1  0  1  1  1  1  0  0  1
   4   1  1  0  0  1  0  1  0  0  0
   5   1  1  1  1  0  0  1  1  1  1
   6   0  0  1  0  0  0  1  0  1  0
   7   0  1  0  1  1  0  0  0  0  0
   8   1  1  0  1  1  0  1  0  1  0
   9   0  1  0  0  1  0  1  0  0  0
  10   1  1  1  0  1  0  1  0  0  0

-----------------------------------

Matrix #2: KNOKM

                                  1
       1  2  3  4  5  6  7  8  9  0
       C  C  E  I  M  W  N  U  W  W
       -  -  -  -  -  -  -  -  -  -
   1   0  0  1  0  1  0  0  1  1  1
   2   0  0  1  0  0  0  0  0  0  0
   3   0  0  0  0  0  0  0  1  0  0
   4   0  1  1  0  0  0  1  1  1  0
   5   0  1  1  0  0  0  0  1  1  0
   6   0  0  0  0  0  0  0  0  0  0
   7   0  1  0  0  0  0  0  1  0  0
   8   0  0  0  0  0  0  0  0  1  1
   9   0  0  1  0  0  0  0  1  0  0
  10   0  0  0  0  0  0  0  0  0  0
```

Figure 18.1: *Listing (Data>Display) of Knoke information and money exchange matrices*

These particular data happen to be asymmetric and binary. Most of the statistical tools for working with network data can be applied to symmetric data, and data where the relations are valued (strength, cost, probability of a tie). As with any descriptive statistics, the scale of measurement (binary or valued) does matter in making proper choices about interpretation and application of many statistical tools.

The data that are analyzed with statistical tools when we are working with network data are the observations about relations among actors. So, in each matrix, we have 10 x 10 = 100 observations or cases. For many analyses, the ties of actors with themselves (the main diagonal) are not meaningful, and are not used, so there would be $(N*N-1=90)$ observations. If data are symmetric (i.e. $X_{ij}=X_{ji}$), half of these are redundant, and wouldn't be used, so there would be $(N*N-1/2=45)$ observations.

What we would like to summarize with our descriptive statistics are some characteristics of the distribution of these scores. *Tools>Univariate Stats* can be used to generate the most commonly used measures for each matrix (select *matrix* in the dialog, and chose whether or not to *include the diagonal*). Figure 18.2 shows the results for our example data, excluding the diagonal.



```
Descriptive Statistics

                     1
                ------
 1      Mean     0.544
 2   Std Dev     0.498
 3       Sum    49.000
 4  Variance     0.248
 5       SSQ    49.000
 6     MCSSQ    22.322
 7  Euc Norm     7.000
 8   Minimum     0.000
 9   Maximum     1.000
10  N of Obs    90.000

Descriptive Statistics

                     1
                ------
 1      Mean     0.244
 2   Std Dev     0.430
 3       Sum    22.000
 4  Variance     0.185
 5       SSQ    22.000
 6     MCSSQ    16.622
 7  Euc Norm     4.690
 8   Minimum     0.000
 9   Maximum     1.000
10  N of Obs    90.000
```

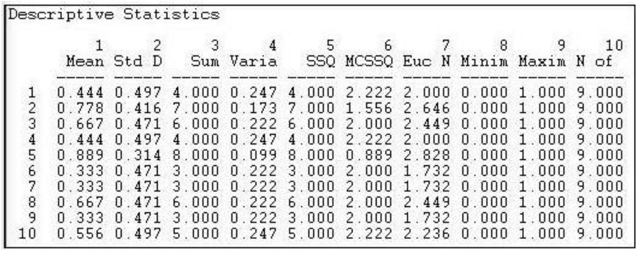
Figure 18.2: *Univariate descriptive statistics for Knoke information and money whole networks*

For the information sharing relation, we see that we have 90 observations which range from a minimum score of zero to a maximum of one. The sum of the ties is 49, and the average value of the ties is 49/90 = .544. Since the relation has been coded as a "dummy" variable (zero for no relation, one for a relation) the mean is also the proportion of possible ties that are present (or the density), or the probability that any given tie between two random actors is present (54.4% chance).

Several measures of the variability of the distribution are also given. The sums of squared deviations from the mean, variance, and standard deviation are computed -- but are more meaningful for valued than binary data. The Euclidean norm (which is the square root of the sum of squared values) is also provided. One measure not given, but sometimes helpful is the coefficient of variation (standard deviation / mean times 100) equals 91.5. This suggests quite a lot of variation as a percentage of the average score. No statistics on distributional shape (skew or kurtosis) are provided by UCINET.

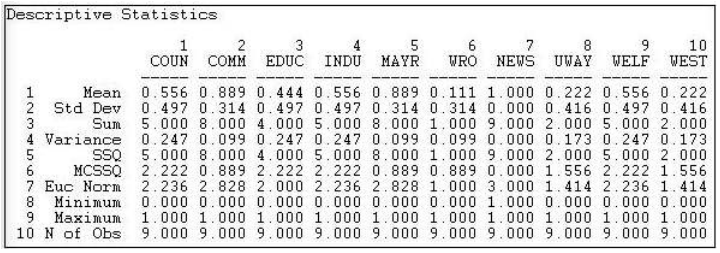
A quick scan tells us that the mean (or density) for money exchange is lower, and has slightly less variability.

In addition to examining the entire distribution of ties, we might want to examine the distribution of ties for each actor. Since the relation we're looking at is asymmetric or directed, we might further want to summarize each actor's sending (row) and receiving (column). Figures 18.3 and 18.4 show the results of *Tools>Univariate Stats* for rows (tie sending) and columns (tie receiving) of the information relation matrix.



```
Descriptive Statistics

           1      2      3      4      5      6      7      8      9     10
        Mean Std D    Sum  Varia    SSQ  MCSSQ Euc N  Minim  Maxim  N of
       ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
  1    0.444  0.497  4.000  0.247  4.000  2.222  2.000  0.000  1.000  9.000
  2    0.778  0.416  7.000  0.173  7.000  1.556  2.646  0.000  1.000  9.000
  3    0.667  0.471  6.000  0.222  6.000  2.000  2.449  0.000  1.000  9.000
  4    0.444  0.497  4.000  0.247  4.000  2.222  2.000  0.000  1.000  9.000
  5    0.889  0.314  8.000  0.099  8.000  0.889  2.828  0.000  1.000  9.000
  6    0.333  0.471  3.000  0.222  3.000  2.000  1.732  0.000  1.000  9.000
  7    0.333  0.471  3.000  0.222  3.000  2.000  1.732  0.000  1.000  9.000
  8    0.667  0.471  6.000  0.222  6.000  2.000  2.449  0.000  1.000  9.000
  9    0.333  0.471  3.000  0.222  3.000  2.000  1.732  0.000  1.000  9.000
 10    0.556  0.497  5.000  0.247  5.000  2.222  2.236  0.000  1.000  9.000
```

Figure 18.3: *Univariate descriptive statistics for Knoke information network rows*



```
Descriptive Statistics

                  1      2      3      4      5      6      7      8      9     10
               COUN   COMM   EDUC   INDU   MAYR    WRO   NEWS   UWAY   WELF   WEST
              ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
 1      Mean  0.556  0.889  0.444  0.556  0.889  0.111  1.000  0.222  0.556  0.222
 2   Std Dev  0.497  0.314  0.497  0.497  0.314  0.314  0.000  0.416  0.497  0.416
 3       Sum  5.000  8.000  4.000  5.000  8.000  1.000  9.000  2.000  5.000  2.000
 4  Variance  0.247  0.099  0.247  0.247  0.099  0.099  0.000  0.173  0.247  0.173
 5       SSQ  5.000  8.000  4.000  5.000  8.000  1.000  9.000  2.000  5.000  2.000
 6     MCSSQ  2.222  0.889  2.222  2.222  0.889  0.889  0.000  1.556  2.222  1.556
 7  Euc Norm  2.236  2.828  2.000  2.236  2.828  1.000  3.000  1.414  2.236  1.414
 8   Minimum  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000
 9   Maximum  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
10  N of Obs  9.000  9.000  9.000  9.000  9.000  9.000  9.000  9.000  9.000  9.000
```

Figure 18.4: *Univariate descriptive statistics for Knoke information network columns*

We see that actor 1 (COUN) has a mean (or density) of tie sending of 0.444. That is, this actor sent four ties to the available nine other actors. Actor 1 received somewhat more information than they sent, as their column mean is .556. In scanning down the column (in figure 18.3) or row (in figure 18.4) of means, we note that there is quite a bit of variability across actors -- some send more and get more information than others.

With valued data, the means produced index the average strength of ties, rather than the probability of ties. With valued data, measures of variability may be more informative than they are with binary data (since the variability of a binary variable is strictly a function of its mean).

The main point of this brief section is that when we use statistics to describe network data, we are describing properties of the distribution of relations, or ties among actors -- rather than properties of the distribution of attributes across actors. The basic ideas of central tendency and dispersion of distributions apply to the distributions of relational ties in exactly the same way that they do to attribute variables -- but we are describing relations, not attributes.

Hypotheses About One Mean or Density

Of the various properties of the distribution of a single variable (e.g. central tendency, dispersion, skewness), we are usually most interested in central tendency.

If we are working with the distribution of relations among actors in a network, and our measure of tie-strength is binary (zero/one), the mean or central tendency is also the proportion of all ties that are present, and is the "density."

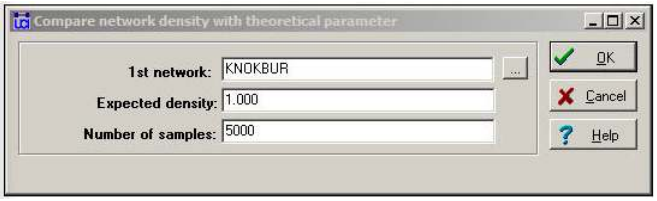
If we are working with the distribution of relations among actors in a network, and our measure of tie-strength is valued, central tendency is usually indicated by the average strength of the tie across all the relations.

We may want to test hypotheses about the density or mean tie strength of a network. In the analysis of variables, this is testing a hypothesis about a single-sample mean or proportion. We might want to be confident that there actually are ties present (null hypothesis: network density is really zero, and any deviation that we observe is due to random variation). We might want to test the hypothesis that the proportion of binary ties present differs from 0.50; we might want to test the hypothesis that the average strength of a valued tie differs from "3."

Network>Compare densities>Against theoretical parameter performs a statistical test to compare the value of a density or average tie strength observed in a network against a test value.

Let's suppose that I think that all organizations have a tendency to want to directly distribute information to all others in their field as a way of legitimating themselves. If this theory is correct, then the density of Knoke's information network should be 1.0. We can see that this isn't true. But, perhaps the difference between what we see (density = 0.544) and what the theory predicts (density = 1.000) is due to random variation (perhaps when we collected the information).

The dialog in figure 18.5 sets up the problem.



Compare network density with theoretical parameter

1st network: KNOKBUR

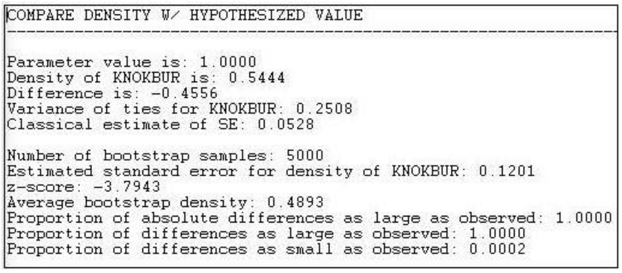
Expected density: 1.000

Number of samples: 5000

Figure 18.5: *Dialog of Compare densities>Against theoretical parameter*

The "Expected density" is the value against which we want to test. Here, we are asking the data to convince us that we can be confident in rejecting the idea that organizations send information to all others in their fields.

The parameter "Number of samples" is used for estimating the standard error for the test by the means of "bootstrapping" or computing estimated sampling variance of the mean by drawing 5000 random sub-samples from our network, and constructing a sampling distribution of density measures. The sampling distribution of a statistic is the distribution of the values of that statistic on repeated sampling. The standard deviation of the sampling distribution of a statistic (how much variation we would expect to see from sample to sample just by random chance) is called the standard error. Figure 18.6 shows the results of the hypothesis test



```
COMPARE DENSITY W/ HYPOTHESIZED VALUE
------------------------------------------------------------------------

Parameter value is: 1.0000
Density of KNOKBUR is: 0.5444
Difference is: -0.4556
Variance of ties for KNOKBUR: 0.2508
Classical estimate of SE: 0.0528

Number of bootstrap samples: 5000
Estimated standard error for density of KNOKBUR: 0.1201
z-score: -3.7943
Average bootstrap density: 0.4893
Proportion of absolute differences as large as observed: 1.0000
Proportion of differences as large as observed: 1.0000
Proportion of differences as small as observed: 0.0002
```

Figure 18.6: *Test results*

We see that our test value was 1.000, the observed value was 0.5444, so the difference between the null and observed values is -0.4556. How often would a difference this large happen by random sampling variation, if the null hypothesis (density = 1.000) was really true in the population?

Using the classical formula for the standard error of a mean $\left(s/\sqrt{N}\right)$ we obtain a sampling variability estimate of 0.0528. If we used this for our test, the test statistic would be -0.4556/0.0528 = 8.6 which would be highly significant as a t-test with N-1 degrees of freedom.

However, if we use the bootstrap method of constructing 5000 networks by sampling random sub-sets of nodes each time, and computing the density each time, the mean of this sampling distribution turns out to be 0.4893, and its standard deviation (or the standard error) turns out to be 0.1201.

Using this alternative standard error based on random draws from the observed sample, our test statistic is -3.7943. This test is also significant (p = 0.0002).

Why do this? The classical formula gives an estimate of the standard error (0.0528) that is much smaller than than that created by the bootstrap method (0.1201). This is because the standard formula is based on the notion that all observations (i.e. all relations) are independent. But, since the ties are really generated by the same 10 actors, this is not a reasonable assumption. Using the actual data on the actual actors -- with the observed differences in actor means and variances, is a much more realistic approximation to the actual sampling variability that would occur if, say, we happened to miss Fred when we collected the data on Tuesday.

In general, the standard inferential formulas for computing expected sampling variability (i.e. standard errors) give unrealistically small values for network data. Using them results in the worst kind of inferential error -- the false positive, or rejecting the null when we shouldn't.

18.3: Comparing Two Relations for the Same Set of Actors

The basic question of bivariate descriptive statistics applied to variables is whether scores on one attribute align (co-vary, correlate) with scores on another attribute, when compared across cases. The basic question of bivariate analysis of network data is whether the pattern of ties for one relation among a set of actors aligns with the pattern of ties for another relation among the same actors. That is, do the relations correlate?

Three of the most common tools for bivariate analysis of attributes can also be applied to the bivariate analysis of relations:

Does the central tendency of one relation differ significantly from the central tendency of another? For example, if we had two networks that described the military and the economic ties among nations, which has the higher density? Are military or are economic ties more prevalent? This kind of question is analogous to the test for the difference between means in paired or repeated-measures attribute analysis.

Is there a correlation between the ties that are present in one network, and the ties that are present in another? For example, are pairs of nations that have political alliances more likely to have high volumes of economic trade? This kind of question is analogous to the correlation between the scores on two variables in attribute analysis.

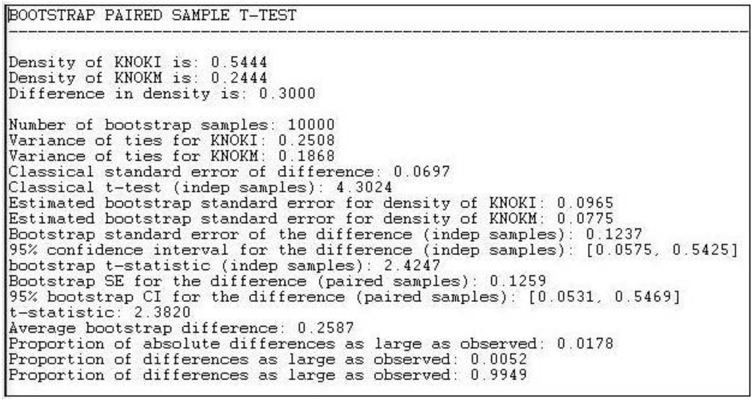
If we know that a relation of one type exists between two actors, how much does this increase (or decrease) the likelihood that a relation of another type exists between them? For example, what is the effect of a one dollar increase in the volume of trade between two nations on the volume of tourism flowing between the two nations? This kind of question is analogous to the regression of one variable on another in attribute analysis.

Hypotheses About Two Paired Means or Densities

In the section above on univariate statistics for networks, we noted that the density of the information exchange matrix for the Knoke bureaucracies appeared to be higher than the density of the monetary exchange matrix. That is, the mean or density of one relation among a set of actors appears to be different from the mean or density of another relation among the same actors.

Network>Compare densities>Paired (same node) compares the densities of two relations for the same actors, and calculates estimated standard errors to test differences by bootstrap methods. When both relations are binary, this is a test for differences in the probability of a tie of one type and the probability of a tie of another type. When both relations are valued, this is a test for a difference in the mean tie strengths of the two relations.

Let's perform this test on the information and money exchange relations in the Knoke data, as shown in Figure 18.7.



```
BOOTSTRAP PAIRED SAMPLE T-TEST
--------------------------------------------------------------------------------

Density of KNOKI is: 0.5444
Density of KNOKM is: 0.2444
Difference in density is: 0.3000

Number of bootstrap samples: 10000
Variance of ties for KNOKI: 0.2508
Variance of ties for KNOKM: 0.1868
Classical standard error of difference: 0.0697
Classical t-test (indep samples): 4.3024
Estimated bootstrap standard error for density of KNOKI: 0.0965
Estimated bootstrap standard error for density of KNOKM: 0.0775
Bootstrap standard error of the difference (indep samples): 0.1237
95% confidence interval for the difference (indep samples): [0.0575, 0.5425]
bootstrap t-statistic (indep samples): 2.4247
Bootstrap SE for the difference (paired samples): 0.1259
95% bootstrap CI for the difference (paired samples): [0.0531, 0.5469]
t-statistic: 2.3820
Average bootstrap difference: 0.2587
Proportion of absolute differences as large as observed: 0.0178
Proportion of differences as large as observed: 0.0052
Proportion of differences as large as observed: 0.9949
```

Figure 18.7: *Test for the difference of density in the Knoke information and money exchange relations*

Results for both the standard approach and the bootstrap approach (this time, we ran 10,000 sub-samples) are reported in the output. The difference between means (or proportions, or densities) is 0.3000. The standard error of the difference by the classical method is 0.0697; the standard error by bootstrap estimate is 0.1237. The conventional approach greatly underestimates the true sampling variability, and gives a result that is too optimistic in rejecting the null hypothesis that the two densities are the same.

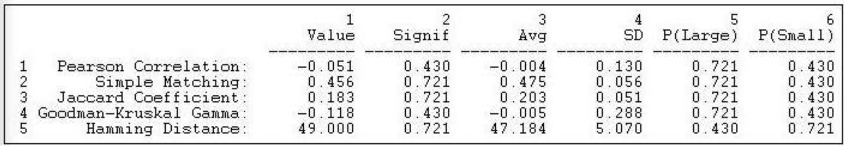
By the bootstrap method, we can see that there is a two-tailed probability of 0.0178. If we had a prior alternative hypothesis about the direction of the difference, we could use the one-tailed p level of 0.0052. So, we can conclude with great confidence that the density of information ties among organizations is greater than the density of monetary ties. That is, the observed difference would arise very rarely by chance in random samples drawn from these networks.

Correlation Between Two Networks with the Same Actors

If there is a tie between two particular actors in one relation, is there likely to be a tie between them in another relation? If two actors have a strong tie of one type, are they also likely to have a strong tie of another?

When we have information about multiple relations among the same sets of actors, it is often of considerable interest whether the probability (or strength) of a tie of one type is related to the probability (or strength) of another. Consider the Knoke information and money ties. If organizations exchange information, this may create a sense of trust, making monetary exchange relations more likely; or, if they exchange money, this may facilitate more open communications. That is, we might hypothesize that the matrix of information relations would be positively correlated with the matrix of monetary relations - pairs that engage in one type of exchange are more likely to engage in the other. Alternatively, it might be that the relations are complementary: money flows in one direction, information in the other (a negative correlation). Or, it may be that the two relations have nothing to do with one another (no correlation).

Tools>Testing Hypotheses>Dyadic (QAP)>QAP Correlation calculates measures of nominal, ordinal, and interval association between the relations in two matrices, and uses quadratic assignment procedures to develop standard errors to test for the significance of association. Figure 18.8 shows the results for the correlation between the Knoke information and monetary exchange networks.



```
                                    1          2          3          4          5          6
                                Value     Signif        Avg         SD   P(Large)   P(Small)
                           ---------- ---------- ---------- ---------- ---------- ----------
1      Pearson Correlation:    -0.051      0.430     -0.004      0.130      0.721      0.430
2          Simple Matching:     0.456      0.721      0.475      0.056      0.721      0.430
3      Jaccard Coefficient:     0.183      0.721      0.203      0.051      0.721      0.430
4 Goodman-Kruskal Gamma:       -0.118      0.430     -0.005      0.288      0.721      0.430
5         Hamming Distance:    49.000      0.721     47.184      5.070      0.430      0.721
```

Figure 18.8: *Association between Knoke information and Knoke monetary networks by QAP correlation*

The first column shows the values of five alternative measures of association. The Pearson correlation is a standard measure when both matrices have valued relations measured at the interval level. Gamma would be a reasonable choice if one or both relations were measured on an ordinal scale. Simple matching and the Jaccard coefficient are reasonable measures when both relations are binary; the Hamming distance is a measure of dissimilarity or distance between the scores in one matrix and the scores in the other (it is the number of values that differ, element-wise, from one matrix to the other).

The third column (Avg) shows the average value of the measure of association across a large number of trials in which the rows and columns of the two matrices have been randomly permuted. That is, what would the correlation (or other measure) be, on the average, if we matched random actors? The idea of the "Quadratic Assignment Procedure" is to identify the value of the measure of association when their really isn't any systematic connection between the two relations. This value, as you can see, is not necessarily zero -- because different measures of association will have limited ranges of values based on the distributions of scores in the two matrices. We note, for example, that there is an observed simple matching of 0.456 (i.e. if there is a 1 in a cell in matrix one, there is a 45.6% chance that there will be a 1 in the corresponding cell of matrix two). This would seem to indicate association. But, because of the density of the two matrices, matching randomly re-arranged matrices will display an average matching of .475. So the observed measure differs hardly at all from a random result.

To test the hypothesis that there is association, we look at the proportion of random trials that would generate a coefficient as large as (or as small as, depending on the measure) the statistic actually observed. These figures are reported (from the random permutation trials) in the columns labeled "P(large)" and "P(small)." The appropriate one of these values to test the null hypothesis of no association is shown in the column "Signif."

Network Regression

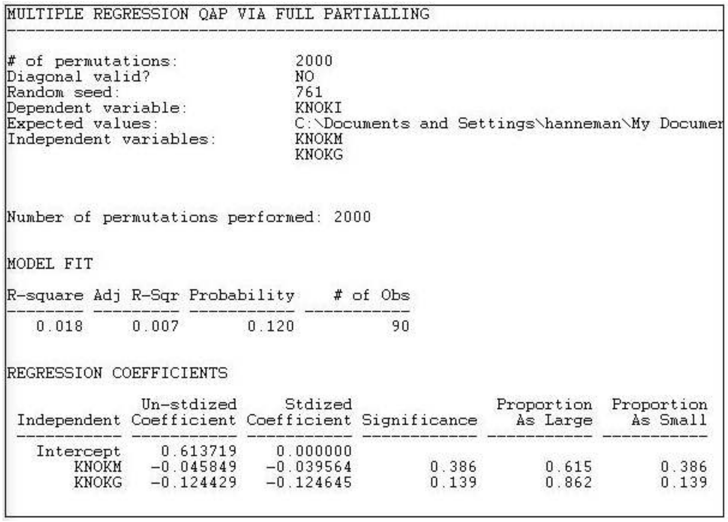
Rather than correlating one relation with another, we may wish to predict one relation knowing the other. That is, rather than symmetric association between the relations, we may wish to examine asymmetric association. The standard tool for this question is linear regression, and the approach may be extended to using more than one independent variable.

Suppose, for example, that we wanted to see if we could predict which of the Knoke bureaucracies sent information to which others. We can treat the information exchange network as our "dependent" network (with N = 90).

We might hypothesize that the presence of a money tie from one organization to another would increase the likelihood of an information tie (of course, from the previous section, we know this isn't empirically supported!). Furthermore, we might hypothesize that institutionally similar organizations would be more likely to exchange information. So, we have created another 10 by 10 matrix, coding each element to be a "1" if both organizations in the dyad are governmental bodies, or both are non-governmental bodies, and "0" if they are of mixed types.

We can now perform a standard multiple regression analysis by regressing each element in the information network on its corresponding elements in the monetary network and the government institution network. To estimate standard errors for R-squared and for the regression coefficients, we can use quadratic assignment. We will run many trials with the rows and columns in the dependent matrix randomly shuffled, and recover the R-square and regression coefficients from these runs. These are then used to assemble empirical sampling distributions to estimate standard errors under the hypothesis of no association.

Version 6.81 of UCINET offers four alternative methods for *Tools>Testing Hypotheses>Dyadic (QAP)>QAP Regression*. Figure 18.9 shows the results of the "*full partialling*" method.



```
MULTIPLE REGRESSION QAP VIA FULL PARTIALLING
--------------------------------------------------------------------------

# of permutations:            2000
Diagonal valid?               NO
Random seed:                  761
Dependent variable:           KNOKI
Expected values:              C:\Documents and Settings\hanneman\My Documer
Independent variables:        KNOKM
                              KNOKG


Number of permutations performed: 2000

MODEL FIT

R-square Adj R-Sqr Probability    # of Obs
-------- --------- ----------- -----------
   0.018     0.007       0.120          90

REGRESSION COEFFICIENTS

              Un-stdized     Stdized              Proportion  Proportion
 Independent Coefficient Coefficient Significance    As Large    As Small
 ----------- ----------- ----------- ------------ ----------- -----------
   Intercept    0.613719    0.000000
       KNOKM   -0.045849   -0.039564        0.386       0.615       0.386
       KNOKG   -0.124429   -0.124645        0.139       0.862       0.139
```

Figure 18.9: *QAP regression of information ties on money ties and governmental status by full partialling method*

The descriptive statistics and measure of goodness of fit are standard multiple regression results -- except, of course, that we are looking at predicting relations between actors, not the attributes of actors.

The model R-square (0.018) indicates that knowing whether one organization sends money to another, and whether the two organizations are institutionally similar reduces uncertainty in predicting an information tie by only about 2%. The significance level (by the QAP method) is 0.120. Usually, we would conclude that we cannot be sure the observed result is non-random.

Since the dependent matrix in this example is binary, the regression equation is interpretable as a linear probability model (one might want to consider logit or probit models -- but UCINET does not provide these). The intercept indicates that, if two organizations are not of the same institutional type, and one does not send money to the other, the probability that one sends information to the other is 0.61. If one organization does send money to the other, this reduces the probability of an information link by 0.046.

If the two organizations are of the same institutional type, the probability of information sending is reduced by 0.124.

Using the QAP method, however, none of these effects are different from zero at conventional (e.g. $p < 0.05$) levels. The results are interesting - they suggest that monetary and informational linkages are, if anything, alternative rather than re-enforcing ties, and that institutionally similar organizations are less likely to communicate. But, we shouldn't take these apparent patterns seriously, because they could appear quite frequently simply by random permutation of the cases.

The tools in the this section are very useful for examining how multiplex relations among a set of actors "go together." These tools can often be helpful additions to some of the tools for working with multiplex data that we examined in chapter 16.

18.4: Explaining Attributes of Networked Actors

In the previous section we examined methods for testing differences and association among whole networks. That is, studying the macro-patterns of how an actor's position in one network might be associated with their position in another.

We are often interested in micro questions, as well. For example: does an actor's gender affect their between-ness centrality? This question relates an attribute (gender) to a measure of the actor's position in a network (between-ness centrality). We might be interested in the relationship between two (or more) aspects of actor's positions. For example: how much of the variation in actor's between-ness centrality can be explained by their out-degree and the number of cliques that they belong to? We might even be interested in the relationship between two individual attributes among a set of actors who are connected in a network. For example, in a school classroom, is there an association between actor's gender and their academic achievement?

In all of these cases we are focusing on variables that describe individual nodes. These variables may be either non-relational attributes (like gender), or variables that describe some aspect of an individual's relational position (like between-ness). In most cases, standard statistical tools for the analysis of variables can be applied to describe differences and associations.

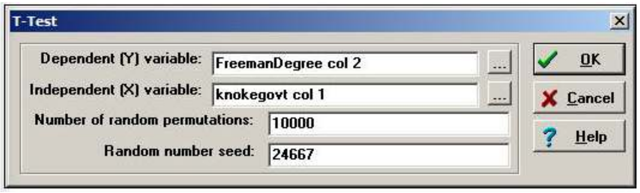
But, standard statistical tools for the analysis of variables cannot be applied to inferential questions -- hypothesis or significance tests, because the individuals we are examining are not independent observations drawn at random from some large population. Instead of applying the normal formulas (i.e. those built into statistical software packages and discussed in most basic statistics texts), we need to use other methods to get more correct estimates of the reliability and stability of estimates (i.e. standard errors). The "boot-strapping" approach (estimating the variation of estimates of the parameter of interest from large numbers of random sub-samples of actors) can be applied in some cases; in other cases, the idea of random permutation can be applied to generate correct standard errors.

Hypotheses About the Means of Two Groups

Suppose we had the notion that private-for-profit organizations were less likely to actively engage in sharing information with others in their field than were government organizations. We would like to test this hypothesis by comparing the average out-degree of governmental and non-governmental actors in one organizational field.

Using the Knoke information exchange network, we've run *Network>Centrality>Degree*, and saved the results in the output file "FreemanDegree" as a UCINET dataset. We've also used *Data>Spreadsheets>Matrix* to create a UCINET attribute file "knokegovt" that has a single column dummy code (1 = governmental organization, 0 = non-governmental organization).

Let's perform a simple two-sample t-test to determine if the mean degree centrality of government organizations is lower than the mean degree centrality of non-government organizations. Figure 18.10 shows the dialog for *Tools>Testing Hypotheses>Node-level>T-Test* to set up this test.

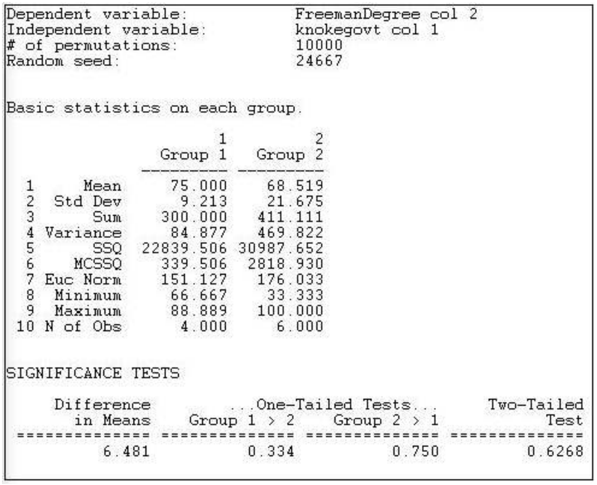


Dependent (Y) variable: FreemanDegree col 2
Independent (X) variable: knokegovt col 1
Number of random permutations: 10000
Random number seed: 24667

Figure 18.10: *Dialog for Tools>Testing Hypotheses>Node-level>T-Test*

Since we are working with individual nodes as observations, the data are located in a column (or, sometimes, a row) of one or more files. Note how the file names (selected by browsing, or typed) and the columns within the file are entered in the dialog. The normed Freeman degree centrality measure happens to be located in the second column of its file; there is only one vector (column) in the file that we created to code government/non-government organizations.

For this test, we have selected the default of 10,000 trials to create the permutation-based sampling distribution of the difference between the two means. For each of these trials, the scores on normed Freeman degree centralization are randomly permuted (that is, randomly assigned to government or non-government, proportional to the number of each type.) The standard deviation of this distribution based on random trials becomes the estimated standard error for our test. Figure 18.11 shows the results.



```
Dependent variable:              FreemanDegree col 2
Independent variable:            knokegovt col 1
# of permutations:               10000
Random seed:                     24667


Basic statistics on each group.

                            1         2
                      Group 1   Group 2
                   ---------- ----------
  1      Mean         75.000     68.519
  2   Std Dev          9.213     21.675
  3       Sum        300.000    411.111
  4  Variance         84.877    469.822
  5       SSQ      22839.506  30987.652
  6     MCSSQ        339.506   2818.930
  7  Euc Norm        151.127    176.033
  8   Minimum         66.667     33.333
  9   Maximum         88.889    100.000
 10  N of Obs          4.000      6.000


SIGNIFICANCE TESTS

     Difference          ...One-Tailed Tests...         Two-Tailed
       in Means      Group 1 > 2     Group 2 > 1              Test
 ============== ============== ============== ==============
          6.481          0.334          0.750           0.6268
```

Figure 18.11: *Test for difference in mean normed degree centrality of Knoke government and non-government organizations*

The output first reports basic descriptive statistics for each group. The group numbers are assigned according to the order of the cases in the file containing the independent variable. In our example, the first node was COUN, a government organization; so, government became "Group 1" and non-government became "Group 2."

We see that the average normed degree centrality of government organizations (75) is 6.481 units higher than the average normed degree centrality of non-governmental organizations (68.519). This would seem to support our hypothesis; but tests of statistical significance urge considerable caution. Differences as large as 6.481 in favor of government organizations happen 33.4% of the time in random trials -- so we would be taking an unacceptable risk of being wrong if we concluded that the data were consistent with our research hypothesis.

UCINET does not print the estimated standard error, or the values of the conventional two-group t-test.

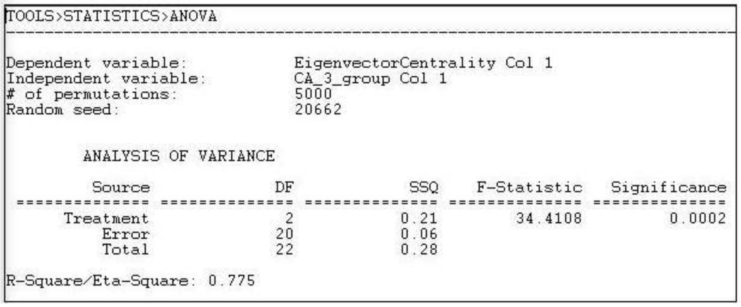
Hypotheses About the Means of Multiple Groups

The approach to estimating difference between the means of two groups discussed in the previous section can be extended to multiple groups with one-way analysis of variance (ANOVA). The procedure *Tools>Testing Hypotheses>Node-level>Anova* provides the regular OLS approach to estimating differences in group means. Because our observations are not independent, the procedure of estimating standard errors by random replications is also applied.

Suppose we divided the 23 large donors to California political campaigns into three groups, and have coded a single column vector in a UCINET attribute file. We've coded each donor as falling into one of three groups: "others," "capitalists," or "workers."

If we examine the network of connections among donors (defined by co-participating in the same campaigns), we anticipate that the worker's groups will display higher eigenvector centrality than donors in the other groups. That is, we anticipate that the "left" interest groups will display considerable interconnection, and -- on the average -- have members that are more connected to highly connected others than is true for the capitalist and other groups. We've calculated eigenvector centrality using *Network>Centrality>Eigenvector*, and stored the results in another UCINET attribute file.

The dialog for *Tools>Testing Hypotheses>Node-level>Anova* looks very much like *Tools>Testing Hypotheses>Node-level>T-test*, so we won't display it. The results of our analysis are shown as figure 18.12.



```
TOOLS>STATISTICS>ANOVA
--------------------------------------------------------------------------------

Dependent variable:           EigenvectorCentrality Col 1
Independent variable:         CA_3_group Col 1
# of permutations:            5000
Random seed:                  20662

        ANALYSIS OF VARIANCE

         Source              DF             SSQ    F-Statistic   Significance
 ============== ============== ============== ============== ==============
      Treatment               2            0.21        34.4108         0.0002
          Error              20            0.06
          Total              22            0.28

R-Square/Eta-Square: 0.775
```

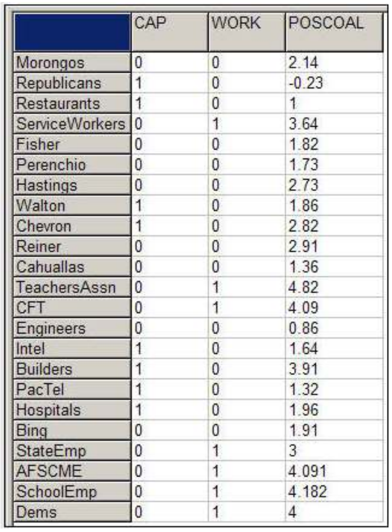
Figure 18.12: *One-way ANOVA of eigenvector centrality of California political donors, with permutation-based standard errors and tests*

The mean eigenvector centrality of the eight "other" donors is .125. For the seven "capitalists" it is 0.106, and for the seven "workers" groups it is 0.323 (calculated elsewhere). The differences among these means is highly significant (F = 34.4 with 2 d.f. and p = 0.0002). The differences in group means account for 78% of the total variance in eigenvector centrality scores among the donors.

Regressing Position on Attributes

Where the attribute of actors that we are interested in explaining or predicting is measured at the interval level, and one or more of our predictors are also at the interval level, multiple linear regression is a common approach. *Tools>Testing Hypotheses>Node-level>Regression* will compute basic linear multiple regression statistics by OLS, and estimate standard errors and significance using the random permutations method for constructing sampling distributions of R-squared and slope coefficients.

Let's continue the example in the previous section. Our dependent attribute, as before, is the eigenvector centrality of the individual political donors. This time, we will use three independent vectors, which we have constructed using *Data>Spreadsheets>Matrix*, as shown in figure 18.13.



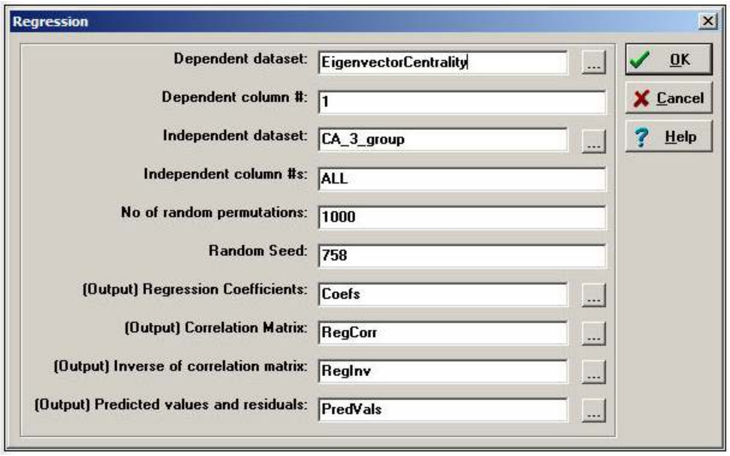
	CAP	WORK	POSCOAL
Morongos	0	0	2.14
Republicans	1	0	-0.23
Restaurants	1	0	1
ServiceWorkers	0	1	3.64
Fisher	0	0	1.82
Perenchio	0	0	1.73
Hastings	0	0	2.73
Walton	1	0	1.86
Chevron	1	0	2.82
Reiner	0	0	2.91
Cahuallas	0	0	1.36
TeachersAssn	0	1	4.82
CFT	0	1	4.09
Engineers	0	0	0.86
Intel	1	0	1.64
Builders	1	0	3.91
PacTel	1	0	1.32
Hospitals	1	0	1.96
Bing	0	0	1.91
StateEmp	0	1	3
AFSCME	0	1	4.091
SchoolEmp	0	1	4.182
Dems	0	1	4

Figure 18.13: *Construction of independent vectors for multiple linear regression*

Two dummy variables have been constructed to indicate whether each donor is a member of the "capitalist" or the "worker" group. The omitted category ("other") will serve as the intercept/reference category. POSCOAL is the mean number of times that each donor participates on the same side of issues with other donors (a negative score indicates opposition to other donors).

Substantively, we are trying to find out whether the "workers" higher eigenvector centrality (observed in the section above) is simply a function of higher rates of participation in coalitions, or whether the workers have better connected allies -- independent of high participation.

Figure 18.14 shows the dialog to specify the dependent and the multiple independent vectors.



Dependent dataset: EigenvectorCentrality

Dependent column #: 1

Independent dataset: CA_3_group

Independent column #s: ALL

No of random permutations: 1000

Random Seed: 758

(Output) Regression Coefficients: Coefs

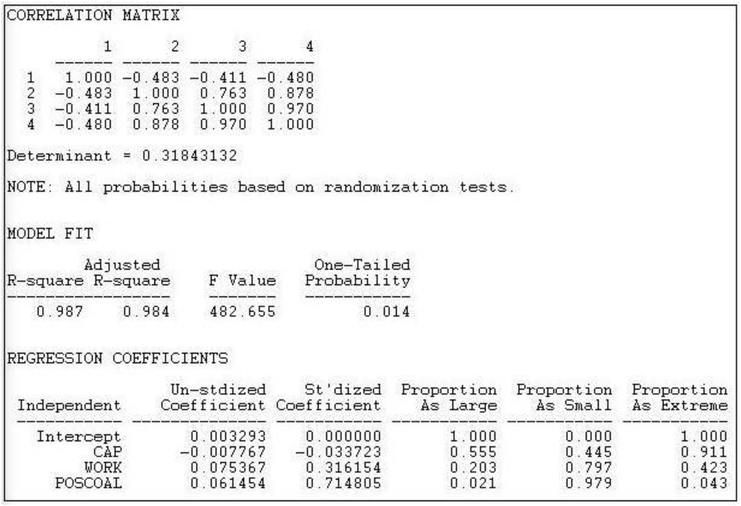
(Output) Correlation Matrix: RegCorr

(Output) Inverse of correlation matrix: RegInv

(Output) Predicted values and residuals: PredVals

Figure 18.14: *Dialog for Tools>Testing Hypotheses>Node-level>Regression*

Note that all of the independent variables need to be entered into a single data set (with multiple columns). All of the basic regression statistics can be saved as output, for use in graphics or further analysis. Figure 18.15 shows the result of the multiple regression estimation.



```
CORRELATION MATRIX

          1      2      3      4
     ------ ------ ------ ------
  1   1.000 -0.483 -0.411 -0.480
  2  -0.483  1.000  0.763  0.878
  3  -0.411  0.763  1.000  0.970
  4  -0.480  0.878  0.970  1.000

Determinant = 0.31843132

NOTE: All probabilities based on randomization tests.


MODEL FIT

        Adjusted                One-Tailed
R-square R-square    F Value   Probability
-----------------    -------   -----------
   0.987    0.984    482.655         0.014


REGRESSION COEFFICIENTS

                  Un-stdized    St'dized  Proportion  Proportion  Proportion
 Independent     Coefficient Coefficient    As Large    As Small  As Extreme
 -----------  -------------- ----------- ----------- ----------- -----------
   Intercept        0.003293    0.000000       1.000       0.000       1.000
         CAP       -0.007767   -0.033723       0.555       0.445       0.911
        WORK        0.075367    0.316154       0.203       0.797       0.423
     POSCOAL        0.061454    0.714805       0.021       0.979       0.043
```

Figure 18.15: *Multiple regression of eigenvector centrality with permutation based significance tests*

The correlation matrix shows a very high collinearity between being in the workers group (variable 3) and participation in coalitions (variable 4). This suggests that it may be difficult to separate effects of simple participation from those of being a workers interest group.

The R-squared is very high for this simple model (.987), and highly significant using permutation tests (p = 0.014).

Controlling for total coalition participation, capitalist interests are likely to have slightly lower eigenvector centrality than others (-0.0078), but this is not significant (p = 0.555). Workers groups do appear to have higher eigenvector centrality, even controlling for total coalition participation (0.075), but this tendency may be a random result (a one-tailed significance is only p = 0.102). The higher the rate of participation in coalitions (POSCOAL), the greater the eigenvector centrality of actors (0.0615, p = 0.021), regardless of which type of interest is being represented.

As before, the coefficients are generated by standard OLS linear modeling techniques, and are based on comparing scores on independent and dependent attributes of individual actors. What differs here is the recognition that the actors are not independent, so that estimation of standard errors by simulation, rather than by standard formula, is necessary.

The t-test, ANOVA, and regression approaches discussed in this section are all calculated at the micro, or individual actor level. The measures that are analyzed as independent and dependent may be either relational or non-relational. That is, we could be interested in predicting and testing hypotheses about actors non-relational attributes (e.g. their income) using a mix of relational (e.g. centrality) and non-relational (e.g. gender) attributes. We could be interested in predicting a relational attribute of actors (e.g. centrality) using a mix of relational and non-relational independent variables.

The examples illustrate how relational and non-relational attributes of actors can be analyzed using common statistical techniques. The key thing to remember, though, is that the observations are not independent (since all the actors are members of the same network). Because of this, direct estimation of the sampling distributions and resulting inferential statistics is needed -- standard, basic statistical software will not give correct answers.

18.5: Explaining the Relations Among Actors in a Network

In the previous section we looked at some tools for hypotheses about individual actors embedded in networks. Models like these are very useful for examining the relationships among relational and non-relational attributes of individuals.

One of the most distinctive ways in which statistical analysis has been applied to social network data is to focus on predicting the relations of actors, rather than their attributes. Rather than building a statistical model to predict each actor's out-degree, we could, instead, predict whether there was a tie from each actor to each other actor. Rather than explaining the variance in individual persons, we could focus on explaining variation in the relations.

In this final section, we will look at several statistical models that seek to predict the presence or absence (or strength) of a tie between two actors. Models like this are focusing directly on a very sociological question: what factors affect the likelihood that two individuals will have a relationship?

One obvious, but very important, predictor of whether two actors are likely to be connected is their similarity or closeness. In many sociological theories, two actors who share some attribute are predicted to be more likely to form social ties than two actors who do not. This "homophily" hypothesis is at the core of many theories of differentiation, solidarity, and conflict. Two actors who are closer to one in a network are often hypothesized to be more likely to form ties; two actors who share attributes are likely to be at closer distances to one another in networks.

Several of the models below explore homophily and closeness to predict whether actors have ties, or are close to one another. The last model that we will look at the "P1" model also seeks to explain relations. The P1 model tries to predict whether there exists no relation, an asymmetrical relation, or a reciprocated tie between pairs of actors. Rather than using attributes or closeness as predictors, however, the P1 model focuses on basic network properties of each actor and the network as a whole (in-degree, out-degree, global reciprocity). This type of model -- a probability model for the presence/absence of each possible relation in a graph as a function of network structures -- is one of the major continuing areas of development in social network methods.

Hypotheses About Relations Within/Between Groups

One of the most commonplace sociological observations is that "birds of a feather flock together." The notion that similarity (or homophily) increases the probability of the formation of social ties is central to most sociological theories. The homophily hypothesis can be read to be making a prediction about social networks. It suggests that if two actors are similar in some way, it is more likely that there will be network ties between them. If we look at a social network that contains two types of actors, the density of ties ought to be greater within each group than between groups.

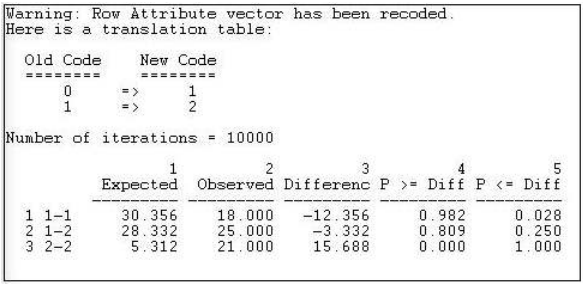
Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Categorical Attributes>Joint-Count Provides a test that the density of ties within and between two groups differs from what we would expect if ties were distributed at random across all pairs of nodes.

The procedure takes a binary graph and a partition (that is, a vector that classifies each node as being in one group or the other), and permutes and blocks the data. If there was no association between sharing the same attribute (i.e. being in the same block) and the likelihood of a tie between two actors, we can predict the number of ties that ought to be present in each of the four blocks of the graph (that is: group 1 by group 1; group 1 by group 2; group 2 by group 1; and group 2 by group 2). These four "expected frequencies" can then be compared to the four "observed frequencies." The logic is exactly the same as the Pearson Chi-square test of independence -- we can generate a "test statistic" that shows how far the 2 by 2 table departs from "independence" or "no association."

To test the inferential significance of departures from randomness, however, we cannot rely on standard statistical tables. Instead, a large number of random graphs with the same overall density and the same sized partitions are calculated. The sampling distribution of differences between observed and expected for random graphs can then be calculated, and used to assess the likelihood that our observed graph could be a result of a random trial from a population where there was no association between group membership and the likelihood of a relation.

To illustrate, if two large political donors contributed on the same side of political campaigns (across 48 initiative campaigns), we code them "1" as having a tie or relation, otherwise, we code them zero. We've divided our large political donors in California initiative campaigns into two groups -- those that are affiliated with "workers" (e.g. unions, the Democratic party), and those that are not.

We would anticipate that two groups that represent workers interests would be more likely to share the tie of being in coalitions to support initiatives than would two groups drawn at random. Figure 18.16 shows the results of *Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Categorical Attributes>Joint-Count* applied to this problem.



```
Warning: Row Attribute vector has been recoded.
Here is a translation table:

  Old Code      New Code
  ========      ========
      0     =>      1
      1     =>      2

Number of iterations = 10000

                    1          2          3          4          5
             Expected   Observed Differenc P >= Diff P <= Diff
           ---------- ---------- ---------- ---------- ----------
  1 1-1        30.356     18.000    -12.356      0.982      0.028
  2 1-2        28.332     25.000     -3.332      0.809      0.250
  3 2-2         5.312     21.000     15.688      0.000      1.000
```

Figure 18.16: *Test for two-group differences in tie density*

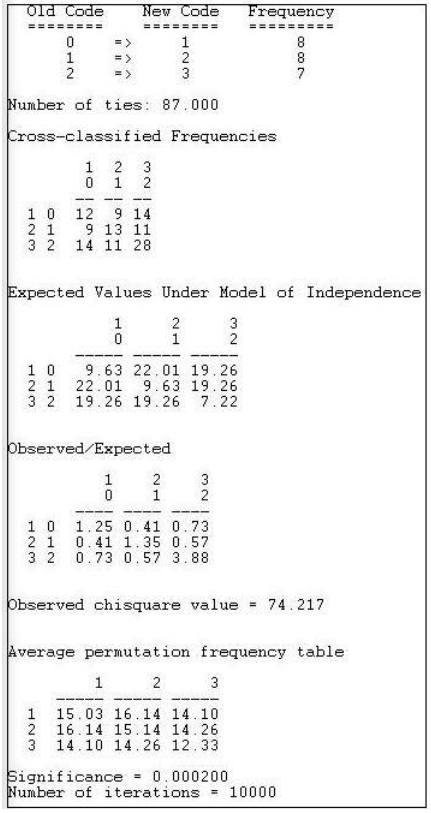
The partition vector (group identification variable) was originally coded as zero for non-worker donors and one for worker donors. These have been re-labeled in the output as one and two. We've used the default of 10,000 random graphs to generate the sampling distribution for group differences.

The first row, labeled "1-1" tells us that, under the null hypothesis that ties are randomly distributed across all actors (i.e. group makes no difference), we would expect 30.356 ties to be present in the non-worker to non-worker block. We actually observe 18 ties in this block, 12 fewer than would be expected. A negative difference this large occurred only 2.8% of the time in graphs where the ties were randomly distributed. It is clear that we have a deviation from randomness within the "non-worker" block. But the difference does not support homophily -- it suggest just the opposite; ties between actors who share the attribute of not representing workers are less likely than random, rather than more likely.

The second row, labeled "1-2" shows no significant difference between the number of ties observed between worker and non-worker groups and what would happen by chance under the null hypothesis of no effect of shared group membership on tie density.

The third row, labeled "2-2" A difference this large indicates that the observed count of ties among interest groups representing workers (21) is much greater than expected by chance (5.3) would almost never be observed if the null hypothesis of no group effect on the probability of ties were true.

Perhaps our result does not support homophily theory because the group "non-worker" is not really as social group at all -- just a residual collection of diverse interests. Using *Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Categorical Attributes>Relational Contingency-Table Analysis* we can expand the number of groups to provide a better test. This time, let's categorize the political donors as representing "others," "capitalists," or "workers." The results of this test are shown as figure 18.17.



```
  Old Code     New Code    Frequency
  ========     ========    =========
      0     =>     1           8
      1     =>     2           8
      2     =>     3           7

Number of ties: 87.000

Cross-classified Frequencies

           1  2  3
           0  1  2
          -- -- --
  1 0     12  9 14
  2 1      9 13 11
  3 2     14 11 28


Expected Values Under Model of Independence

              1      2      3
              0      1      2
          ------ ------ ------
  1 0       9.63  22.01  19.26
  2 1      22.01   9.63  19.26
  3 2      19.26  19.26   7.22


Observed/Expected

            1     2     3
            0     1     2
          ----  ----  ----
  1 0     1.25  0.41  0.73
  2 1     0.41  1.35  0.57
  3 2     0.73  0.57  3.88


Observed chisquare value = 74.217


Average permutation frequency table

           1      2      3
        ------ ------ ------
  1     15.03  16.14  14.10
  2     16.14  15.14  14.26
  3     14.10  14.26  12.33

Significance = 0.000200
Number of iterations = 10000
```

Figure 18.17: *Test for three-group differences in tie density*

The "other" group has been re-labeled "1," the "capitalist" group re-labeled "2," and the "worker" group re-labeled "3." There are 87 total ties in the graph, with the observed frequencies shown ("Cross-classified Frequencies).

We can see that the the observed frequencies differ from the "Expected Values Under Model of Independence." The magnitudes of the over and under-representation are shown as "Observed/Expected." We note that all three diagonal cells (that is, ties within groups) now display homophily -- greater than random density.

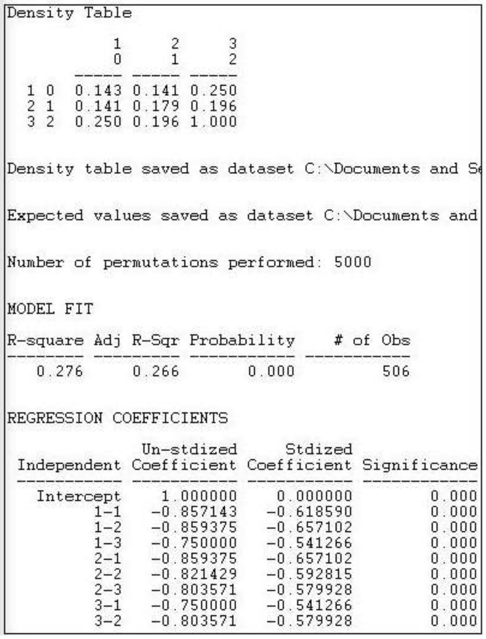
A Pearson chi-square statistic is calculated (74.217). And, we are shown the average tie counts in each cell that occurred in the 10,000 random trials. Finally, we observe that p < 0.0002. That is, the deviation of ties from randomness is so great that it would happen only very rarely if the no-association model was true.

Homophily Models

The result in the section above seems to support homophily (which we can see by looking at where the deviations from independence occur. The statistical test, though, is just a global test of difference from random distribution. The routine *Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Categorical Attributes>ANOVA Density Models* provides specific tests of some quite specific homophily models.

The least-specific notion of how members of groups relate to members of other groups is simply that the groups differ. Members of one group may prefer to have ties only within their group; members of another group might prefer to have ties only outside of their group.

The *Structural Blockmodel* option of *Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Categorical Attributes>ANOVA Density Models* provides a test that the patterns of within and between group ties differ across groups -- but does not specify in what way they may differ. Figure 18.18 shows the results of fitting this model to the data on strong coalition ties (sharing 4 or more campaigns) among "other," "capitalist," and "worker" interest groups.



```
Density Table

             1     2     3
             0     1     2
         ----- ----- -----
  1 0    0.143 0.141 0.250
  2 1    0.141 0.179 0.196
  3 2    0.250 0.196 1.000


Density table saved as dataset C:\Documents and Se

Expected values saved as dataset C:\Documents and

Number of permutations performed: 5000

MODEL FIT

R-square Adj R-Sqr Probability    # of Obs
-------- --------- ----------- -----------
   0.276     0.266       0.000         506

REGRESSION COEFFICIENTS

                 Un-stdized      Stdized
 Independent Coefficient Coefficient Significance
 ----------- ----------- ----------- ------------
   Intercept    1.000000    0.000000        0.000
         1-1   -0.857143   -0.618590        0.000
         1-2   -0.859375   -0.657102        0.000
         1-3   -0.750000   -0.541266        0.000
         2-1   -0.859375   -0.657102        0.000
         2-2   -0.821429   -0.592815        0.000
         2-3   -0.803571   -0.579928        0.000
         3-1   -0.750000   -0.541266        0.000
         3-2   -0.803571   -0.579928        0.000
```

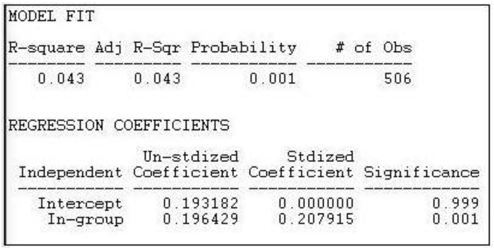
Figure 18.18: *Structural block model of differences in group tie density*

The observed density table is shown first. Members of the "other" group have a low probability of being tied to one another (0.143) or to "capitalists" (0.143), but somewhat stronger ties to "workers" (0.250). Only the "workers" (category 2, row 3) show strong tendencies toward within-group ties.

Next, a regression model is fit to the data. The presence or absence of a tie between each pair of actors is regressed on a set of dummy variables that represent each of cells of the 3-by-3 table of blocks. In this regression, the last block (i.e. 3-3) is used as the reference category. In our example, the differences among blocks explain 27.6% of the variance in the pair-wise presence or absence of ties. The probability of a tie between two actors, both of whom are in the "workers" block (block 3) is 1.000. The probability in the block describing ties between "other" and "other" actors (block 1-1) is .857 less than this.

The statistical significance of this model cannot be properly assessed using standard formulas for independent observations. Instead, 5000 trials with random permutations of the presence and absence of ties between pairs of actors have been run, and estimated standard errors calculated from the resulting simulated sampling distribution.

A much more restricted notion of group differences is named the *Constant Homophily* model in *Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Categorical Attributes>ANOVA Density Models*. This model proposes that all groups may have a preference for within-group ties, but that the strength of the preference is the same within all groups. The results of fitting this model to the data is shown in figure 18.19.



```
MODEL FIT

R-square Adj R-Sqr Probability    # of Obs
-------- --------- ----------- -----------
   0.043     0.043       0.001         506


REGRESSION COEFFICIENTS

              Un-stdized      Stdized
 Independent Coefficient Coefficient Significance
 ----------- ----------- ----------- ------------
   Intercept    0.193182    0.000000        0.999
    In-group    0.196429    0.207915        0.001
```

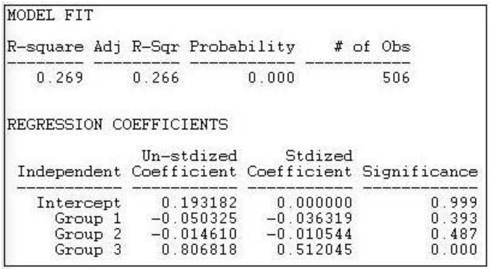
Figure 18.19: *Constant Homophily block model of differences in group tie density*

Given what we observed in looking directly at the block densities (shown in figure 18.18), it is not surprising that the constant homophily model does not fit these data well. We know that two of the groups ("others" and "capitalists") have no apparent tendency to homophily -- and that differs greatly from the "workers" group. The block model of group differences only accounts for 4.3% of the variance in pair-wise ties; however, permutation trials suggest that this is not a random result (p = 0.001).

This model only has two parameters, because the hypothesis is proposing a simple difference between the diagonal cells (the within group ties 1-1, 2-2, and 3-3) and all other cells. The hypothesis is that the densities within these two partitions are the same. We see that the estimated average tie density of pairs who are not in the same group is 0.193 -- there is a 19.3% chance that heterogeneous dyads will have a tie. If the members of the dyad are from the same group, the probability that they share a tie is 0.196 greater, or 0.389.

So, although the model of constant homophily does not predict individual's ties at all well, there is a notable overall homophily effect.

We noted that the strong tendency toward within-group ties appears to describe only the "workers" group. A third block model, labeled *Variable Homophily* by *Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Categorical Attributes>ANOVA Density Models* tests the model that each diagonal cell (that is ties within group 1, within group 2, and within group 3) differ from all ties that are not within-group. Figure 18.20 displays the results.



```
MODEL FIT

R-square Adj R-Sqr Probability    # of Obs
-------- --------- ----------- -----------
   0.269     0.266       0.000         506

REGRESSION COEFFICIENTS

              Un-stdized      Stdized
 Independent Coefficient Coefficient Significance
 ----------- ----------- ----------- ------------
   Intercept    0.193182    0.000000        0.999
     Group 1   -0.050325   -0.036319        0.393
     Group 2   -0.014610   -0.010544        0.487
     Group 3    0.806818    0.512045        0.000
```

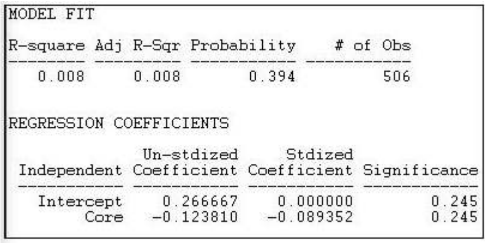
Figure 18.20: *Variable homophily block model of differences in group tie density*

This model fits the data much better (R-square = 0.269, with p < 0.000) than the constant homophily model. It also fits the data nearly as well as the un-restricted structural block model (Figure 18.18), but is simpler.

Here, the intercept is the probability that there well be a dyadic tie between any two members of different groups (0.193). We see that the probability of within group ties among group 1 ("others") is actually 0.05 less than this (but not significantly different). Within group ties among capitalist interest groups (group 2) are very slightly less common (-0.01) than heterogeneous group ties (again, not significant). Ties among interest groups representing workers (group 3) however, are dramatically more prevalent (0.81) than ties within heterogeneous pairs.

In our example, we noted that one group seems to display in-group ties, and others do not. One way of thinking about this pattern is a "core-periphery" block model. There is a strong form, and a more relaxed form of the core-periphery structure.

The *Core-periphery 1* model supposes that there is a highly organized core (many ties within the group), but that there are few other ties -- either among members of the periphery, or between members of the core and members of the periphery. Figure 18.21 shows the results of fitting this block model to the California donors data.

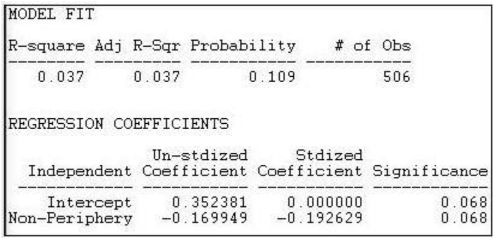


```
MODEL FIT

R-square Adj R-Sqr Probability    # of Obs
-------- --------- ----------- -----------
   0.008     0.008       0.394         506

REGRESSION COEFFICIENTS

              Un-stdized      Stdized
 Independent Coefficient Coefficient Significance
 ----------- ----------- ----------- ------------
   Intercept    0.266667    0.000000        0.245
        Core   -0.123810   -0.089352        0.245
```

Figure 18.21: *"Strong" core-periphery block model of California political donors*

It's clear that this model does not do a good job of describing the pattern of within and between group ties. The R-square is very low (0.008), and results this small would occur 39.4% of the time in trials from randomly permuted data. The (non-significant) regression coefficients show density (or the probability of a tie between two random actors) in the periphery as .27, and the density in the "Core" as 0.12 less than this. Since the "core" is, by definition, the maximally dense area, it appears that the output in version 6.8.5 may be mislabeled.

Core-Periphery 2 offers a more relaxed block model in which the core remains densely tied within itself, but is allowed to have ties with the periphery. The periphery is, to the maximum degree possible, a set of cases with no ties within their group. Figure 18.22 shows the results of this model for the California political donors.



```
MODEL FIT

R-square Adj R-Sqr Probability    # of Obs
-------- --------- -----------  -----------
   0.037     0.037       0.109          506

REGRESSION COEFFICIENTS

                Un-stdized      Stdized
  Independent  Coefficient  Coefficient  Significance
  -----------  -----------  -----------  ------------
    Intercept     0.352381     0.000000         0.068
Non-Periphery    -0.169949    -0.192629         0.068
```

Figure 18.22: *"Relaxed" core-periphery block model of California political donors*

The fit of this model is better (R-square = 0.037) but still very poor. Results this strong would occur about 11% of the time in trials from randomly permuted data. The intercept density (which we interpret as the "non-periphery") is higher (about 35% of all ties are present), and the probability of a tie between two cases in the periphery is .17 lower.

Hypotheses About Similarity and Distance

The homophily hypothesis is often thought of in categorical terms: is there a tendency for actors who are of the same "type" to be adjacent (or close) to one another in the network?

This idea, though, can be generalized to continuous attributes: is there a tendency for actors who have more similar attributes to be located closer to one another in a network?

UCINET's *Tools>Testing Hypotheses>Mixed Dyadic?Nodal>Continuous Attributes>Moran/Geary statistics* provides two measures that address the question of the "autocorrelation" between actor's scores on interval-level measures of their attributes, and the network distance between them. The two measures (Moran's I and Geary's C) are adapted for social network analysis from their origins in geography, where they were developed to measure the extent to which the similarity of the geographical features of any two places was related to the spatial distance between them.

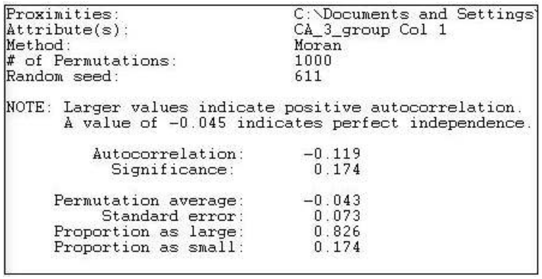
Let's suppose that we were interested in whether there was a tendency for political interest groups that were "close" to one another to spend similar amounts of money. We might suppose that interest groups that are frequent allies may also influence one another in terms of the levels of resources they contribute -- that a sort of norm of expected levels of contribution arises among frequent allies.

Using information about the contributions of very large donors (who gave over a total of $5,000,000) to at least four (of 48) ballot initiatives in California, we can illustrate the idea of network autocorrelation.

First, we create an attribute file that contains a column that has the attribute score of each node, in this case, the amount of total expenditures by the donors.

Second, we create a matrix data set that describes the "closeness" of each pair of actors. There are several alternative approaches here. One is to use an adjacency (binary) matrix. We will illustrate this by coding two donors as adjacent if they contributed funds on the same side of at least four campaigns (here, we've constructed adjacency from "affiliation" data; often we have a direct measure of adjacency, such as one donor naming another as an ally). We could also use a continuous measure of the strength of the tie between actors as a measure of "closeness." To illustrate this, we will use a scale of the similarity of the contribution profiles of donors that ranges from negative numbers (indicating that two donors gave money on opposite sides of initiatives) to positive numbers (indicating the number of times the donated on the same side of issues. One can easily imagine other approaches to indexing the network closeness of actors (e.g. 1/geodesic distance). Any "proximity" matrix that captures the pair-wise closeness of actors can be used (for some ideas, see *Tools>Similarities* and *Tools>Dissimilarities and Distances*).

Figures 18.23 and 18.24 display the results of *Tools>Testing Hypotheses>Mixed Dyadic/Nodal>Continuous Attributes>Moran/Geary statistics* where we have examined the autocorrelation of the levels of expenditures of actors using adjacency as our measure of network distance. Very simply: do actors who are adjacent in the network tend to give similar amounts of money? Two statistics and some related information are presented (the Moran statistic in Figure 18.23, and the Geary statistic in Figure 18.24.



```
Proximities:                  C:\Documents and Settings\
Attribute(s):                 CA_3_group Col 1
Method:                       Moran
# of Permutations:            1000
Random seed:                  611

NOTE: Larger values indicate positive autocorrelation.
      A value of -0.045 indicates perfect independence.

         Autocorrelation:      -0.119
           Significance:        0.174

     Permutation average:      -0.043
          Standard error:       0.073
     Proportion as large:       0.826
     Proportion as small:       0.174
```

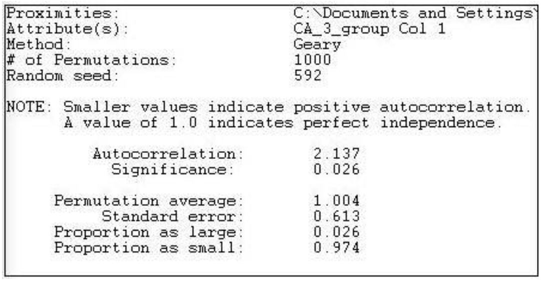
Figure 18.23: *Moran autocorrelation of expenditure levels by political donors with network adjacency*

The Moran "I" statistic of autocorrelation (originally developed to measure spatial autocorrelation, but used here to measure network autocorrelation) ranges from -1.0 (perfect negative correlation) through 0 (no correlation) to +1.0 (perfect positive correlation). Here we see the value of -0.119, indicating that there is a very modest tendency for actors who are adjacent to differ more in how much they contribute than two random actors. If anything, it appears that coalition members may vary more in their contribution levels than random actors -- another hypothesis bites the dust!

The Moran statistic (see any geo-statistics text, or do a Google search) is constructed very much like a regular correlation coefficient. It indexes the product of the differences between the scores of two actors and the mean, weighted by the actor's similarity - that is, a covariance weighted by the closeness of actors. This sum is taken in ratio to variance in the scores of all actors from the mean. The resulting measure, like the correlation coefficient, is a ratio of covariance to variance, and has a conventional interpretation.

Permutation trials are used to create a sampling distribution. Across many (in our example 1,000) trials, scores on the attribute (expenditure, in this case) are randomly assigned to actors, and the Moran statistic calculated. In these random trials, the average observed Moran statistic is -0.043, with a standard deviation of .073. The difference between what we observe (-0.119) and what is predicted by random association (-0.043) is small relative to sampling variability. In fact, 17.4% of all samples from random data showed correlations at least this big -- far more than the conventional 5% acceptable error rate.

The Geary measure of correlation is calculated and interpreted somewhat differently. Results are shown in Figure 18.24 for the association of expenditure levels by network adjacency.



```
Proximities:                  C:\Documents and Settings\
Attribute(s):                 CA_3_group Col 1
Method:                       Geary
# of Permutations:            1000
Random seed:                  592

NOTE: Smaller values indicate positive autocorrelation.
      A value of 1.0 indicates perfect independence.

         Autocorrelation:        2.137
           Significance:         0.026

     Permutation average:        1.004
          Standard error:        0.613
     Proportion as large:        0.026
     Proportion as small:        0.974
```

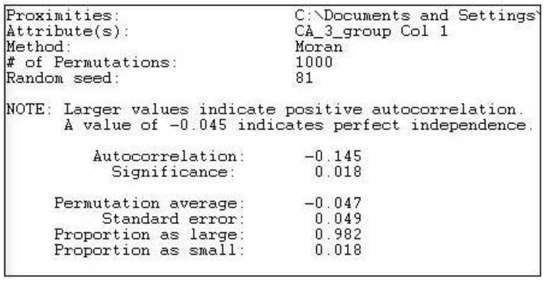
Figure 18.24: *Geary autocorrelation of expenditure levels by political donors with network adjacency*

The Geary statistic has a value of 1.0 when there is no association. Values less than 1.0 indicate a positive association (somewhat confusingly), values greater than 1.0 indicate a negative association. Our calculated value of 2.137 indicates negative autocorrelation, just as the Moran statistic did. Unlike the Moran statistic though, the Geary statistic suggests that the difference of our result from the average of 1,000 random trials (1.004) is statistically significant (p = 0.026).

The Geary statistic is sometimes described in the geo-statistics literature as being more sensitive to "local" differences than to "global" differences. The Geary C statistic is constructed by examining the differences between the scores of each pair of actors, and weighting this by their adjacency. The Moran statistic is constructed by looking at differences between each actor's score and the mean, and weighting the cross-products. The difference in approach means that the Geary statistic is more focused on how different members of each pair are from each other - a "local" difference; the Moran statistic is focused more on how the similar or dissimilar each pair are to the overall average -- a "global" difference.

In data where the "landscape" of values displays a lot of variation, and non-normal distribution, the two measures are likely to give somewhat different impressions about the effects of network adjacency on similarity of attributes. As always, it's not that one is "right" and the other "wrong." It's always best to compute both, unless you have strong theoretical priors that suggest that one is superior for a particular purpose.

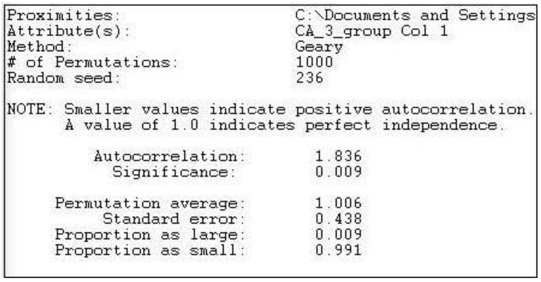
Figures 18.25 and 18.26 repeat the exercise above, but with one difference. In these two examples, we measure the closeness of two actors in the network on a continuous scale. Here, we've used the net number of campaigns on which each pair of actors were in the same coalition as a measure of closeness. Other measures, like geodesic distances might be more commonly used for true network data (rather than a network inferred from affiliation).



```
Proximities:                  C:\Documents and Settings\
Attribute(s):                 CA_3_group Col 1
Method:                       Moran
# of Permutations:            1000
Random seed:                  81

NOTE: Larger values indicate positive autocorrelation.
      A value of -0.045 indicates perfect independence.

         Autocorrelation:       -0.145
           Significance:         0.018

     Permutation average:       -0.047
          Standard error:        0.049
     Proportion as large:        0.982
     Proportion as small:        0.018
```

Figure 18.25: *Moran autocorrelation of expenditure levels by political donors with network closeness*

Using a continuous measure of network closeness (instead of adjacency) we might expect a stronger correlation. The Moran measure is now -0.145 (compared to -0.119), and is significant at p = 0.018. There is a small, but significant tendency for actors who are "close" allies to give different amounts of money than two randomly chosen actors -- a negative network autocorrelation.



```
Proximities:                      C:\Documents and Settings
Attribute(s):                     CA_3_group Col 1
Method:                           Geary
# of Permutations:                1000
Random seed:                      236

NOTE: Smaller values indicate positive autocorrelation.
      A value of 1.0 indicates perfect independence.

         Autocorrelation:           1.836
           Significance:            0.009

     Permutation average:           1.006
          Standard error:           0.438
     Proportion as large:           0.009
     Proportion as small:           0.991
```

Figure 18.26: *Geary autocorrelation of expenditure levels by political donors with network closeness*

The Geary measure has become slightly smaller in size (1.836 versus 2.137) using a continuous measure of network distance. The result also indicates a negative autocorrelation, and one that would rarely occur by chance if there truly was no association between network distance and expenditure.

The Probability of a Dyadic Tie: Leinhardt's P1

The approaches that we've been examining in this section look at the relationship between actor's attributes and their location in a network. Before closing our discussion of how statistical analysis has been applied to network data, we need to look at one approach that examines how ties between pairs of actors relate to particularly important relational attributes of the actors, and to a more global feature of the graph.

For any pair of actors in a directed graph, there are three possible relationships: no ties, an asymmetric tie, or a reciprocated tie. *Network>P1* is a regression-like approach that seeks to predict the probability of each of these kinds of relationships for each pair of actors. This differs a bit from the approaches that we've examined so far which seek to predict either the presence/absence of a tie, or the strength of a tie.

The P1 model (and its newer successor the P* model), seek to predict the dyadic relations among actor pairs using key relational attributes of each actor, and of the graph as a whole. This differs from most of the approaches that we've seen above, which focus on actor's individual or relational attributes, but do not include overall structural features of the graph (at least not explicitly).

The P1 model consists of three prediction equations, designed to predict the probability of a mutual (i.e. reciprocated) relation (m_{ij}), an asymmetric relation (a_{ij}), or a null relation (n_{ij}) between actors. The equations, as stated by the authors of UCINET are:

$$m_{ij} = \lambda_{ij} e^{\rho + 2\theta + \alpha_i + \alpha_j + \hat{a}_i + \hat{a}_j} \tag{18.5.1}$$

$$a_{ij} = \lambda_{ij} e^{\theta + \alpha_i + \beta_j} \tag{18.5.2}$$

$$n_{ij} = \lambda_{ij} \tag{18.5.3}$$

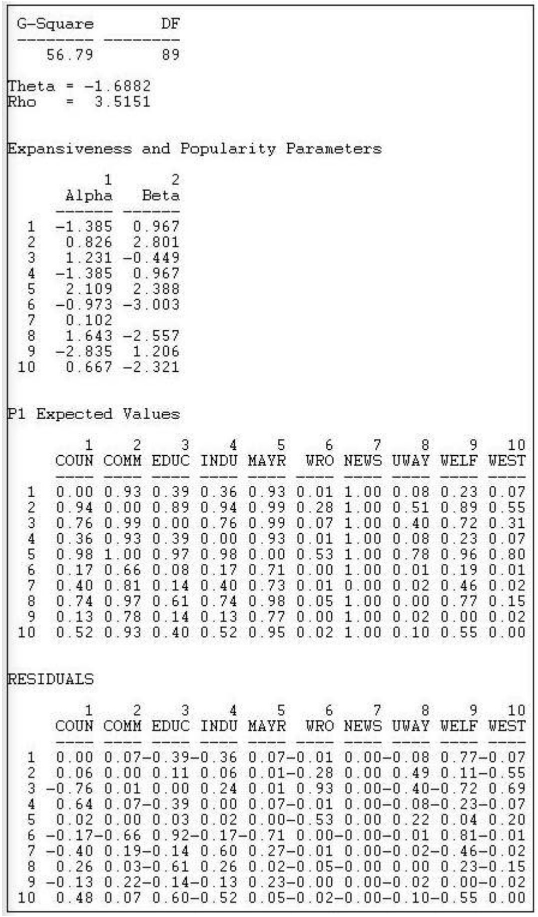
The first equation says that the probability of a reciprocated tie between two actors is a function of the out-degree (or "expansiveness") of each actor: alpha_i and alpha_j. It is also a function of the overall density of the network (theta). It is also a function of the global tendency in the whole network toward reciprocity (rho). The equation also contains scaling constants for each actor in the pair (a_i and a_j), as well as a global scaling parameter (lambda).

The second equation describes the probability that two actors will be connected with an asymmetric relation. This probability is a function of the overall network density (theta), and the propensity of one actor of the pair to send ties (expansiveness, or alpha), and the propensity of the other actor to receive ties ("attractiveness" or beta).

The probability of a null relation (no tie) between two actors is a "residual." That is, if ties are not mutual or asymmetric, they must be null. Only the scaling constant "lambda," and no causal parameters enter the third equation.

The core idea here is that we try to understand the relations between pairs of actors as functions of individual relational attributes (individual's tendencies to send ties, and to receive them) as well as key features of the graph in which the two actors are embedded (the overall density and overall tendency towards reciprocity). More recent versions of the model (P*, P2) include additional global features of the graph such as tendencies toward transitivity and the variance across actors in the propensity to send and receive ties.

Figure 18.27 shows the results of fitting the P1 model to the Knoke binary information network.



```
 G-Square        DF
 --------- ---------
     56.79        89

Theta = -1.6882
Rho   =  3.5151


Expansiveness and Popularity Parameters

          1      2
      Alpha   Beta
     ------ ------
  1  -1.385  0.967
  2   0.826  2.801
  3   1.231 -0.449
  4  -1.385  0.967
  5   2.109  2.388
  6  -0.973 -3.003
  7   0.102
  8   1.643 -2.557
  9  -2.835  1.206
 10   0.667 -2.321


P1 Expected Values

         1    2    3    4    5    6    7    8    9   10
      COUN COMM EDUC INDU MAYR  WRO NEWS UWAY WELF WEST
      ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  1   0.00 0.93 0.39 0.36 0.93 0.01 1.00 0.08 0.23 0.07
  2   0.94 0.00 0.89 0.94 0.99 0.28 1.00 0.51 0.89 0.55
  3   0.76 0.99 0.00 0.76 0.99 0.07 1.00 0.40 0.72 0.31
  4   0.36 0.93 0.39 0.00 0.93 0.01 1.00 0.08 0.23 0.07
  5   0.98 1.00 0.97 0.98 0.00 0.53 1.00 0.78 0.96 0.80
  6   0.17 0.66 0.08 0.17 0.71 0.00 1.00 0.01 0.19 0.01
  7   0.40 0.81 0.14 0.40 0.73 0.01 0.00 0.02 0.46 0.02
  8   0.74 0.97 0.61 0.74 0.98 0.05 1.00 0.00 0.77 0.15
  9   0.13 0.78 0.14 0.13 0.77 0.00 1.00 0.02 0.00 0.02
 10   0.52 0.93 0.40 0.52 0.95 0.02 1.00 0.10 0.55 0.00


RESIDUALS

         1    2    3    4    5    6    7    8    9   10
      COUN COMM EDUC INDU MAYR  WRO NEWS UWAY WELF WEST
      ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  1   0.00 0.07-0.39-0.36 0.07-0.01 0.00-0.08 0.77-0.07
  2   0.06 0.00 0.11 0.06 0.01-0.28 0.00 0.49 0.11-0.55
  3  -0.76 0.01 0.00 0.24 0.01 0.93 0.00-0.40-0.72 0.69
  4   0.64 0.07-0.39 0.00 0.07-0.01 0.00-0.08-0.23-0.07
  5   0.02 0.00 0.03 0.02 0.00-0.53 0.00 0.22 0.04 0.20
  6  -0.17-0.66 0.92-0.17-0.71 0.00-0.00-0.01 0.81-0.01
  7  -0.40 0.19-0.14 0.60 0.27-0.01 0.00-0.02-0.46-0.02
  8   0.26 0.03-0.61 0.26 0.02-0.05-0.00 0.00 0.23-0.15
  9  -0.13 0.22-0.14-0.13 0.23-0.00 0.00-0.02 0.00-0.02
 10   0.48 0.07 0.60-0.52 0.05-0.02-0.00-0.10-0.55 0.00
```

Figure 18.27: *Results of P1 analysis of Knoke information network*

The technical aspects of the estimation of the P1 model are complicated, and maximum likelihood methods are used. A G-square (likelihood ratio chi-square) badness of fit statistic is provided, but has no direct interpretation or significance test.

Two descriptive parameters for global network properties are given:

- θ = -1.6882 refers to the effect of the global density of the network on the probability of reciprocated or asymmetric ties between pairs of actors.
- ρ = 3.5151 refers to the effect of the overall amount of reciprocity in the global network on the probability of a reciprocated tie between any pair of actors.

Two descriptive parameters are given for each actor (these are estimated across all of the pair-wise relations of each actor):

α ("expansiveness") refers to the effect of each actor's out-degree on the probability that they will have reciprocated or asymmetric ties with other actors. We see, for example, that the Mayor (actor 5) is a relatively "expansive" actor.

β ("attractiveness") refers to the effect of each actor's in-degree on the probability that they will have a reciprocated or asymmetric relation with other actors. We see here, for example, that the welfare rights organization (actor 6) is very likely to be shunned.

Using the equations, it is possible to predict the probability of each directed tie based on the model's parameters. These are shown as the "P1 expected values." For example, the model predicts a 93% chance of a tie from actor 1 to actor 2.

The final panel of the output shows the difference between the ties that actually exist, and the predictions of the model. The model predicts the tie from actor 1 to actor 2 quite well (residual = .07), but it does a poor job of predicting the relation from actor 1 to actor 9 (residual = 0.77).

The residuals important because they suggest places where other features of the graph or individuals may be relevant to understanding particular dyads, or where the ties between two actors is well accounted for by basic "demographics" of the network. Which actors are likely to have ties that are not predicted by the parameters of the model can also be shown in a dendogram, as in Figure 18.28.

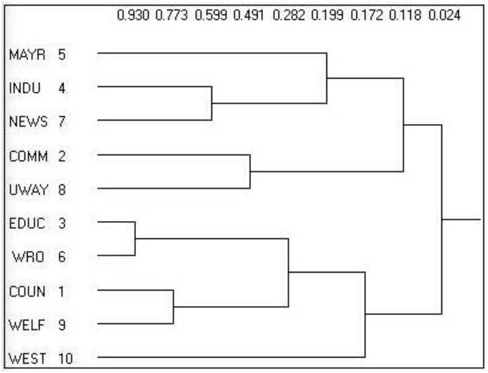


Figure 18.28: *Diagram of P1 clustering of Knoke information network*

Here we see that, for example, that actors 3 and 6 are much more likely to have ties than the P1 model predicts.

18.S: Some Statistical Tools (Summary)

In this chapter we've taken a look at some of the most basic and common approaches to applying statistical analysis to the attributes of actors embedded in networks, the relations among these actors, and the similarities between multiple relational networks connecting the same actors. We've covered a lot of ground. But, there is still a good bit more, as the application of statistical modeling to network data is one of the "leading edges" of the field of social (and other) network analyses.

There are two main reasons for the interest in applying statistics to what was, originally, deterministic graph theory from mathematics. First, for very large networks, methods for finding and describing the distributions of network features provide important tools for understanding the likely patterns of behavior of the whole network and the actors embedded in it. Second, we have increasingly come to realize that the relations we see among actors in a network at a point in time are best seen as probabilistic ("stochastic") outcomes of underlying processes of evolution of networks, and probabilistic actions of actors embedded in those networks. Statistical methods provide ways of dealing with description and hypothesis testing that take this uncertainty into account.

We've reviewed methods for examining relations between two (or more) graphs involving the same actors. These tools are particularly useful for trying to understand multi-plex relations, and for testing hypotheses about how the pattern of relations in one whole network relate to the pattern of relations in another.

We've also looked at tools that deal individual nodes. These tools allow us to examine hypotheses about the relational and non-relational attributes of actors, and to draw correct inferences about relations between variables when the observations (actors) are not independent.

And, we've taken a look at a variety of approaches that relate attributes of actors to their positions in networks. Much of the focus here is on how attributes may pattern relations (e.g. homophily), or how network closeness of distance may affect similarity of attributes (or vice versa).

Taken together, the marriage of statistics and mathematics in social network analysis has already produced some very useful ways of looking at patterns of social relations. It is likely that this interface will be one of the areas of most rapid development in the field of social network methods in the coming years.

Index

Glossary

Sample Word 1 | Sample Definition 1

Detailed Licensing

Overview

Title: Book: Introduction to Social Network Methods (Hanneman)

Webpages: 126

All licenses found:

- Undeclared: 100% (126 pages)

By Page

- Book: Introduction to Social Network Methods (Hanneman) - *Undeclared*
  - Front Matter - *Undeclared*
    - TitlePage - *Undeclared*
    - InfoPage - *Undeclared*
    - Table of Contents - *Undeclared*
    - Licensing - *Undeclared*
  - 1: Social Network Data - *Undeclared*
    - 1.1: Introduction - What's different about social network data? - *Undeclared*
    - 1.2: Nodes - *Undeclared*
    - 1.3: Relations - *Undeclared*
    - 1.4: Scales of Measurement - *Undeclared*
    - 1.5: A note on Statistics and Social Network Data - *Undeclared*
  - 2: Why Formal Methods? - *Undeclared*
    - 2.1: Introduction - *Undeclared*
    - 2.2: Efficiency - *Undeclared*
    - 2.3: Using computers - *Undeclared*
    - 2.4: Seeing patterns - *Undeclared*
    - 2.5: Summary - *Undeclared*
  - 3: Using Graphs to Represent Social Relations - *Undeclared*
    - 3.E: Using graphs to represent social relations (Exercises) - *Undeclared*
    - 3.1: Introduction - Representing networks with graphs - *Undeclared*
    - 3.2: Graphs and Sociograms - *Undeclared*
    - 3.3: Kinds of Graphs - *Undeclared*
    - 3.4: Summary - *Undeclared*
  - 4: Working with Netdraw to Visualize Graphs - *Undeclared*
    - 4.1: Introduction - A picture is worth... - *Undeclared*
    - 4.2: Node Attributes - *Undeclared*
    - 4.3: Relation Properties - *Undeclared*
    - 4.4: Location, location, location - *Undeclared*
    - 4.5: Highlighting parts of the network - *Undeclared*
    - 4.6: A few hints on data handling with NetDraw - *Undeclared*
  - 5: Using Matrices to Represent Social Relations - *Undeclared*
    - 5.E: Using matrices to represent social relations (Exercises) - *Undeclared*
    - 5.S: Using Matrices to Represent Social Relations (Summary) - *Undeclared*
    - 5.1: What is a matrix? - *Undeclared*
    - 5.2: The "adjacency" matrix - *Undeclared*
    - 5.3: Matrix permutation, blocks, and images - *Undeclared*
    - 5.4: Doing mathematical operations on matrices - *Undeclared*
  - 6: Working with Network Data - *Undeclared*
    - 6.S: Working with Network Data (Summary) - *Undeclared*
    - 6.1: Manipulating Network Data Structures - *Undeclared*
    - 6.2: Making UCINET Datasets - *Undeclared*
    - 6.3: Transforming Data Values - *Undeclared*
    - 6.4: File Handling Tools - *Undeclared*
    - 6.5: Selecting Subsets of the Data - *Undeclared*
    - 6.6: Making New Kinds of Graphs from Existing Graphs - *Undeclared*
  - 7: Connection - *Undeclared*
    - 7.E: Connection (Exercises) - *Undeclared*
    - 7.S: Connection (Summary) - *Undeclared*
    - 7.1: Networks and Actors - *Undeclared*
    - 7.2: Connection - *Undeclared*
    - 7.3: Distance - *Undeclared*
  - 8: Embedding - *Undeclared*
    - 8.1: Introduction - *Undeclared*
    - 8.2: Density - *Undeclared*
    - 8.3: Reciprocity - *Undeclared*
    - 8.4: Transitivity - *Undeclared*
    - 8.5: Clustering - *Undeclared*
    - 8.6: Group-External and Group-Internal Ties - *Undeclared*
    - 8.7: Krackhardt's Graph Theoretical Dimensions of Hierarchy - *Undeclared*