2.2: Minterms and MATLAB Calculations

The concepts and procedures in this unit play a significant role in many aspects of the analysis of probability topics and in the use of MATLAB throughout this work.

Minterm vectors and MATLAB

The systematic formulation in the previous module Minterms shows that each Boolean combination, as a union of minterms, can be designated by a vector of zero-one coefficients. A coefficient one in the ith position (numbering from zero) indicates the inclusion of minterm M_i in the union. We formulate this pattern carefully below and show how MATLAB logical operations may be utilized in problem setup and solution.

Suppose E is a Boolean combination of A, B, C. Then, by the minterm expansion,

$$E = \bigvee_{J_E} M_i$$

where M_i is the ith minterm and J_E is the set of indices for those M_i included in E. For example, consider

$$E = A(B \cup C^c) \cup A^c(B \cup C^c)^c = M_1 \bigvee M_4 \bigvee M_6 \bigvee M_7 = M(1, 4, 6, 7)$$

$$F = A^c B^c \cup AC = M_0 \bigvee M_1 \bigvee M_5 \bigvee M_7 = M(0, 1, 5, 7)$$

We may designate each set by a pattern of zeros and ones $(e_0, e_1, \cdots, e_7)$. The ones indicate which minterms are present in the set. In the pattern for set E, minterm M_i is included in E iff $e_i = 1$. This is, in effect, another arrangement of the minterm map. In this form, it is convenient to view the pattern as a *minterm vector*, which may be represented by a row matrix or row vector [$e_0, e_1, \cdots, e_7$]. We find it convenient to use *the same symbol for the name of the event and for the minterm vector or matrix representing it*. Thus, for the examples above,

$$E \sim [0\ 1\ 0\ 0\ 1\ 0\ 1\ 1] \text{ and } F \sim [1\ 1\ 0\ 0\ 0\ 1\ 0\ 1]$$

It should be apparent that this formalization can be extended to sets generated by any finite class.

Minterm vectors for Boolean combinations

If E and F are combinations of n generating sets, then each is represented by a unique minterm vector of length 2^n. In the treatment in the module Minterms, we determine the minterm vector with the aid of a minterm map. We wish to develop a systematic way to determine these vectors.

As a first step, we suppose we have minterm vectors for E and F and want to obtain the minterm vector of Boolean combinations of these.

1. The minterm expansion for $E \cup F$ has all the minterms in either set. This means the jth element of the vector for $E \cup F$ is the *maximum* of the jth elements for the two vectors.
2. The minterm expansion for $E \cap F$ has only those minterms in both sets. This means the jth element of the vector for $E \cap F$ is the *minimum* of the jth elements for the two vectors.
3. The minterm expansion for E^c has only those minterms not in the expansion for E. This means the vector for E^c has zeros and ones interchanged. The jth element of E^c is one iff the corresponding element of E is zero.

We illustrate for the case of the two combinations E and F of three generating sets, considered above

$$E = A(B \cup C^c) \cup A^c(B \cup C^c)^c \cup A^c(B \cup C^c)^c \sim \quad [0\ 1\ 0\ 0\ 1\ 0\ 1\ 1] \text{ and } F = A^c B^c \cup AC \sim\ [1\ 1\ 0\ 0\ 0\ 1\ 0\ 1]$$

Then

$$E \cup F \sim\ [1\ 1\ 0\ 0\ 1\ 1\ 1\ 1],\ E \cap F \sim\ [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1], \text{ and } E^c \sim [1\ 0\ 1\ 1\ 0\ 1\ 0\ 0]$$

MATLAB logical operations

MATLAB logical operations on zero-one matrices provide a convenient way of handling Boolean combinations of minterm vectors represented as matrices. For two *zero-one* matrices E, F of the same size

- $E|F$ is the matrix obtained by taking the maximum element in each place.
- $E\&F$ is the matrix obtained by taking the minimum element in each place.
- E^c is the matrix obtained by interchanging one and zero in each place in E.

Thus, if E, F are minterm vectors for sets by the same name, then $E|F$ is the minterm vector for $E \cup F$, $E\&F$ is the minterm vector for $E \cap F$, and $E = 1 - E$ is the minterm vector for E^c.

This suggests a general approach to determining minterm vectors for Boolean combinations.

Start with minterm vectors for the generating sets.
Use MATLAB logical operations to obtain the minterm vector for any Boolean combination.

Suppose, for example, the class of generating sets is $\{A, B, C\}$. Then the minterm vectors for A, B, and C, respectively, are

$$A = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]\ B = [0\ 0\ 1\ 1\ 0\ 0\ 1\ 1]\ C = [0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

If $E = AB \cup C^c$, then the logical combination $E = (A\&B) \mid C$ of the matrices yields $E = [1\ 0\ 1\ 0\ 1\ 0\ 1\ 1]$.

MATLAB implementation

A key step in the procedure just outlined is to obtain the minterm vectors for the generating elements $\{A, B, C\}$. We have an m-function to provide such fundamental vectors. For example to produce the second minterm vector for the family (i.e., the minterm vector for B), the basic zero-one pattern `0 0 1 1` is replicated twice to give

```
  0     0     1     1     0     0     1     1
```

The function *minterm(n,k)* generates the *k*th minterm vector for a class of *n* generating sets.

minterms for the class {a, b, c}.

```
>> A = minterm(3,1)
A =  0     0     0     0     1     1     1     1
>> B = minterm(3,2)
B =  0     0     1     1     0     0     1     1
>> C = minterm(3,3)
C =  0     1     0     1     0     1     0     1
```

minterm patterns for the boolean combinations

$F = AB \cup B^cC$ $G = A \cup A^cC$

```
F = (A&B)|(~B&C)
F =  0     1     0     0     0     1     1     1
>> G = A|(~A&C)
G =  0     1     0     1     1     1     1     1
>> JF = find(F)-1           % Use of find to determine index set for F
JF =   1     5     6     7  % Shows F = M(1, 5, 6, 7)
```

These basic minterm patterns are useful not only for Boolean combinations of events but also for many aspects of the analysis of those random variables which take on only a finite number of values.

Zero-one arrays in MATLAB

The treatment above hides the fact that a rectangular array of zeros and ones can have two quite different meanings and functions in MATLAB.

A *numerical matrix* (or vector) subject to the usual operations on matrices..
A *logical array* whose elements are combined by a. Logical operators to give new logical arrays; b. Array operations (element by element) to give numerical matrices; c. Array operations with numerical matrices to give numerical results.

Some simple examples will illustrate the principal properties.

```
>>> A = minterm(3,1);
>> B = minterm(3,2);
>> C = minterm(3,3);
>> F = (A&B)|(~B&C)
F =  0     1     0     0     0     1     1     1
>> G = A|(~A&C)
G =  0     1     0     1     1     1     1     1
>> islogical(A)       % Test for logical array
ans =    0
>> islogical(F)
ans =   1
>> m = max(A,B)       % A matrix operation
m =   0     0     1     1     1     1     1     1
>> islogical(m)
ans =   0
>> m1 = A|B           % A logical operation
m1 =   0     0     1     1     1     1     1     1
>> islogical(m1)
ans = 1
>> a = logical(A)      % Converts 0-1 matrix into logical array
a =   0     0     0     0     1     1     1     1
>> b = logical(B)
>> m2 = a|b
m2 =   0     0     1     1     1     1     1     1
>> p = dot(A,B)        % Equivalently, p = A*B'
p =  2
>> p1 = total(A.*b)
p1 =  2
>> p3 = total(A.*B)
p3 =  2
>> p4 = a*b'           % Cannot use matrix operations on logical arrays
??? Error using ==> mtimes   % MATLAB error signal
Logical inputs must be scalar.
```

Often it is desirable to have a table of the generating minterm vectors. Use of the function minterm in a simple "for loop" yields the following m-function.

The function *mintable(n)* Generates a table of minterm vectors for n generating sets.

mintable for three variables

```
>> M3 = mintable(3)
M3 = 0     0     0     0     1     1     1     1
     0     0     1     1     0     0     1     1
     0     1     0     1     0     1     0     1
```

As an application of mintable, consider the problem of determining the probability of k of n events. If $\{A_i : 1 \le i \le n\}$ is any finite class of events, the event that exactly k of these events occur on a trial can be characterized simply in terms of the minterm

expansion. The event A_{kn} that exactly k occur is given by

$$A_{kn} = the\ disjoint\ union\ of\ those\ minterms\ with\ exactly\ k\ positions\ uncomplemented$$

In the matrix `M = mintable(n)` these are the minterms corresponding to columns with exactly k ones. The event BknBkn that k or more occur is given by

$$B_{kn} = \bigvee_{r=k}^{n} A_{rn}$$

If we have the minterm probabilities, it is easy to pick out the appropriate minterms and combine the probabilities. The following example in the case of three variables illustrates the procedure.

the software survey (continued)

In the software survey problem, the minterm probabilities are

$$pm = [0\ 0.05\ 0.10\ 0.05\ 0.20\ 0.10\ 0.40\ 0.10]$$

where $A =$ event has word processor, $B =$ event has spread sheet, $C =$ event has a data base program. It is desired to get the probability an individual selected has k of these, $k = 0, 1, 2, 3$.

Solution

We form a mintable for three variables. We count the number of "successes" corresponding to each minterm by using the MATLAB function sum, which gives the sum of each column. In this case, it would be easy to determine each distinct value and add the probabilities on the minterms which yield this value. For more complicated cases, we have an m-function called *csort* (for sort and consolidate) to perform this operation.

```
>> pm = 0.01*[0 5 10 5 20 10 40 10];
>> M = mintable(3)
M =
0     0     0     0     1     1     1     1
0     0     1     1     0     0     1     1
0     1     0     1     0     1     0     1
>> T = sum(M)                                  % Column sums give number
T =  0     1     1     2     1     2     2     3 % of successes on each
>> [k,pk] = csort(T,pm);                       % minterm, determines
                                               % distinct values in T and
>> disp([k;pk]')                               % consolidates probabilities
0         0
1.0000    0.3500
2.0000    0.5500
3.0000    0.1000
```

For three variables, it is easy enough to identify the various combinations "by eye" and make the combinations. For a larger number of variables, however, this may become tedious. The approach is much more useful in the case of Independent Events, because of the ease of determining the minterm probabilities.

Minvec procedures

Use of the tilde $\sim$ to indicate the complement of an event is often awkward. It is customary to indicate the complement of an event E by E^c. In MATLAB, we cannot indicate the superscript, so we indicate the complement by E^c instead of $\sim E$. To facilitate writing combinations, we have a family of *minvec* procedures (minvec3, minvec4, ..., minvec10) to expedite expressing Boolean combinations of $n = 3, 4, 5, \cdots, 10$ sets. These generate and name the minterm vector for each generating set and its complement.

boolean combinations using minvec3

We wish to generate a matrix whose rows are the minterm vectors for $\Omega = A \cup A^c, A, AB, ABC, C,$ and A^cC^c, respectively.

```
>> minvec3                                  % Call for the setup procedure
Variables are A, B, C, Ac, Bc, Cc
They may be renamed, if desired
>> V = [A|Ac; A; A&B; A&B&C; C; Ac&Cc];  % Logical combinations (one per
                                          % row) yield logical vectors
>> disp(V)
1     1     1     1     1     1     1     1   % Mixed logical and
0     0     0     0     1     1     1     1   % numerical vectors
0     0     0     0     0     0     1     1
0     0     0     0     0     0     0     1
0     1     0     1     0     1     0     1
1     0     1     0     0     0     0     0
```

Minterm probabilities and Boolean combination

If we have the probability of every minterm generated by a finite class, we can determine the probability of any Boolean combination of the members of the class. When we know the minterm expansion or, equivalently, the minterm vector, we simply pick out the probabilities corresponding to the minterms in the expansion and add them. In the following example, we do this "by hand" then show how to do it with MATLAB .

Consider $E = A(B \cup C^c) \cup A^c(B \cup C^c)^c$ and $F = A^cB^c \cup AC$ of the example above, and suppose the respective minterm probabilites are

$p_0 = 0.21, p_1 = 0.06, p_2 = 0.29, p_3 = 0.11, p_4 = 0.09, p_5 = 0.03, p_6 = 0.14, p_7 = 0.07$

Use of a minterm map shows $E = M(1, 4, 6, 7)$ and $F = M(0, 1, 5, 7)$. so that

$P(E) = p_1 + p_4 + p_6 + p_7 = p(1, 4, 6, 7) = 0.36$ and $P(F) = p(0, 1, 5, 7) = 0.37$

This is easily handled in MATLAB.

- Use minvec3 to set the generating minterm vectors.
- Use *logical* matrix operations

  $E = (A\&(B|Cc))|(Ac\&((B|Cc)))$ and $F = (Ac\&Bc)|(A\&C)$

  to obtain the (logical) minterm vectors for E and F
- If pm is the matrix of minterm probabilities, perform the *algebraic* dot product or scalar product of the pmpm matrix and the minterm vector for the combination. This can be called for by the MATLAB commands PE = E*pm' and PF = F*pm'.

The following is a transcript of the MATLAB operations.

```
>> minvec3                 % Call for the setup procedure
Variables are A, B, C, Ac, Bc, Cc
They may be renamed, if desired.
>> E  = (A&(B|Cc))|(Ac&~(B|Cc));
>> F  = (Ac&Bc)|(A&C);
>> pm = 0.01*[21 6 29 11 9 3 14 7];
>> PE = E*pm'              % Picks out and adds the minterm probabilities
PE =  0.3600
```

```
>> PF = F*pm'
PF =  0.3700
```

solution of the software survey problem

We set up the matrix equations with the use of MATLAB and solve for the minterm probabilities. From these, we may solve for the desired "target" probabilities.

```
>> minvec3
Variables are A, B, C, Ac, Bc, Cc
They may be renamed, if desired.
Data vector combinations are:
>> DV = [A|Ac; A; B; C; A&B&C; Ac&Bc; (A&B)|(A&C)|(B&C); (A&Bc&C) - 2*(Ac&B&C)]
DV =
1     1     1     1     1     1     1     1    % Data mixed numerical
0     0     0     0     1     1     1     1    % and logical vectors
0     0     1     1     0     0     1     1
0     1     0     1     0     1     0     1
0     0     0     0     0     0     0     1
1     1     0     0     0     0     0     0
0     0     0     1     0     1     1     1
0     0     0    -2     0     1     0     0
>> DP = [1 0.8 0.65 0.3 0.1 0.05 0.65 0];  % Corresponding data probabilities
>> pm = DV\DP'                              % Solution for minterm probabilities
pm =
-0.0000                                          % Roundoff -3.5 x 10-17
0.0500
0.1000
0.0500
0.2000
0.1000
0.4000
0.1000
>> TV = [(A&B&Cc)|(A&Bc&C)|(Ac&B&C); Ac&Bc&C]       % Target combinations
TV =
0     0     0     1     0     1     1     0    % Target vectors
0     1     0     0     0     0     0     0
>> PV = TV*pm                           % Solution for target probabilities
PV =
0.5500                              % Target probabilities
0.0500
```

An alternate approach

The previous procedure first obtained all minterm probabilities, then used these to determine probabilities for the target combinations. The following procedure does not require calculation of the minterm probabilities. Sometimes the data are not sufficient to calculate all minterm probabilities, yet are sufficient to allow determination of the target probabilities.

Suppose the data minterm vectors are linearly independent, and the target minterm vectors are linearly dependent upon the data vectors (i.e., the target vectors can be expressed as linear combinations of the data vectors). Now each target probability is the same linear combination of the data probabilities. To determine the linear combinations, solve the matrix equation

$TV = CT * DV \; which \; has \; the \; MATLAB \; solution \; CT = TV/DV$

Then the matrix tp of target probabilites is given by $tp = CT * DP'$. Continuing the MATLAB procedures above, we have:

```
>> CT = TV/DV;
```

```
>> tp = CT*DP'
```

```
tp = 0.5500
         0.0500
```

The procedure mincalc

The procedure *mincalc* performs calculations as in the preceding examples. The *refinements* consist of determining consistency and computability of various individual minterm probabilities and target probilities. The consistency check is principally for negative minterm probabilities. The computability tests are tests for linear independence by means of calculation of ranks of various matrices. The procedure picks out the computable minterm probabilities and the computable target probabilities and calculates them.

To utilize the procedure, the problem must be formulated appropriately and precisely, as follows:

Use the MATLAB program *minvecq* to set minterm vectors for each of *q* basic events.
Data consist of Boolean combinations of the basic events and the respective probabilities of these combinations. These are organized into two matrices:

- The *data vector matrix* DV has the data Boolean combinations– one on each row. MATLAB translates each row into the minterm vector for the corresponding Boolean combination. The first entry (on the first row) is A | Ac (for $A \bigvee A^c$), which is the whole space. Its minterm vector consists of a row of ones.
- The *data probability matrix* DP is a row matrix of the data probabilities. The first entry is one, the probability of the whole space.

The *objective* is to determine the probability of various *target* Boolean combinations. These are put into the *target vector matrix* TV, one on each row. MATLAB produces the minterm vector for each corresponding target Boolean combination.

Computational note. In mincalc, it is necessary to turn the arrays DV and TV consisting of zero-one patterns into zero-one matrices. This is accomplished for DV by the operation `DV = ones(size(DV)).*DV`. and similarly for TV. Both the original and the transformed matrices have the same zero-one pattern, but MATLAB interprets them differently.

Usual case

Śuppose the data minterm vectors are linearly independent and the target vectors are each linearly dependent on the data minterm vectors. Then each target minterm vector is expressible as a linear combination of data minterm vectors. Thus, there is a matrix CT such that $TV = CT * DV$. MATLAB solves this with the command $CT = TV/DV$ *The target probabilities are the same linear combinations of the data probabilities*. These are obtained by the MATLAB operation $tp = DP * CT'$.

Cautionary notes

The program mincalc depends upon the provision in MATLAB for solving equations when less than full data are available (based on the singular value decomposition). There are several situations which should be dealt with as special cases. It is usually a good idea to check results by hand to determine whether they are consistent with data. The checking by hand is usually much easier than obtaining the solution unaided, so that use of MATLAB is advantageous even in questionable cases.

The Zero Problem. If the total probability of a group of minterms is zero, then it follows that the probability of each minterm in the group is zero. However, if mincalc does not have enough information to calculate the separate minterm probabilities in

the case they are not zero, it will not pick up in the zero case the fact that the separate minterm probabilities are zero. It simply considers these minterm probabilities not computable.

Linear dependence. In the case of linear dependence, the operation called for by the command CT = TV/DV may not be able to solve the equations. The matrix may be singular, or it may not be able to decide which of the redundant data equations to use. Should it provide a solution, the result should be checked with the aid of a minterm map.

Consistency check. Since the consistency check is for negative minterms, if there are not enough data to calculate the minterm probabilities, there is no simple check on the consistency. Sometimes the probability of a target vector included in another vector will actually exceed what should be the larger probability. Without considerable checking, it may be difficult to determine consistency.

In a few unusual cases, the command CT = TV/DV does not operate appropriately, even though the data should be adequate for the problem at hand. Apparently the approximation process does not converge.

MATLAB Solutions for examples using mincalc

software survey

```
% file mcalc01   Data for software survey
minvec3;
DV = [A|Ac; A; B; C; A&B&C; Ac&Bc; (A&B)|(A&C)|(B&C); (A&Bc&C)  - 2*(Ac&B&C)];
DP = [1 0.8 0.65 0.3 0.1 0.05 0.65 0];
TV = [(A&B&Cc)|(A&Bc&C)|(Ac&B&C); Ac&Bc&C];
disp('Call for mincalc')
>> mcalc01         % Call for data
Call for mincalc   % Prompt supplied in the data file
>> mincalc
Data vectors are linearly independent
Computable target probabilities
1.0000    0.5500
2.0000    0.0500
The number of minterms is 8
The number of available minterms is 8
Available minterm probabilities are in vector pma
To view available minterm probabilities, call for PMA
>> disp(PMA)       % Optional call for minterm probabilities
0         0
1.0000    0.0500
2.0000    0.1000
3.0000    0.0500
4.0000    0.2000
5.0000    0.1000
6.0000    0.4000
7.0000    0.1000
```

computer survey

```
% file mcalc02.m    Data for computer survey
minvec3
DV =   [A|Ac; A; B; C; A&B&C; A&C; (A&B)|(A&C)|(B&C); ...
2*(B&C) - (A&C)];
```

```
DP = 0.001*[1000 565 515 151 51 124 212 0];   TV = [A|B|C; Ac&Bc&C];
disp('Call for mincalc')
>> mcalc02
Call for mincalc
>> mincalc
Data vectors are linearly independent
Computable target probabilities
1.0000    0.9680
2.0000    0.0160
The number of minterms is 8
The number of available minterms is 8
Available minterm probabilities are in vector pma
To view available minterm probabilities, call for PMA
>> disp(PMA)
0         0.0320
1.0000    0.0160
2.0000    0.3760
3.0000    0.0110
4.0000    0.3640
5.0000    0.0730
6.0000    0.0770
7.0000    0.0510
```

```
% file mcalc03.m    Data for opinion survey
minvec4
DV = [A|Ac; A; B; C; D; A&(B|Cc)&Dc; A|((B&C)|Dc) ; Ac&B&Cc&D; ...
A&B&C&D; A&Bc&C; Ac&Bc&Cc&D; Ac&B&C; Ac&Bc&Dc; A&Cc; A&C&Dc; A&B&Cc&Dc];
DP =  0.001*[1000 200 500 300 700 55 520 200 15 30 195 120 120 ...
              140 25 20];
TV = [Ac&((B&Cc)|(Bc&C)); A|(B&Cc)];
disp('Call for mincalc')
>> mincalc03
Call for mincalc
>> mincalc
Data vectors are linearly independent
Computable target probabilities
1.0000    0.4000
2.0000    0.4800
The number of minterms is 16
The number of available minterms is 16
Available minterm probabilities are in vector pma
To view available minterm probabilities, call for PMA
>> disp(minmap(pma))    % Display arranged as on minterm map
0.0850    0.0800    0.0200    0.0200
0.1950    0.2000    0.0500    0.0500
```

```
0.0350    0.0350    0.0100    0.0150
0.0850    0.0850    0.0200    0.0150
```

The procedure mincalct

A useful modification, which we call *mincalct*, computes the available target probabilities, without checking and computing the minterm probabilities. This procedure assumes a data file similar to that for mincalc, except that it does not need the target matrix TV, since it prompts for target Boolean combination inputs. The procedure mincalct may be used after mincalc has performed its operations to calculate probabilities for additional target combinations.

(continued) Additional target datum for the opinion survey

Suppose mincalc has been applied to the data for the opinion survey and that it is desired to determine $P(AD \cup BD^c)$. It is not necessary to recalculate all the other quantities. We may simply use the procedure mincalct and input the desired Boolean combination at the prompt.

```
>> mincalct
Enter matrix of target Boolean combinations  (A&D)|(B&Dc)
Computable target probabilities
    1.0000    0.2850
```

Repeated calls for mcalct may be used to compute other target probabilities.