User's Guide to EMMIX – Version 1.3 1999

D. Peel, and G.J. McLachlan

Note: This program is available freely for **non-commercial** use only

Contents

1 Introduction

This document outlines the operation and the available options of the program EMMIX. Brief instructions on the form of the input and output files are also given.

The main purpose of the program is to fit a mixture model of multivariate normal or t-distributed components to a given data set. This is approached by using maximum likelihood via the EM algorithm of Dempster, Laird, and Rubin (1977); for a full examination of the EM algorithm and related topics, see McLachlan and Krishnan (1997). Many other features are also included, that were found to be of use when fitting mixture models.

2 Compilation

The version you have obtained consists of the files **EMMIX.f** and **EMMIX.max**. To compile the program, simply use a FORTRAN compiler. On a UNIX system this is done by simply typing,

```
f77 -o EMMIX EMMIX.f
```

Consult your relevant compiler manuals for other platforms.

2.1 Compatibility

The program was developed using a UNIX based compiler, although the program has been successfully compiled on a number of machines. In previous versions of EMMIX the main problem of incompatibility seemed to be the use of the inbuilt random number generator. This version of EMMIX uses the applied statistics random number generator. EMMIX implements a test of the generator at the start of the run, if this fails; ie. gives a zero, or repeats a number within the 1000 point test, then a warning message appears and the program will still run, but any features that utilise random numbers can not be used. To simplify matters all calls in the program to the random number generator are done via calling the function RANDNUM, which in turn calls the appropriate generator. So if required, the change to another generator should be a quick and simple modification.

Most non-ANSI extensions that were used in previous versions of EMMIX have been removed in this cross platform version, although as a result the input and output is not as aesthetically pleasing, but it is hoped the program will be easier to compile and run on different systems.

The main non-ANSI extension still used is the INCLUDE ‘filename’ command at the head of all subroutines. This command is used to set the maximum size of the various arrays. If your compiler does not allow this extension then the INCLUDE statements must be manually replaced by parameter definitions, as outlined in at the beginning of the program. Alternatively, since this would be quite time consuming simply contact us and request a different version of the program.

2.2 Precision

The program is in double precision, but may be converted to single precision by replacing the statements ‘IMPLICIT DOUBLE PRECISION ..’ at the head of most subroutines to ‘IMPLICIT REAL ..’. Also some of the intrinsic functions may need to be changed to their real counterparts.

2.3 Size Restrictions

At compilation all arrays are specified an upper limit. This limits some of the variables to certain sizes. If the need arises these limits can simply be increased by simply modifying the file EMMIX.max and re-compiling. The current limits are given in the Appendix A.

3 Input File

For most of the analysis options the input file, mainly contains the data set to be analysed. The data is listed as a data point on each line, with each data point consisting of one or more variables separated by one or more space(s), tab(s) or comma(s). Depending on which options are utilised when running the program, extra information may be required and should be appended to the end of the input file as will be discussed n later sections.

Example

For a sample consisting of 5 data points each with 3 measurements

```
3.456 2.657 1.542
5.768 3.876 1.345
3.567 7.986 0.932
6.431 6.532 2.012
0.423 9.741 1.034
```

4 Interacting with the Program

Due to the need for the code to be compatible across a number of platforms, much of the input to the program is specified by answering sequential questions at the beginning of the program rather than a graphical user interface. Where possible if an incorrect answer is given the program will repeat the question. Specific instructions and examples are given in the following sections. Firstly, the user will be presented with the main menu:

```
       ------------------------------------------------------
                _____  __   __    __   __ __ _    _
               | ____| | \_/ |    | \_/ | || \\  //
               ||____  ||\_/||    ||\_/|| ||  \\//
               | ____| ||   || __ ||   || ||   ||
               ||____  ||   || -- ||   || ||  //\\
               |_____| ||   ||    ||   || || //  \\
       ------------------------------------------------------
```

EM based MIXTURE program

Version 1.3 1999

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

          Do you wish to:
           0. Simulate a sample from a normal mixture model
           1. Carry out a bootstrap-based assessment of
              standard errors and/or the number of components (g)
           2. Fit a g-component normal mixture model for a
              specified g
           3. Fit a g-component normal mixture model for a
              range of values of g
           4. Perform discriminant analysis
           5. Make predictions for new data
           6. Form parameter estimates from data + allocation
------------------------------------------------------
```

5 Mixture Analysis for a Given Number of Components

This section corresponds to the situation where the number of components (g) in the normal mixture model is known and specified by the user.

Input file

The input file should contain the data set as described in Section 3, plus any other information appended at the end of the file depending on what options are chosen.

Output file

The output file contains the results of the fit of a mixture model with the user specified number of components.

User input

The following is an example how how to start an analysis for a specified number of components (comments are given in square brackets).

```
   ------------------------------------------------------
             Do you wish to:
              0. Simulate a sample from a normal mixture model
              1. Carry out a bootstrap-based assessment of
                 standard errors and/or the number of components (g)
              2. Fit a g-component normal mixture model for a
                 specified g
              3. Fit a g-component normal mixture model for a
                 range of values of g
              4. Perform discriminant analysis
              5. Make predictions for new data
              6. Form parameter estimates from data + allocation
   ------------------------------------------------------

2
Enter name of input file:
test.in                     [Specify the file containing the data]
 Enter name of output file:
test.out

Number of entities:
100                         [Number of samples in the data set]
Total Number of variables/dimensions in the input file:
2                           [Number of variables measured on each sample point]
 How many variables to be used in the analysis
 (re-enter  2 if you wish to use all the variables):
2                           [Number of variables to be used in analysis]
 How many components do you want to fit:
2
 Covariance matrix option (1 = equal,2 = unrestricted,
        3 = diagonal equal,4 = diagonal unrestricted)
2                           [See Section 5.1]
```

5.1 Covariance Structure

When fitting mixture model with EMMIX the user may constrain the covariance matrices to be either equal for all components, arbitrary, or diagonal (equal or unequal). Generally unless the user has some prior knowledge of the covariance structure arbitrary covariances should be used. If the no solution can be found due to singular covariance matrices then equal covariances may give a solution. Should the singularity problems still occur this may be because:

1. Two or more of the variables are highly correlated.
2. There are too many variables and not enough points.
3. One of the variables is discrete and a cluster is being fitted to a single point of high density.

5.2 Specified Initial Classification

This option initializes the EM algorithm from a specified classification of the data.

Additions to Input File

When this option is chosen, the user-defined partition must be appended to the end of the input file. For example:

```
data
 .....
data
1 1 1 2 2
2 2 2 2 2
```

This example would give the starting partition with the first 3 points belonging to component 1 and the remaining 7 points belonging to component 2.

5.3 Specified Initial Parameter Values

This option starts the EM algorithm from a specified initial values of the unknown mixture model parameters, ie. the elements of the component means, covariance matrices and mixing proportions.

Additions to Input File

When this option is chosen, the user-specified values of the parameters must be appended to the end of the input file in the form outlined below:

mean component 1
lower diagonal form of covariance for component 1
mean component 2
lower diagonal form of covariance for component 1
etc.
mixing proportions component 1 component 2 etc.

for example:

```
data
etc.
0 0
1
0 1
2 1
.7
.1 .7
.25 .75
```

This example would give the starting parameters as,

$$\boldsymbol{\mu}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \boldsymbol{\mu}_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.7 & 0.1 \\ 0.1 & 0.7 \end{pmatrix}$$

and mixing proportions $\pi_1 = 0.25$ and $\pi_2 = 0.75$.

5.4 Specified Initial Posterior Probabilities of Component Membership

This option initialises the EM algorithm by specifying the posterior probabilities of component membership for each observation in the data set. For example, in the case of two components, they might be specified as 0.7 and 0.3, corresponding to components 1 and 2, respectively. The case where these probabilities are either 1 or 0 corresponds to the case, discussed previously, of an initially specified (hard) classification of the data set.

Input file (add)

When this option is chosen the user defined posterior probabilities (or weights) are appended to the end of the input file for example:

```
data
etc.
.7 .3
.5 .5
.2 .8
 etc.
```

In the case above, the probability of first point belonging to first component is 0.7 and second component is 0.3.

5.5 Unspecified Initial Start (Automatic Approach)

With this option, the user does not supply any information concerning an initial value to start the EM algorithm. The program proceeds by obtaining an outright classification of the data by considering the output obtained by applying various clustering techniques to the data set. The clustering that produces the highest log likelihood is adopted as the initial classification for the purposes of starting the EM algorithm.

Additions to Input File

No addition to the main input file is required.
(Optional): the file 'hier.inp' may be used to control which hierarchical methods are utilised.

The various clustering methods available in the current version are:

- Hierarchical clustering (on standardised and unstandardised data):
  - Nearest Neighbour (Single Linkage)
  - Furthest Neighbour (Complete Linkage)
  - Group Average (Average Linkage)
  - Median
  - Centroid
  - Flexible Sorting
  - Incremental Sum of Squares (Ward's Method)
- Random partitions of the data
- K-means clustering algorithm

The choice of these methods is controlled in two ways. The random and k-means clustering are controlled by the following two questions:

```
How many random starts:
10
 What percentage of the data is to
 be used to form random starts:
70
How many k-means starts:
10
```

Concerning the randomly selected starts, there is the provision whereby the program can first subsample the data before using a random start based on the subsample each time. This is to limit the effect of the central limit theorem which would have the randomly selected starts being similar for each component in large samples.

To specify which hierarchical methods are to be used a file called 'hier.inp' must be created. The file should consist of pairs of numbers, each pair specifying a hierarchical clustering method to be used by the program. The last pair of numbers **MUST** be two negative ones (to indicate that no continuation is to occur).

For each pair of values (not including the terminating negative ones) a hierarchical clustering strategy will be produced. The two numbers refer to the programs variables ISU and IS:
IF ISU =1 then the data is to be standardised
IF ISU =2 then the data is not to be standardised
The value of IS corresponds to the clustering method to be used:

1. Nearest Neighbour (Single Linkage)

2. Furthest Neighbour (Complete Linkage)

3. Group Average (Average Linkage)

4. Median

5. Centroid

6. Flexible Sorting*

7. Incremental Sum of Squares (Ward's Method)

*If IS = 6, then an extra parameter BETA is needed; this should be entered
on the next line by itself. BETA equal to zero corresponds to the Furthest Neighbour method, as BETA tends to 1 the method generally produces long shaped clusters, and for BETA smaller than zero the method produces small compact clusters.

EXAMPLE ‘hier.inp’

```
1 3
2 3
1 6
.9
1 2
2 2
1 7
2 7
-1 -1
```

If this file is not present then default values are used.

NOTE : In situations where the data sets contain a large number of points the hierarchical methods are generally infeasible in terms of both space and time. To use no hierarchical methods the file ‘hier.inp’ should be created containing only two negative ones. Alternatively, the hierarchical methods may be permanently switched off at compilation time; see Appendix A.

6 Bootstrap estimate of the null distribution or $-2\log\lambda$

A resampling approach may be used to assess the null distribution (and hence the P-value) of the log likelihood ratio test ($-2\log\lambda$) to test for H_0: $g = g_0$ versus H_1: $g = g_0{+}1$; see McLachlan (1987).

Input file

The input file should contain the parameters under the null for the original sample ONLY. The format of the parameters is the same as specified in Section 5.3 for the user-specified initial parameter values option.

Output file

The output file contains the sorted values of -2 $\log\lambda$ and their corresponding likelihood under the null and composite hypotheses.

RespH0.out contains the fit from the last bootstrap replicate produced under H_0.

RespH1.out contains the fit from the last bootstrap replicate produced under H_1.

Bsamp.out contains he bootstrap sample from the last bootstrap replicate.

If a particular replicate is of interest the random seeds should be noted and the program run again with these seeds and only a single replication specified. This will give the desired output files for this replication.

Any errors are reported in the output file and a warning is added if the log likelihood, under H_1, is less than the log likelihood, under H_0. This phenomena reflects that a good maxima has not been found, under H_1, and that maybe more starts should be used.

Given below is an example of how to produce a bootstrap analysis, with comments in square brackets.

User input

```
   ------------------------------------------------------
             Do you wish to:
              0. Simulate a sample from a normal mixture model
              1. Carry out a bootstrap-based assessment of
                 standard errors and/or the number of components (g)
              2. Fit a g-component normal mixture model for a
                 specified g
              3. Fit a g-component normal mixture model for a
                 range of values of g
              4. Perform discriminant analysis
              5. Make predictions for new data
              6. Form parameter estimates from data + allocation
   ------------------------------------------------------

1               [A bootstrap analysis is specified]

Enter name of input file:
boot.in         [Specify the file containing the  parameters
                       of the original sample under the null]

  Do you want:        [Calculate Standard Errors if required]
   1. A Bootstrap analysis of -2log(Lambda)
   2. A Standard Error analysis
   3. Both 1 and 2
1
 Enter name of output file for Bootstrap:
boot.out                [Specify the output file]

  How many bootstrap replications
99          [The number of bootstrap replications required]
  Number of entities:
100        [Number of samples or data points]
  Total Number of variables/dimensions in the input file:
2          [Number of variables measured on each data point]
  How many variables to be used in the analysis
  (re-enter  2 if you wish to use all the variables):
2
  What value of g do you wish to test (g vs g+1)
1         [The number of components under the null hypothesis]
  Covariance matrix option (1 = equal,2 = unrestricted,
        3 = diagonal equal,4 = diagonal unrestricted)
2         [See Section 5.1]
How many random starts:
```

```
10        [Number of random starts used when fitting under H1]
What percentage of the data is to
be used:
70
How many k-means starts:
10
Modify extra Options(Y/N):
n
```

7 Standard Error Analysis

This analysis produces estimates of the standard errors for the estimated parameters in the mixture model. However, no standard errors are reported for correlations between the estimated parameters due to the large number of combinations this would involve. Although, upon request a modified version of the program could be created that produces a specified combination. The standard errors may be assessed using one of the following methods.

- The parametric bootstrapping
- The nonparametric bootstrapping (ie. by sampling with replacement)
- The using the weighted likelihood bootstrap to create samples
- The using an information-based method (unequal covariance matrices only)

See Basford, Greenway, McLachlan and Peel (1997) for more details.

Input file

The input file should contain the parameters under the null for the original sample. The format is as specified in Section 5.3 for the user parameter option.

Output file

The output file will contain the parameter estimates for individual bootstrap samples and the standard errors.

RespSE.out contains the fit from the last bootstrap replicate produced.

SEsamp.out contains the bootstrap sample from the last replicate.

User input

```
   ------------------------------------------------------
             Do you wish to:
              0. Simulate a sample from a normal mixture model
              1. Carry out a bootstrap-based assessment of
                 standard errors and/or the number of components (g)
              2. Fit a g-component normal mixture model for a
                 specified g
              3. Fit a g-component normal mixture model for a
                 range of values of g
              4. Perform discriminant analysis
              5. Make predictions for new data
              6. Form parameter estimates from data + allocation
   ------------------------------------------------------

1                          [Specify a Standard Error analysis]
 Enter name of input file:
test.in

 Do you want:
   1. A Bootstrap analysis of -2log(Lambda)
   2. A Standard Error analysis
   3. Both 1 and 2
2           [Incorporate a bootstrap analysis of -2log(lambda) if required]
  Enter name of output file for Standard Errors:
test.out
 Which method of estimation:
  1 Parametric
  2 Sampling with replacement
  3 weighted likelihood
  4 information based method
1              [Specify type of method to estimate Standard Errors]
  [Warning may need extensive time]
```

```
   How many replications to estimate
   the Standard Errors
100
 Number of entities:
100            [Number of sample points in original sample]
 Total Number of variables/dimensions in the input file:
2
 How many variables to be used in the analysis
 (re-enter  2 if you wish to use all the variables):
2
 How many components do you want to fit:
2
  Covariance matrix option (1 = equal,2 = unrestricted,
        3 = diagonal equal,4 = diagonal unrestricted)
2            [See section 5.1]
```

8 Simulation from Multivariate Normal Mixtures

EMMIX allows the generation of samples from a user specified multivariate normal mixture model.

Input file

A user specified file with the mixture model parameters in the format described in Section 5.3.

Output file

A user specified file containing the generated sample and the true allocation.

User input

The following gives an example input to generate a sample:

```
------------------------------------------------------
 Do you wish to:
```

```
              0. Simulate a sample from a normal mixture model
              1. Carry out a bootstrap-based assessment of
                 standard errors and/or the number of components (g)
              2. Fit a g-component normal mixture model for a
                 specified g
              3. Fit a g-component normal mixture model for a
                 range of values of g
              4. Perform discriminant analysis
              5. Make predictions for new data
              6. Form parameter estimates from data + allocation
            ------------------------------------------------------
0
 Enter name of input file:
samp.inp                      [input file containing model parameters]
 Enter name of output file:
samp.out

 Number of entities:
150
 Total Number of variables/dimensions
  in the input file:
3
 How many variables to be used in the analysis
 (re-enter  3 if you wish to use all the variables):
3
 How many components do you want to generate:
2
```

9 Mixture Analysis for a Range of Number of Components

This analysis is undertaken in the case of fitting a mixture model where the number of components is unspecified. The user must specify a range for the number of components in the mixture model to be fitted; eg. 1 to 10. For this specified range, the program fits the mixture model for each value of g, in turn, in the specified range. Finally, various statistics are reported comparing the fits obtained to aid in the decision on the number of components. Estimates of the P-values may also be reported.

Input file

The input file should contain the data set or sample listed as described in Section 3.

Output file

The output file contains the fits obtained sequentially for the range specified plus a summary of the fits.

User input

```
   ------------------------------------------------------
     Do you wish to:
      0. Simulate a sample from a normal mixture model
      1. Carry out a bootstrap-based assessment of
         standard errors and/or the number of components g
      2. Fit a g-component normal mixture model for a
         specified g
      3. Fit a g-component normal mixture model for a
         range of values of g
      4. Perform a Discriminant Analysis
      5. Make Predictions for new data
   ------------------------------------------------------
3
Enter name of input file:
test.in
 Enter name of output file:
test.out

Do you wish to carry out a bootstrap test
 to assess the number of components (Yes/No)-
n
 Number of entities:
100
 Total Number of variables/dimensions in the input file:
2
 How many variables to be used in the analysis
 (re-enter  2 if you wish to use all the variables):
2
 What is the minimum number of components
 you wish to test (eg 1):
1
```

```
 What is the maximum number of components
 you wish to test (eg 10):
10
 Covariance matrix option (1 = equal,2 = unrestricted,
       3 = diagonal equal,4 = diagonal unrestricted)
2
How many random starts:
10
What percentage of the data is to
be used:
70
How many k-means starts:
10
```

9.1 Bootstrap-Based Approach to Tests on Number of Components

In the case where the number of groups is unknown one approach is to use the likelihood ratio test statistic -2log(lambda) and utilise a bootstrap procedure to estimate it's corresponding P-value ; see McLachlan (1987). EMMIX has the option when fitting a range of values of g (where g is the number of components), as per the previous section, to implement a bootstrap of the likelihood ratio test statistic at each stage. Hence P-values are provided to establish how many components to fit.

Input file

The sample to be analysed.

Output file

The output file contains the fits obtained sequentially for the range specified plus a summary of the fits. Appended to the standard output file is a table which lists the estimated P-values.

Optional

RespH0.out contains the fit from the last bootstrap replicate produced under H_0.

RespH1.out contains the fit from the last bootstrap replicate produced under H_1.

Bsamp.out contains he bootstrap sample from the last bootstrap replicate.

User input

The input is as per the last section except for:

```
Do you wish to carry out a bootstrap test
 to assess the number of components (Yes/No)-
y
 [Warning may need extensive time]
  How many bootstrap replications
99
```

Optional files

The optional output files *bootXvsY.out* contain the sorted values of $-2\log\lambda$ and their corresponding likelihood under the null and composite hypothesis for $g = X$ vs $g = Y$.

RespH0.out contains the fit from the last bootstrap replicate produced under H_0.

RespH1.out contains the fit from the last bootstrap replicate produced under H_1.

Bsamp.out contains he bootstrap sample from the last bootstrap replicate.

9.2 Stopping Rules for Assessment of P-Values

This option allows the program to stop the analysis when the P-value (assessed by bootstrapping $-2\log\lambda$) becomes insignificant. To use this option, simply answer the relevant question with a 1, and then give the significance level as a percentage from the upper tail.

```
 Do you wish to stop when P-value is   insignificant (0-No,1-Yes)
1
 What level of significance (ie. 10 =10%)
10
```

10 Discriminant Analysis

Using this option the user supplies a classified sample (training data) then EMMIX will classify the remaining sample.

Input file

The user defined input file should contain the data set, or sample, as described in Section 3, followed by the allocation of the classified sample in the form of the point number followed by the point's classification on a separate line, for each of the classified points. When the list is complete two negative ones should be used to denote the end.

EXAMPLE

```
Sample +
1  3
2  3
3  3
4  2
5  1
6  2
10 3
11 2
-1 -1
```

Output file

The user defined output file contains the resulting classification of the sample plus other relevant information.

User input

To use this option:

```
            ------------------------------------------------------
             Do you wish to:
              0. Simulate a sample from a normal mixture model
              1. Carry out a bootstrap-based assessment of
                 standard errors and/or the number of components (g)
              2. Fit a g-component normal mixture model for a
                 specified g
              3. Fit a g-component normal mixture model for a
                 range of values of g
              4. Perform discriminant analysis
              5. Make predictions for new data
              6. Form parameter estimates from data + allocation
            ------------------------------------------------------
4
 Enter name of input file:
test
 Enter name of output file:
test.out

 Number of entities:
50
 Total Number of variables/dimensions
  in the input file:
4
 How many variables to be used in the analysis
 (re-enter  4 if you wish to use all the variables):
4
 How many components do you want to fit:
2
 Covariance matrix option (1 = equal,2 = unrestricted,
    3 = diagonal equal,4 = diagonal unrestricted):
2
```

11 Prediction for a New Sample

Given a mixture model parameters this option predicts the posterior probabilities and allocation for a new sample based on these existing model parameters.

Input file

The user defined input file should contain the new data set or sample listed as described in Section 3, followed by the existing mixture model parameters in the form specified in Section 5.3 for the user parameter option.

Output file

The user defined output file contains the resulting allocation of the new sample plus other relevant information.

Input file

To use this option simply type:

```
            ------------------------------------------------------
             Do you wish to:
              0. Simulate a sample from a normal mixture model
              1. Carry out a bootstrap-based assessment of
                 standard errors and/or the number of components (g)
              2. Fit a g-component normal mixture model for a
                 specified g
              3. Fit a g-component normal mixture model for a
                 range of values of g
              4. Perform discriminant analysis
              5. Make predictions for new data
              6. Form parameter estimates from data + allocation
            ------------------------------------------------------
5
 Enter name of input file:
test
 Enter name of output file:
test.out

 Number of entities:
50
 Total Number of variables/dimensions
  in the input file:
4
```

```
 How many variables to be used in the analysis
 (re-enter  4 if you wish to use all the variables):
4
```

12 Random Seeds

If the program requires random numbers it will ask the user for some sort of random seed(s) depending on which random number generator is being used, for example:

```
Random seeds 3 seeds needed :
  random seed 1 [0-30000]:
54
  random seed 2 [0-30000]:
3546
  random seed 3 [0-30000]:
6464
```

13 Other Options

Various options have been added during the programs development and are contained under the sub-menu of ‘extra options’. Some of these options have been added for the use of specific users of this program and may not be of use to the average user.

The options are accessed by replying yes to the question:

```
Modify extra Options(Y/N):
y
```

The user is then presented with a menu of the extra options as well as the current status, ie. on or off. Selecting an option will either toggle the option on to off (or vice versa), or enter a question/answer environment to gain more information. Options that are only available in certain types of analysis are given a ’N/A’ status when they are not valid.

```
               EXTRA OPTIONS
  ---------------------------------------
 Please select option (selection will toggle):
   1. Stochastic EM option : NO
   2. Modify EM stopping criteria
   3. Space efficiency : OFF
   4. Add extra output files
   5. Partial classification : OFF
   6. Estimate standard errors : NO
   7. Bootstrap test : NO
   8. Display discriminant density values : NO
   9. Change component distribution
      (Currently fitting NORMAL components)
   0. Run program
 ------------------------------------
```

13.1 Stochastic EM Algorithm

The Stochastic EM is an extension of the EM algorithm which may be specified. The basic principle of the Stochastic EM is similar in spirit to simulated annealing, in that randomness is added to the iterative process to give the algorithm a chance to escape local maxima.

13.2 Adjusting Stopping Criteria for the EM Algorithm

The stopping criteria used in EMMIX is based on the change in the log likelihood from the current iteration and the log likelihood from ten iterations previously. If this change differs by less than a specified tolerance multiplied by the current log likelihood then the algorithm will stop. If the algorithm does not converge before a predetermined number of iterations the algorithm stops and a warning is reported. These values may differ for the final fit and the investigative fits used when finding a start automatically. To change the values permanently the values are changed at compilation as outlined in the Appendix A. To change the values temporarily just for the current analysis, choose option 2 from the extra options menu. The program then asks for new values, a zero will leave the value as its default value.

```
      -Set tolerance automatic methods
       (Default=    1.00000D-06)
       Either set new value or 0 for default:
```

```
.00001
  -Set max number of iterations for automatic
   methods (Default=  500)
    Either set new value or 0 for default:
300
  -Set tolerance final fit
   (Default=    1.0000D-06)
   Either set new value or 0 for default:
0
  -Set max number of iterations for final
   fit (Default=  500)
     Either set new value or 0 for default:
0
```

13.3 Partial Classification

This option allows the user to specify the classification of some data points. The specified points will retain their classification throughout the fitting process.

The input file is appended with the classification of the specified points. The form is simply a list of the point number followed by the point's classification (group number). When the list is complete two negative ones should be used to denote the end.

13.4 Optional Standard Errors

The standard errors of the estimates as discussed in Section 7 may be calculated during any general cluster analysis. To produce standard errors choose option 6 from the extra options menu then,

```
Which method of estimation:
 1 Parametric
 2 Sampling with replacement
 3 weighted likelihood
 4 information based method
1
 How many replications do you wish to use:
99
```

13.5 Space Efficiency

Due to some users analysing extremely large data sets the output files have in some cases become very large causing the machine to run out of space and the program to crash. Since much of the information in these output files is probably not needed for a general analysis the output may be optionally shortened to save space. This space saving can be applied at two levels moderate or extreme. To use the space efficient version choose option 3 from the extra options menu.

```
What level of space efficiency:
 0. None
 1. Moderate
 2. Extreme
```

13.6 Files for Exportation to External Plotting Programs

This option has been requested by users of the program and added in this version of EMMIX. When selected an additional user specified output file is created containing the point index and its corresponding allocation for easy exportation to external plotting software. To produce this file option 4 is taken from the extra options menu:

```
Do you want to output the data and
 resulting allocations (0-no, 1=yes)
1
What do you wish this file to be called:
plot.clus
```

Similarly a plotting file may be produced for the bootstrap distribution of $-2\log\lambda$. To produce this file the following option is taken

```
Do you want to output the bootstrap
 distribution values (0-no, 1-yes)
1
What do you wish this file to be called:
plot.boot
```

13.7 Fitting Mixtures of t-distributions

For general applications fitting mixtures of multivariate normals offer a good all round model. However, in cases where outliers are present in the data fitting mixtures of multivariate t-distributions may be more appropriate.

To fit mixtures of t-distributions option 9 must be taken in the other options menu. The following sub-menu is then displayed:

```
1-Fixed user-defined degrees of freedom $\nu$ for each component
2-Degrees of freedom NU estimated for each component
  (from user-supplied initial value)
3-Common degrees of freedom NU estimated for the components
  (from user-supplied initial common value)
4-Degrees of freedom NU estimated for each component
  (moments estimates used as the initial values)
```

This sub-menu is used to initialize the degrees of freedom parameter ν; see McLachlan and Peel (1998) for more details. Utilising options 2 and 3 the degrees of freedom are estimated from the sample.

The resulting ν values are reported in the output file as well as the weights u_{ij} which give an indication of points that are atypical.

13.8 Using Aitken's Acceleration

This feature is applicable when utilising the bootstrap option of EMMIX to assess an appropriate number of components to fit. Aitken's acceleration can be used to reduce the number of iterations required at each fit by predicting the likelihood value that the EM algorithm is converging to, and using this estimate to calculate the likelihood ratio test statistic. From initial tests it would seem the error inccured from using Aitken's acceleration is minimal so this option should be selected when using the bootstrap option.

14 Program Output

14.1 Screen Output

A summary of the information given to the program is presented on the screen for the user to check, plus an outline of what the form of input file should be, and then the programs progress is reported.

14.2 The Output File

A thorough description of the fit is given in user specified output file (in the examples presented here 'test.out'). The first thing written to the output file is a summary of the analysis parameters ie input/output files, type of analysis etc. Next, any information for the starting point of the EM algorithm is reported; eg. if user parameters are used they are written. For an automatic start, the clustering method is named, the allocation found, and the log likelihood is reported, as well as any problem that has occurred during the fitting procedure. See the example below:

```
------------------------------------------------------
     1 UNSTANDARDIZED GROUP AVERAGE
     2   2   1   2   2   1   2   1   2   1
     2   2   2   2   2   2   2   1   1   2
     2   2   1   2   2   1   2   2   2   1
     1   1   2   2   2   1   2   2   2   2
     2   2   2   2   2   2   2   2   2   2

  Log likelihood value from EM algorithm started
  from this grouping is        -36.994
------------------------------------------------------
```

After this has been done for all the starting methods, a list of the log likelihood values for the starting methods used, is given (as below, for example).

```
------------------------------------------------------
  Final log likelihood values from each initial grouping
-36.994  -36.994  -36.994  -36.994  -36.994  -36.994
-40.359 -43.303 -49.624  -40.359  -45.621  -40.359
```

```
-36.994  -43.303 -43.303 -45.591  -36.994
------------------------------------------------------
  Best initial grouping (corresponding to the
  highest value of likelihood found by the
  STANDARDIZED   GROUP AVERAGE     method
```

Next the output from the best initial start is reported.

```
  Estimated mean (as a row vector) for component  1
     6.38617       2.94637       5.37070       2.03828

  Estimated mean (as a row vector) for component  2
     7.52561       3.10235       6.39424       1.96897

  Estimated covariance matrix for component    1
    0.2392
    0.7246E-01  0.8376E-01
    0.1405      0.5735E-01  0.1511
    0.6416E-01  0.5698E-01  0.5641E-01  0.7985E-01

  Estimated covariance matrix for component    2
    0.5733E-01
    0.3586E-01  0.1662
    0.6557E-01 -0.2904E-02  0.1208
    0.3851E-01  0.7687E-02  0.6641E-01  0.4239E-01

  Mixing proportion from each component
       0.823  0.177

  Starting Grouping Found
     1   1   1   1   1   2   1   2   1   1
     1   1   1   1   1   1   1   2   2   1
     1   1   2   1   1   2   1   1   1   2
     2   2   1   1   1   1   1   1   1   1
     1   1   1   1   1   1   1   1   1   1
```

The resultant likelihood and determinant for each iteration are then given.

```
  Determinants of component covariance matrices
    3.6961163320559D-05    1.4321301000881D-06
  After iteration   0 the log likelihood =          -36.994

  Determinants of component covariance matrices
    3.6961163320689D-05    1.4321301000887D-06
  After iteration   1 the log likelihood =          -36.994
            etc.                        etc.

  Determinants of component covariance matrices
    3.6961163320719D-05    1.4321301000888D-06
  After iteration  10 the log likelihood =          -36.994
  Final log likelihood is         -36.994
```

Then the data (if less than 4 variables) and the posterior probabilities are reported for each data point for the final fit.

```
Observation mixture log density Component 1, Component 2, ..etc...
       1    0.51150E-01  1.0000 0.0000
       2     1.4686      1.0000 0.0000
       3    0.77566      1.0000 0.0000

                 etc.                     etc.

      49    0.38811      1.0000 0.0000
      50    0.77427      1.0000 0.0000
```

The final implied outright clustering is given and the parameters estimates.

```
 Implied grouping of the entities into   2 component
     2   2   2   2   2   1   2   1   2   2
     2   2   2   2   2   2   2   1   1   2
     2   2   1   2   2   1   2   2   2   1
     1   1   2   2   2   2   2   2   2   2
     2   2   2   2   2   2   2   2   2   2

  Number assigned to each component
       9    41

  Estimate of mixing proportion for each component
    0.177  0.823

  Estimates of correct allocation rates for each component
    1.000  0.996
  Estimate of overall correct allocation rate   0.997

  Estimated mean (as a row vector) for each component
       7.525611     3.102347     6.394242     1.968968
       6.386173     2.946372     5.370702     2.038277
```

```
Estimated covariance matrix for component  1
  5.7339D-02
  3.5869D-02   0.1662
  6.5576D-02  -2.9045D-03   0.1208
  3.8513D-02   7.6876D-03   6.6412D-02   4.2397D-02

Estimated covariance matrix for component  2
  0.2392
  7.2466D-02  8.3764D-02
  0.1405         5.7356D-02   0.1511
  6.4166D-02   5.6983D-02   5.6417D-02   7.9859D-02
```

If a mixture analysis is performed for a range of g, the above listing for the output file is repeated sequentially for each value fitted for the number of components (g). Finally a table is given summarising the values of the tests to help decide on the number of components (as shown in the example that follows).

g	log lik	$-2\log\lambda$	AIC	BIC	AWE	P-value
1	-230.76	-	465.52	472.52	487.53	-
2	-54.64	352.24	119.28	136.79	174.29	0.01
3	-47.83	13.63	111.65	139.66	199.67	0.02
4	-40.95	13.75	103.90	142.41	224.93	0.05
5	-37.78	6.33	103.56	152.58	257.60	0.39

The various criteria currently reported by EMMIX are AIC, BIC and AWE. The number of groups is given by the value for which the criteria value is minimised; for example, in this case, AIC predicts 5, BIC and AWE both predict 2 clusters.

The *P*-value (P-VAL) is produced by the optional bootstrap analysis. By sequentially testing eg. '1 versus 2' then '2 versus 3', and so on, and stopping when the step becomes insignificant, the number of components can be assessed. In this case we would stop at 4 components.

A EMMIX.MAX

Many of the arrays and matrices used by the program are set maximum sizes at compilation. These limits will control such things as the size of data set that may be analysed. To change any of these limits simply modify the relevant value in the file 'EMMIX.max' and recompile. This file also contains flags to control various options at compile-time, rather than run-time. Below is a copy of the file 'EMMIX.max', the changes required and relevant parameters should be obvious, for example to increase the maximum number of data points from 1110 to 4000 simply change the line,

```
       PARAMETER (MNIND=1000)
C     maximum number of data points is 1000
```

to

```
       PARAMETER (MNIND=5000)
c     maximum number of data points is 5000
```

If an analysis is attempted that exceeds any of these limits an error is reported and the program stops.

```
      PARAMETER (MNIND=1000)
C         maximum number of data points
      PARAMETER (MNATT=10)
C         maximum dimensionality of data points
      PARAMETER (MAXNG=10)
C         maximum number of components
      PARAMETER (MSTART=200)
C         maximum number of initial starts to be displayed
C         in the final list
      PARAMETER (LIMZ=400000)
C          maximum size of global array used for storage
C          within hierarchical section.
      PARAMETER (MHIER=10)
C         maximum number of hierarchical methods to be used
      PARAMETER (MKMEAN=500)
C         maximum number of iterations used in k-means
      PARAMETER (TAUTO=.000001)
C         the default tolerance for the EM algorithm when
C         investigating initial starts
```

```
      PARAMETER (MITAUT=500)
C       the default maximum number of iterations when
C       investigating initial starts
      PARAMETER (TFINAL=.000001)
C         the default tolerance for the EM algorithm when
C         iterating the final fit (The best initial fit found)
      PARAMETER (MITFIN=500)
C         the default maximum number of iterations when
C         iterating the final fit (The best initial fit found)
      PARAMETER (MITER=1000)
C         maximum number of iterations for the EM algorithm
      PARAMETER (HIRFLG=1)
C         flag to switch on (1) and off (0) hierarchical
C         methods switch off for large data sets
      PARAMETER (MAXREP=1000)
C          maximum number of bootstrap replications
      PARAMETER (NUMAX=300)
C          maximum value Nu can take when fitting t-distributions
      PARAMETER (XLOWEM=1.0E-30)
C          minimum value density of a point is before it is considered
C          to be zero (also minimum value of the mixing proportion
      PARAMETER (DENMAX=175)
C          maximum value of the A term in exp(-A) used when calculating
C          the density of a point. Above this value exp(-A) is equated
C          to zero.
```

B Flags

FLAG	DESCRIPTION
1	Different random starting methods (Not this version)
2	Stochastic EM FLAG (0-normal EM, 1-Stochastic EM)
3	Temp 1- tru data fit 2- bootstrap fit (no output to screen) 3 -Bootstrap under H_0
4	Type of start 1 -partition, 2 -parameter 3 -auto 4 -weights
5	Number of k-means starts
6	Display density values to use as a discriminant rule
7	T density (U ,0 -no T)
8	0 -simulate 1 -Bootstrap analysis, 2-Specific analysis, 3 -Full auto analysis, 4 -Discriminant, 5 -Prediction
9	1 -Final EM iterations / 2 -Initial EM iterations
10	Resamp test (0-No, $>$ 0 -yes (Number of replications))
11	Space efficient version (0 -no 1 -partial, 2 -extreme)
12	Partial user allocation knowledge (0=no, 1=yes)
13	Unused
14	Weighted data set (0=no, 1=yes)
15	Output data+partition for external plot (0=no, 1=yes)
16	Output boot distrib for external plot (0=no,1=yes)
17	Estimate Standard Errors (0 -no, $> 0 =$ Num of its or =1 yes)
18	S.E. Method (0 -para, 1 -samp w/replace, 2 -weight lik, 4 -info method)
19	Variable Selection : 1 -adjust data, 2 -adjust parameters as well
20	Output to separate file 1 -parameters, 2 -point likelihoods, 3 -data
21	Use Aitken's acceleration during bootstrapping ($<$ 0 active $>$ 0 on)
22	Output subset of data to separate file

C Error Codes

CODE	DESCRIPTION
1	Covariance matrix pivot zero (ie close to singular)
2	Covariance matrix is not positive semi-definite
4	Nullity = 0
5	Determinant = 0
11	Number of data points too large for this compilation
12	Number of data variables too large for this compilation
14	Maximum Number of clusters too large for this compilation
15	Number of clusters too large for this compilation
21	Not enough points in cluster at initial estimation stage
22	No points allocated to component during an EM iteration
23	Problem in the generation of a bootstrap sample
40	Random number generator not working
51	Warning : k-means did not converge
52	Warning : Some points have zero likelihood

D Input/Output File ID Numbers

ID	PURPOSE
21	Main data file + starting parameters or partition
22	Main output file from main gives clusterings
56	Optional allocation for export to external plotting package
57	Optional bootstrap for export to external plotting package
28	‘hier.inp’ optional input file specifies hierarchical methods
42	‘respH0.out’ output file for fit under H_0 for last bootstrap replicate
	‘respH1.out’ output file for fit under H_1 for last bootstrap replicate
43	Output file of bootstrap sample for last bootstrap replicate
25	‘boot?versus?.out’ output file contain bootstrap replicates of $-2\log\lambda$
26	Parameter estimates for replications used to estimate Standard errors

E Example Input File

For 5 data points each with 2 variables and 2 components

3.456 2.657

5.768 3.876

3.567 7.986

6.431 6.532

0.423 9.741

followed by

option 1 (user partition)

1 2 1 2 2 [user- supplied classification or option 2 (parameter estimates)]

0 0 [mean for component 1]

1 [Lower triang of covariance component 1]

0.3 2

4 3.4 [mean for component 2]

5 [Lower triang of covariance component 2]

.4 1

.4 .6 [mixing proportions of components]

option 4 (user weights)

.1 .2 .7 [prob component 1 prob component 2 prob component 3 for point 1]

.2 .3 .5 [prob component 1 prob component 2 prob component 3 for point 2]

etc.