The Variable Selection for Model-based Clustering (*clustvarsel*) package

Nema Dean

17th March 2006

Contents

1 Overview **1**

2 Main `clustvarsel` function **2**

3 Adjustments for Speeding Up Algorithm **5**

1 Overview

The *clustvarsel* package can be used to find the (locally) optimal subset of variables with group/cluster information in a dataset with continuous variables.

Each variable's evidence for being useful for clustering given the currently selected clustering variables is given by the difference between the BIC for the model with clustering (allowed to vary over 2 to a maximum number of groups and any of the different covariance parameterizations allowed in mclust02) using the set of clustering variables including the variable being checked and the sum of BICs for the model with clustering (allowed to vary over 2 to a maximum number of groups and any of the different covariance parameterizations allowed in mclust02) using the set of clustering variables without the variable being checked and the model for the variable being checked being conditionally independent of the clustering given the other clustering variables (this is modeled as a regression of the variable being checked on the other clustering variables).

For more details on the model see Raftery and Dean (2004), for an overview of model-based clustering see Fraley and Raftery (2002b) and for more details on the *mclust02* package see Fraley and Raftery (2002a) or `http://www.stat.washington.edu/mclust/`.

Two different search algorithms are available for checking single variables for inclusion into/exclusion from the set of selected clustering variables. The *"greedy"* search option checks at every inclusion step the inclusion of each single variable not currently selected into the current set of selected clustering variables. The variable that has highest evidence of inclusion is proposed and, if its clustering evidence is stronger than the evidence against clustering it is included. At every exclusion step the *"greedy"* search option checks the exclusion of each single variable in the currently selected set of clustering variables and proposes the variable that has lowest evidence of clustering. The proposed variable is removed if its evidence of clustering is weaker than its evidence against clustering.This is similar to the idea for stepwise regression and may suffer from the same instabilities mentioned in Miller (1990) inherent in that approach (although this has not been apparent in the simulations and examples tried thus far).

The *"headlong"* search (see Badsberg (1992) for full details) involves potentially checking less variables at each inclusion or exclusion step and so may be quicker than the *"greedy"* search (if possibly less optimal) for use on datasets with a large number of variables.

At each inclusion step, the algorithm only checks single variables not currently in the set of clustering variables up until the difference between the BIC for clustering versus not clustering is above a prespecified level *upper* (the default is 0, i.e. where evidence for clustering is greater than that for not clustering by any amount). Because the algorithm will stop once this criterion is satisfied it will not necessarily check all the variables available and the variable selected will not necessarily be the best possible variable at that step. Any variables who are checked during this step, whose difference between the BIC for clustering versus not clustering is below a prespecified level *lower* (the default is -10, i.e. whose evidence is strongly against clustering) are removed from consideration for the rest of the algorithm. Because of this, possibly irrelevant variables can be removed early on in the algorithm and further reduce the number of variables checked at each step. Similarly at each exclusion, the algorithm only checks single variables currently in the set of clustering variables up until the difference the BIC for clustering versus not clustering is below the same prespecified level *upper*. The algorithm stops checking once a variable satisfies this criterion and that variable is removed from the set. If the difference in BIC is smaller than *lower* the variable is removed from consideration for the rest of the algorithm, otherwise it can still be checked in future inclusion/exclusion steps.

The speed/optimally tradeoff can be changed by increasing or decreasing the different levels, e.g. by setting *upper* to 10 instead of 0 we require a variable to have stronger evidence of clustering before it is included and by setting *lower* to 0 we remove variables that at any stage have evidence of clustering weaker by any amount than evidence against clustering.

2 Main `clustvarsel` function

We begin by loading the package (the required *mclust02* package will automatically be loaded at the same time).

```
> library(clustvarsel)
```

The basic `clustvarsel` function has the default settings of using the *"greedy"* search, not using a subset for the hierarchical clustering phase of cluster model fitting, allowing an equal covariances hierarchical model when variable covariances model gives no viable answers, automatically selecting the first two variables regardless of whether the evidence of univariate/bivariate clustering is greater than not clustering (to provide starting variables) and a maximum number of paired inclusion and exclusion steps of 100. The default options for allowing different covariance parameterizations for both univariate clustering in *emModels1* and multivariate clustering in *emModels2* are all possible parameterizations currently available in *mclust02* (see the help file for `EMclust` for more details).

The only inputs required are a matrix X of continuous data with rows corresponding to observations and columns (at least 2) corresponding to variables and the maximum number of groups G believed to be possible in the data.

For our example we will use the crabs data in the *MASS* package. We know that there are 50 observations for each class (blue males, blue females, orange males and orange females) that are in order so we create a vector `crabscl`, identifying the true classification of the observations. We create our data matrix X and look at the pairs plot of the data.

```
> library(MASS)
> data(crabs)
> X <- crabs[, 4:8]
> colnames(X)

[1] "FL" "RW" "CL" "CW" "BD"

> pairs(X)
> crabscl <- c(rep(1, 50), rep(2, 50), rep(3, 50), rep(4, 50))
```

The pairs plot for X is given in Figure 1. First we look at the result from choosing the best model in terms of BIC for clustering on all variables allowing all possible parameterizations and the number of groups to range over 1 to 5 (our maximum number of possible clusters G in the dataset). We cluster the data using `EMclust` from the *mclust02* and look at the results using the `summary` function. We then extract the estimated classifications from the best model in terms of BIC from the summary and use `classError` to look at the misclassification error between the current and true classifications. We can use `table` to look at the cross tabulated classifications.

```
> clust1 <- EMclust(X, G = 1:5)
> summary(clust1, X)

classification table:

 1  2  3
75 77 48

uncertainty (quartiles):
          0%          25%          50%          75%         100%
0.000000e+00 1.171732e-07 1.853438e-03 2.770624e-02 4.892871e-01

best BIC values:
    EEV,3     EEV,4     EEE,3
-2925.590 -3016.711 -2933.521

best model: ellipsoidal, equal volume and shape

> cl1 <- summary(clust1, X)$class
> classError(cl1, crabscl)

[1] 0.375

> table(cl1, crabscl)
```

```
   crabscl
cl1  1  2  3  4
  1 25 50  0  0
  2  0  0 27 50
  3 25  0 23  0
```

Now we run the algorithm to automatically select which variables in the dataset are truly useful for clustering using `clustvarsel` and look at the steps of the algorithm. We then create `S`, the matrix of the selected variables and check which variables are in it.

```
> result <- clustvarsel(X, G = 5)
> result$steps.info
```

```
     Variable proposed BIC of new clustering variables set BIC difference
[1,] "CW"              "-1408.72243051261"                "-6.23030510251988"
[2,] "RW"              "-1908.96423491868"                "127.398162308462"
[3,] "FL"              "-2357.17088014855"                "81.3271493966722"
[4,] "FL"              "-2357.17088014855"                "81.3271493966722"
[5,] "BD"              "-2609.88963370441"                "55.8879054648505"
[6,] "BD"              "-2609.88963370441"                "55.8879054648505"
[7,] "CL"              "-2609.88963370441"                "-107.399613408393"
[8,] "BD"              "-2609.88963370441"                "55.8879054648505"
     Type of step Decision
[1,] "Add"        "Accepted"
[2,] "Add"        "Accepted"
[3,] "Add"        "Accepted"
[4,] "Remove"     "Rejected"
[5,] "Add"        "Accepted"
[6,] "Remove"     "Rejected"
[7,] "Add"        "Rejected"
[8,] "Remove"     "Rejected"
```

```
> S <- result$sel.var
> colnames(S)
```

```
[1] "CW" "RW" "FL" "BD"
```

Next we look at the results from clustering (using `EMclust`) on S using the `summary` command. Again we extract the classifications from this model, `cl2` and use `classError` and `table` to look at the misclassifications comparing it to the true classifications and the classifications from the model with all variables, `cl1`.

```
> clust2 <- EMclust(S, G = 1:5)
> summary(clust2, S)
```

```
classification table:

 1  2  3  4
60 40 45 55
```

```
uncertainty (quartiles):
          0%          25%          50%          75%         100%
0.000000e+00 1.878522e-08 4.637432e-05 1.225672e-02 4.964781e-01

best BIC values:
    EEV,4     EEV,5     VVV,4
-2609.890 -2760.546 -2731.526

best model: ellipsoidal, equal volume and shape

> cl2 <- summary(clust2, S)$class
> classError(cl2, crabscl)

[1] 0.075

> table(cl2, crabscl)

   crabscl
cl2  1  2  3  4
  1 10 50  0  0
  2 40  0  0  0
  3  0  0  0 45
  4  0  0 50  5

> table(cl1, cl2)

   cl2
cl1  1  2  3  4
  1 60 15  0  0
  2  0  0 45 32
  3  0 25  0 23
```

3 Adjustments for Speeding Up Algorithm

If we have a large number of observations in our dataset we allow `EMclust` to use only a subset of the data at the computationally expensive hierarchical stage of clustering to speed up the algorithm. We construct a medium sized dataset below for which we compare the time between using a subset (setting *samp* to TRUE) of 200 observations (setting *sampsize* to 200) and using all observations for the hierarchical clustering stage.

We set up the 4 dimensional data matrix X with clustering information only in the first two variables and define the mixing proportion `pro`, the two cluster means `mu1` and `mu2` and the two cluster covariances `sigma1` and `sigma2`. We then simulate from the model with these parameters and simulate independent normally distributed noise for the remaining two variables. We then use the function `system.time` to measure the time taken to run `clustvarsel` with a sample of 200 (*samp*=T, *sampsize*=200) and without sampling. Finally we check to make sure both selected the correct variables.

```
> X <- matrix(0, 400, 4)
> pro <- 0.5
> mu1 <- c(0, 0)
> mu2 <- c(3, 3)
> sigma1 <- matrix(c(1, 0.5, 0.5, 1), 2, 2, byrow = TRUE)
> sigma2 <- matrix(c(1.5, -0.7, -0.7, 1.5), 2, 2, byrow = TRUE)
> u <- runif(400)
> library(MASS)
> for (i in 1:400) {
+     ifelse(u[i] < pro, X[i, 1:2] <- mvrnorm(1, mu1, sigma1),
+         X[i, 1:2] <- mvrnorm(1, mu2, sigma2))
+ }
> X[, 3] <- rnorm(400, 1.5, 2)
> X[, 4] <- rnorm(400, 2, 1)
> system.time(result1 <- clustvarsel(X, G = 3, samp = T, sampsize = 200))
```

```
[1]  6.154  0.177 16.074  0.000  0.000
```

```
> system.time(result2 <- clustvarsel(X, G = 3))
```

```
[1]  7.144  0.212 22.668  0.000  0.000
```

```
> colnames(result1$sel.var)
```

```
[1] "2" "1"
```

```
> colnames(result2$sel.var)
```

```
[1] "2" "1"
```

If we have a large number of variables in our dataset we may find that using the *“headlong”* search algorithm option may be faster than the default *“greedy”* search. We construct a medium sized dataset below for which we compare the time between using the headlong method (search=*“headlong”*) and using the greedy method.

We set up the 8 dimensional data matrix X with clustering information only in the last two variables and define the mixing proportion `pro`, the two cluster means `mu1` and `mu2` and the two cluster covariances `sigma1` and `sigma2`. We then simulate from the model with these parameters and simulate independent and multivariate normally distributed noise for the remaining six variables. We then use the function `system.time` to measure the time taken to run `clustvarsel` with headlong search (*search*=“headlong”) versus greedy search (the default). Finally we check to make sure both selected the correct variables. We can also speed up the algorithm by restricting the models checked to equal and unconstrained covariances (*emModels2*=c(”EEE”,”VVV”)).

```
> X <- matrix(0, 200, 8)
> pro <- 0.5
> mu1 <- c(0, 0)
> mu2 <- c(3, 3)
> sigma1 <- matrix(c(1, 0.5, 0.5, 1), 2, 2, byrow = TRUE)
> sigma2 <- matrix(c(1.5, -0.7, -0.7, 1.5), 2, 2, byrow = TRUE)
> u <- runif(200)
> library(MASS)
> for (i in 1:200) {
+     ifelse(u[i] < pro, X[i, 7:8] <- mvrnorm(1, mu1, sigma1),
+         X[i, 7:8] <- mvrnorm(1, mu2, sigma2))
+ }
> X[, 1] <- rnorm(200, 1.5, 2)
> X[, 2] <- rnorm(200, 2, 1)
> X[, 3:4] <- mvrnorm(200, mu1, sigma1)
> X[, 5:6] <- mvrnorm(200, mu2, sigma2)
> system.time(result1 <- clustvarsel(X, G = 3, search = "headlong"))

[1]  3.999  0.121 11.818  0.000  0.000

> system.time(result2 <- clustvarsel(X, G = 3))

[1]  7.439  0.185 17.731  0.000  0.000

> colnames(result1$sel.var)

[1] "7" "8"

> colnames(result2$sel.var)

[1] "7" "8"

> system.time(result3 <- clustvarsel(X, G = 3, emModels2 = c("EEE",
+     "VVV"), search = "headlong"))

[1] 2.540 0.078 7.031 0.000 0.000

> colnames(result3$sel.var)

[1] "7" "8"
```

If we have both a large number of observations and variables we can use both the *samp* and *"headlong"* options. Again we generate an 8 dimensional data matrix X in a similar way to the previous example, this time with 400 observations. We then use `system.time` to compare the time taken for variable selection with the models checked restricted to equal and unconstrained covariances (*emModels2*=c(”EEE”,”VVV”)), “*headlong*” search and sampling 200 observations for the hierarchical clustering stage (*samp*=T and *sampsize*=200) versus variable selection just with *headlong* search and variable selection with *greedy* search.

```
> X <- matrix(0, 400, 8)
> pro <- 0.5
> mu1 <- c(0, 0)
> mu2 <- c(3, 3)
> sigma1 <- matrix(c(1, 0.5, 0.5, 1), 2, 2, byrow = TRUE)
> sigma2 <- matrix(c(1.5, -0.7, -0.7, 1.5), 2, 2, byrow = TRUE)
> u <- runif(400)
> library(MASS)
> for (i in 1:400) {
+     ifelse(u[i] < pro, X[i, 7:8] <- mvrnorm(1, mu1, sigma1),
+         X[i, 7:8] <- mvrnorm(1, mu2, sigma2))
+ }
> X[, 1] <- rnorm(400, 1.5, 2)
> X[, 2] <- rnorm(400, 2, 1)
> X[, 3:4] <- mvrnorm(400, mu1, sigma1)
> X[, 5:6] <- mvrnorm(400, mu2, sigma2)
> system.time(result1 <- clustvarsel(X, G = 3, emModels2 = c("EEE",
+     "VVV"), search = "headlong", samp = T, sampsize = 200))

[1]  5.420  0.190 16.108  0.000  0.000

> system.time(result2 <- clustvarsel(X, G = 3, search = "headlong"))

[1] 10.567  0.330 25.471  0.000  0.000

> system.time(result3 <- clustvarsel(X, G = 3))

[1] 17.827  0.480 48.359  0.000  0.000

> colnames(result1$sel.var)

[1] "8" "7"

> colnames(result2$sel.var)

[1] "8" "7"

> colnames(result3$sel.var)

[1] "8" "7"
```

References

Badsberg, J. H. (1992). Model search in contingency tables by CoCo. In Y. Dodge and J. Whittaker (Eds.), *Computational Statistics*, Volume 1, pp. 251–256.

Fraley, C. and A. E. Raftery (2002a, October). Mclust:software for model-based clustering, density estimation and discriminant analysis. Technical Report 415, University of Washington.

Fraley, C. and A. E. Raftery (2002b). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association 97*, 611–631.

Miller, A. J. (1990). *Subset Selection in Regression.* Number 40 in Monographs on Statistics and Applied Probability. Chapman and Hall.

Raftery, A. E. and N. Dean (2004). Variable selection for model-based clustering. Technical Report 452, Department of Statistics, University of Washington.

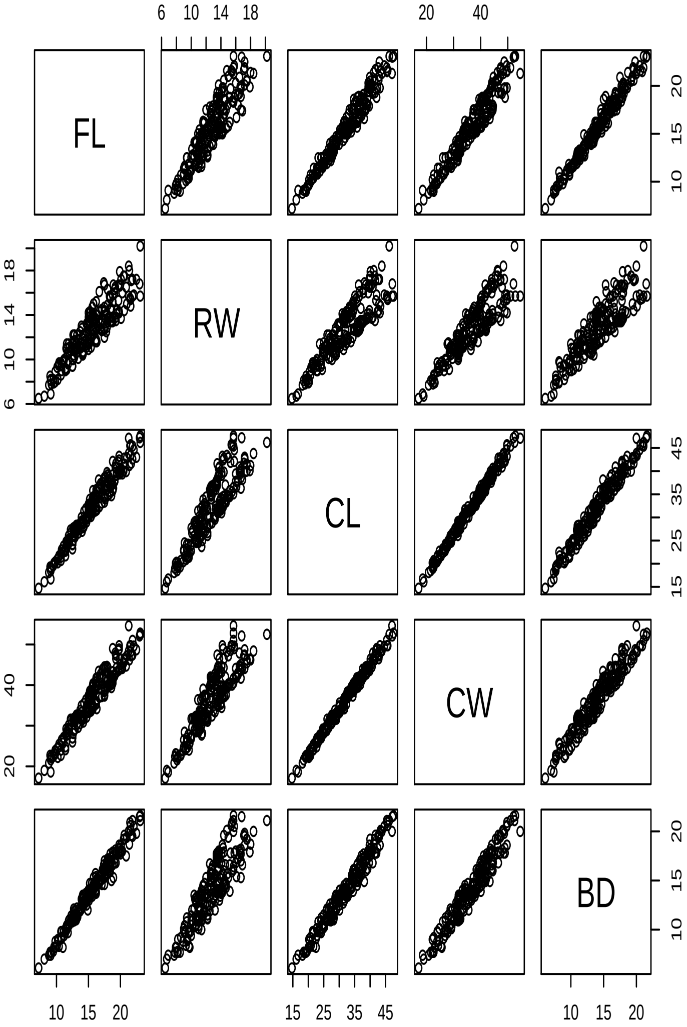

Figure 1: Pairs plot of the crabs data