VarSelLCM

Variable Selection for Model-Based Clustering of Mixed-Type Data Set with Missing Values

2018-08-30

References:

- Marbac, M. and Sedki, M. (2017), Variable selection for model-based clustering using the integrated complete-data likelihood, Statistics and Computing, Volume 27, Issue 4, pp 1049–1063.
- Marbac, M., Patin, E. and Sedki, M. (2018), Variable selection for mixed data clustering: Application in human population genomics, Journal of Classification, to appear.

This section performs the whole analysis of the *Heart* data set. *Warning the univariate margin distribution are defined by class of the features: numeric columns imply Gaussian distributions, integer columns imply Poisson distribution while factor (or ordered) columns imply multinomial distribution*

```r
library(VarSelLCM)
```

```

Attaching package: 'VarSelLCM'

The following object is masked from 'package:stats':

    predict
```

```r
# Data loading:
# x contains the observed variables
# z the known status (i.e. 1: absence and 2: presence of heart disease)
data(heart)
ztrue <- heart[,"Class"]
x <- heart[,-13]
# Add a missing value artificially (just to show that it works!)
x[1,1] <- NA
```

Clustering is performed with variable selection. Model selection is done with BIC because the number of observations is large (compared to the number of features). The number of components is between 1 and 3. Do not hesitate to use parallelization (here only four cores are used).

```r
# Cluster analysis without variable selection
res_without <- VarSelCluster(x, gvals = 1:3, vbleSelec = FALSE, crit.varsel = "BIC")

# Cluster analysis with variable selection (with parallelisation)
res_with <- VarSelCluster(x, gvals = 1:3, nbcores = 4, crit.varsel = "BIC")
```

Comparison of the BIC for both models: variable selection permits to improve the BIC

```r
BIC(res_without)
```

```
[1] -6516.216
```

```r
BIC(res_with)
```

```
[1] -6509.506
```

Evaluation of the partition accuracy: Adjusted Rand Index (ARI) is computed between the true partition (ztrue) and its estimators. The expectation of ARI is zero if the two partitions are independent. The ARI is equal to one if the partitions are equals. Variable selection permits to improve the ARI. Note that ARI cannot be used for model selection in clustering, because there is no true partition.

```
ARI(ztrue, fitted(res_without))
```

```
[1] 0.2218655
```

```
ARI(ztrue, fitted(res_with))
```

```
[1] 0.2661321
```

To obtained the partition and the probabilities of classification

```
# Estimated partition
fitted(res_with)
```

```
  [1] 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 1 2 1 1 1 1 1 2 1 2 2 2 2 1 2 2
 [36] 2 2 2 1 1 1 1 1 1 2 1 2 1 2 2 2 1 1 1 1 1 2 2 2 2 1 2 1 1 2 2 1 1 1 1
 [71] 2 1 1 2 2 2 2 1 1 1 2 2 2 1 2 1 1 2 1 2 1 1 2 2 1 2 2 2 2 1 1 2 1 2 2
[106] 2 2 2 2 1 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 2 1 1 2 2 2 1 2 1 1 1 2 1 2
[141] 2 1 2 2 1 2 1 2 1 1 1 1 1 2 1 1 2 1 2 2 2 2 1 2 1 2 1 1 2 2 2 2 2 2 1
[176] 2 2 1 2 1 1 2 1 2 1 1 2 2 1 2 1 2 1 1 1 1 2 1 2 2 2 2 2 2 2 1 1 2 2 1
[211] 2 1 1 2 1 1 1 2 2 1 2 2 1 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 1 1 2 1 2 2 1
[246] 2 2 1 1 2 2 1 1 2 1 1 2 2 1 1 1 2 1 1 1 1 1 1 2 2
```

```
# Estimated probabilities of classification
head(fitted(res_with, type="probability"))
```

```
          class-1   class-2
[1,] 8.261832e-06 0.9999917
[2,] 3.665383e-01 0.6334617
[3,] 8.244708e-01 0.1755292
[4,] 4.443406e-08 1.0000000
[5,] 3.884859e-03 0.9961151
[6,] 4.521816e-02 0.9547818
```

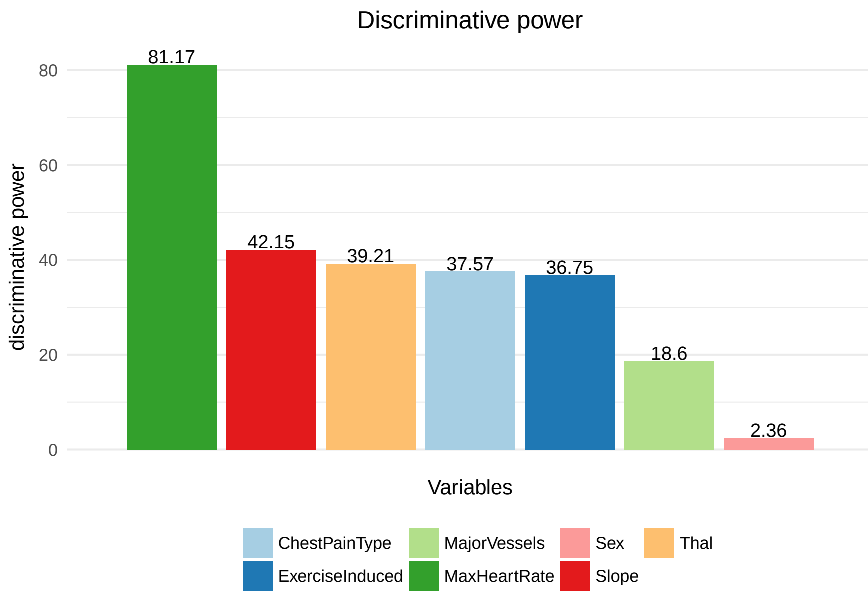
To get a summary of the selected model.

```
# Summary of the best model
summary(res_with)
```

```
Model:
   Number of components: 2
   Model selection has been performed according to the BIC  criterion
   Variable selection has been performed, 8  ( 66.67 % ) of the variables are relevant for clustering
```

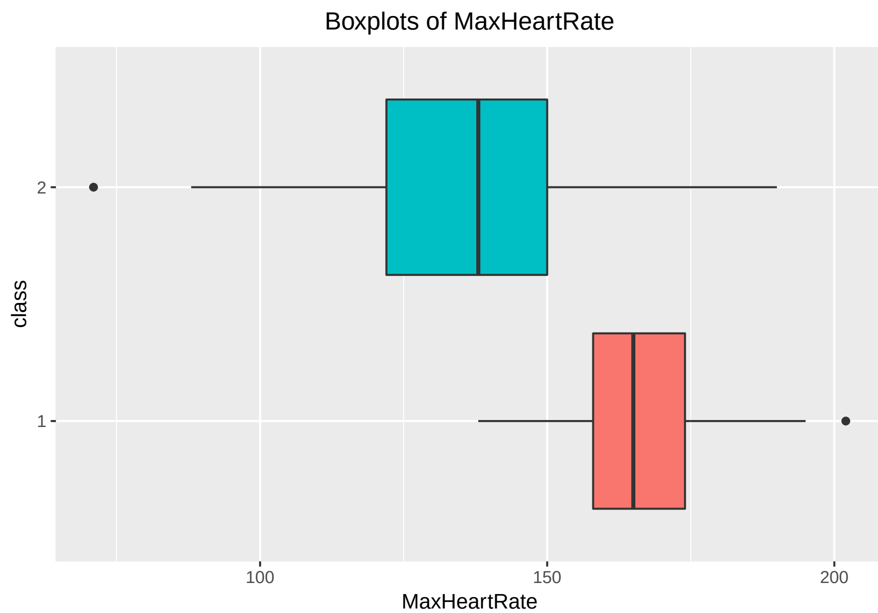
Discriminative power of the variables (here, the most discriminative variable is MaxHeartRate). The greater this index, the more the variable distinguishes the clusters.

```
plot(res_with)
```



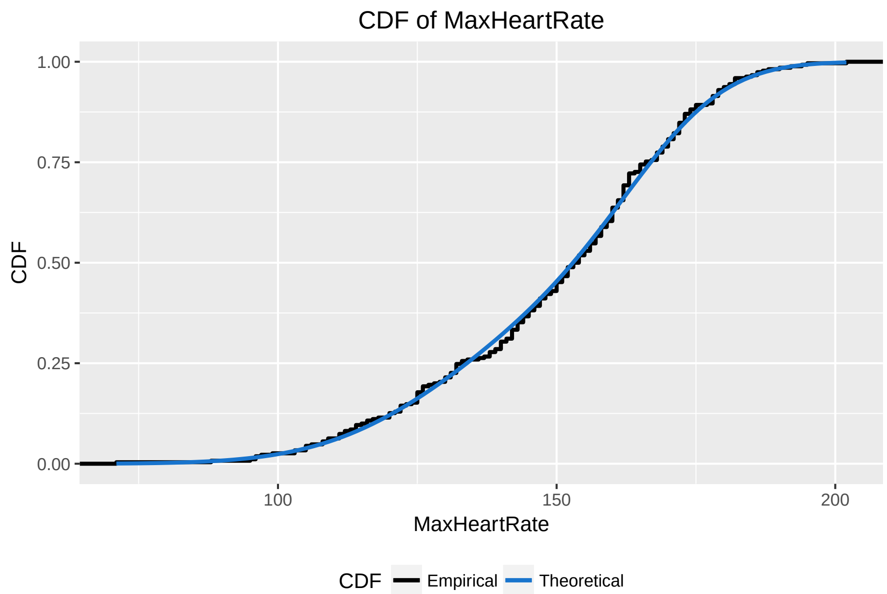
Distribution of the most discriminative variable per clusters

```
# Boxplot for the continuous variable MaxHeartRate
plot(x=res_with, y="MaxHeartRate")
```



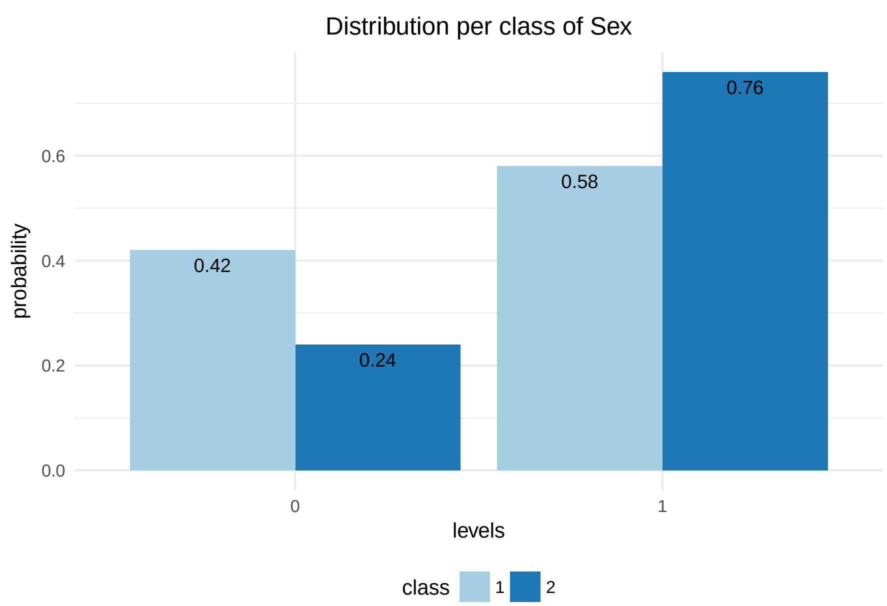
Empirical and theoretical distributions of the most discriminative variable (to check that the distribution is well-fitted)

```
# Empirical and theoretical distributions (to check that the distribution is well-fitted)
plot(res_with, y="MaxHeartRate", type="cdf")
```



Distribution of a categorical variable per clusters

```
# Summary of categorical variable
plot(res_with, y="Sex")
```



To have details about the selected model

```
# More detailed output
print(res_with)
```

```
Data set:
   Number of individuals: 270
   Number of continuous variables: 3
   Number of count variables: 1
   Percentile of missing values for the integer variables: 0.37
   Number of categorical variables: 8
```

```
Model:
   Number of components: 2
   Model selection has been performed according to the BIC  criterion
   Variable selection has been performed, 8  ( 66.67 % ) of the variables are relevant for clustering

Information Criteria:
   loglike: -6403.136
   AIC:     -6441.136
   BIC:     -6509.506
   ICL:     -6638.116
```

To print the parameters

```r
# Print model parameter
coef(res_with)
```

```
An object of class "VSLCMparam"
Slot "pi":
  class-1   class-2
0.4778884 0.5221116

Slot "paramContinuous":
An object of class "VSLCMparamContinuous"
Slot "pi":
numeric(0)

Slot "mu":
                   class-1  class-2
RestBloodPressure 131.3444 131.3444
SerumCholestoral  249.6593 249.6593
MaxHeartRate      165.2587 135.4166

Slot "sd":
                   class-1  class-2
RestBloodPressure 17.82850 17.82850
SerumCholestoral  51.59043 51.59043
MaxHeartRate      13.14847 20.98136


Slot "paramInteger":
An object of class "VSLCMparamInteger"
Slot "pi":
numeric(0)

Slot "lambda":
     class-1  class-2
Age 50.32062 58.11338


Slot "paramCategorical":
An object of class "VSLCMparamCategorical"
Slot "pi":
numeric(0)

Slot "alpha":
```

```
$Sex
                0         1
class-1 0.4166333 0.5833667
class-2 0.2358079 0.7641921

$ChestPainType
                 1          2         3         4
class-1 0.05752308 0.28954358 0.4223080 0.2306254
class-2 0.08922319 0.03291639 0.1738642 0.7039963

$FastingBloodSugar
                0         1
class-1 0.8518519 0.1481481
class-2 0.8518519 0.1481481

$ResElectrocardiographic
                0           1         2
class-1 0.4851852 0.007407407 0.5074074
class-2 0.4851852 0.007407407 0.5074074

$ExerciseInduced
                0          1
class-1 0.9128091 0.08719086
class-2 0.4484663 0.55153370

$Slope
                1         2          3
class-1 0.7599021 0.1933839 0.04671397
class-2 0.2266432 0.6884274 0.08492936

$MajorVessels
                0         1         2          3
class-1 0.7915988 0.1402685 0.0598784 0.00825425
class-2 0.4104423 0.2830470 0.1792855 0.12722522

$Thal
                3            6         7
class-1 0.8302549 6.555078e-11 0.1697451
class-2 0.3183108 9.931182e-02 0.5823774
```

Probabilities of classification for new observations

```r
# Probabilities of classification for new observations
predict(res_with, newdata = x[1:3,])
```

```
          class-1   class-2
[1,] 8.635891e-06 0.9999914
[2,] 3.768792e-01 0.6231208
[3,] 8.307873e-01 0.1692127
```

The model can be used for imputation (of the clustered data or of a new observation)

```r
# Imputation by posterior mean for the first observation
not.imputed <- x[1,]
imputed <- VarSelImputation(res_with, x[1,], method = "sampling")
rbind(not.imputed, imputed)
```

```
  Age Sex ChestPainType RestBloodPressure SerumCholestoral
1  NA   1             4               130              322
2  69   1             4               130              322
  FastingBloodSugar ResElectrocardiographic MaxHeartRate ExerciseInduced
1                 0                       2          109               0
2                 0                       2          109               0
  Slope MajorVessels Thal
1     2            3    3
2     2            3    3
```

Shiny application

All the results can be analyzed by the Shiny application…

```
# Start the shiny application
VarSelShiny(res_with)
```

… but this analysis can also be done on R.