



HAL
open science

Variable selection in model-based clustering and discriminant analysis with a regularization approach

Mohammed Sedki, Gilles Celeux, Cathy Maugis

► **To cite this version:**

Mohammed Sedki, Gilles Celeux, Cathy Maugis. Variable selection in model-based clustering and discriminant analysis with a regularization approach. 2014. hal-01053784v1

HAL Id: hal-01053784

<https://inria.hal.science/hal-01053784v1>

Preprint submitted on 1 Aug 2014 (v1), last revised 17 Apr 2018 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



SelvarMix: A R package for variable selection in model-based clustering and discriminant analysis with a regularization approach.

Mohammed Sedki, Gilles Celeux, Cathy Maugis-Rabusseau

**RESEARCH
REPORT**

N° ????

Juillet 2014

Project-Team Select



SelvarMix: A R package for variable selection in model-based clustering and discriminant analysis with a regularization approach.

Mohammed Sedki ^{*}, Gilles Celeux[†], Cathy Maugis-Rabusseau [‡]

Project-Team Select

Research Report n° 1000 — Juillet 2014 — 17 pages

Abstract: In the R package **SelvarMix**, a regularization approach of variable selection is considered in the model-based clustering and classification frameworks. First, the variables are arranged in order with a lasso-like procedure. Second, the method of Maugis *et al.* (2009b, 2011) is adapted to define the role of variables in the two frameworks. This variable ranking allows us to avoid the painfully slow stepwise algorithms of Maugis *et al.* (2009b). Thus, **SelvarMix** provides a much faster variable selection procedure than Maugis *et al.* (2009b, 2011) and allows us to study high-dimensional datasets.

Key-words: Variable Selection, Lasso Procedure, Gaussian Mixture, Clustering, Classification, R, **SelvarMix**.

* CESP U1018, Bât. 15/16, Hôpital Paul Brousse, 16 av. Paul Vaillant Couturier, F-94807, Villejuif cedex

† INRIA Saclay Île-de-France, Projet SELECT, Bat 425, Université Paris-Sud 11, F-91405 Orsay cedex

‡ Institut de Mathématiques de Toulouse, INSA de Toulouse, 135 avenue de Rangueil, F-31077 Toulouse, cedex

**RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE**

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

SelvarMix: un package R pour la sélection de variables en classification supervisée et non supervisée par modèles de mélange avec une approche de régularisation.

Résumé : Dans le package R **SelvarMix**, nous proposons une approche de régularisation pour la sélection de variables dans le contexte de la classification supervisée et non supervisée par des mélanges gaussiens. Dans une première étape, les variables sont ordonnées à l'aide d'une procédure de régularisation de type lasso. Dans un deuxième temps, nous adaptons les techniques proposées dans Maugis *et al.* (2009b, 2011) pour définir le rôle des variables dans ces deux contextes. Cet ordonnancement des variables nous permet d'éviter l'utilisation des algorithmes pas à pas (*stepwise*) proposés par Maugis *et al.* (2009b), gourmands en temps calcul. Ainsi, la procédure de sélection de variables proposée dans le package **SelvarMix** nous permet de traiter des jeux données en grande dimension avec un gain de temps considérable.

Mots-clés : Sélection de variables, régularisation par lasso, mélange gaussien, classification non supervisée, classification supervisée, R, **SelvarMix**.

1 Introduction

In data mining and statistical learning, available datasets are larger and larger. As a result, there is more and more interest in variable selection procedures for clustering and classification tasks. After a series of papers on variable selection in model-based clustering (Law *et al.*, 2004; Tadesse *et al.*, 2005; Raftery and Dean, 2006; Maugis *et al.*, 2009a), Maugis *et al.* (2009b) proposed a general model for selecting variables for clustering with Gaussian mixtures. This model, called SRUW, distinguishes between relevant variables (S) and irrelevant variables (S^c) for clustering. In addition, the irrelevant variables are divided into two categories. A part of the irrelevant variables (U) may be dependent on a subset R of the relevant variables and an other part (W) are independent of other variables. In Maugis *et al.* (2009b), a procedure using embedded stepwise variable selection algorithms is used to identify the SRUW sets. It leads to compare two models at each step in order to determine which variable should be excluded or included in the set S , R , U or W . But these stepwise procedures implemented in **SelvarClustIndep**, are limited as soon as the number of variables is of the order of a few tens. The SRUW model has been adapted to the Gaussian model-based classification framework in Maugis *et al.* (2011), see also Murphy *et al.* (2010). In this supervised framework, the identification of the sets S , R , U and W is simpler since the stepwise procedures are not performed inside an EM algorithm but the stepwise algorithms still encounters combinatorial explosion phenomenons.

In parallel Pan and Shen (2007) were inspired by the success of the lasso regression to develop a method of variable selection in model-based clustering using ℓ_1 regularization of the likelihood. This approach was successively extended in Xie *et al.* (2008); Wang and Zhu (2008) and finally Zhou *et al.* (2009) proposed a regularized Gaussian mixture model with unconstrained covariance matrices. A first attempt to mix the ranking of the variables through a lasso-like procedure and the clustering via the maximum loglikelihood estimation on Gaussian mixtures have been proposed in Meynet and Maugis-Rabusseau (2012), assuming that the component covariance matrices of Gaussian mixtures were proportional to the identity matrix.

In the present paper, the preliminary work of Meynet and Maugis-Rabusseau (2012) is extended to the SRUW model and more general situations where no restriction is placed on the mixture component covariance matrices. The ℓ_1 penalty is placed on the Gaussian mixture mean vectors and the Gaussian mixture component precision matrices. This is made feasible by exploiting the abundant literature on lasso penalization in Gaussian graphical models (see Friedman *et al.*, 2007; Meinshausen and Bühlmann, 2006). Using the resulting ranking of the variables avoids the need to use stepwise variable selection algorithms and combinatorial problems. And, it is hoped that using this lasso-like ranking of the variables instead of stepwise algorithms would not deteriorate the identification of the sets S , R , U and W .

This article is organized as follows. In Section 2, the SRUW model is reviewed in the Gaussian model-based clustering context and its simple extension to the Gaussian model-based classification context is sketched. The variable selection procedure using lasso-like penalization is presented in Section 3. Again, we concentrate on the clustering framework on the mixture model for clustering with unknown mixture labels and present rapidly the extension of the regularization approach to the supervised classification problem. The R package **SelvarMix** is presented in Section 4 and Section 5 is devoted to the comparison of the procedures **SelvarClustIndep** and **SelvarMix** on several simulated datasets. A short discussion section ends the article.

2 The SRUW model

2.1 Model-based clustering

Let n observations $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)'$ be described by p continuous variables ($\mathbf{y}_i \in \mathbb{R}^p$). In the model-based clustering framework, the multivariate continuous data \mathbf{y} are assumed to come from several subpopulations (clusters) modeled with a multivariate Gaussian density. The observations are assumed to arise from a finite Gaussian mixture with K components and a mixture form m , namely

$$f(\mathbf{y}_i | K, m, \alpha) = \sum_{k=1}^K \pi_k \phi(\mathbf{y}_i | \mu_k, \Sigma_{k(m)}),$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ is the mixing proportion vector ($\pi_k \in (0, 1)$ for all $k = 1, \dots, K$ and $\sum_{k=1}^K \pi_k = 1$), $\phi(\cdot | \mu_k, \Sigma_k)$ is the p -dimensional Gaussian density function with mean μ_k and covariance matrix Σ_k , and $\alpha = (\boldsymbol{\pi}, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K)$ is the parameter vector. Several Gaussian mixture forms m are available, each corresponding to different assumptions on the forms of the covariance matrices, arising from a modified spectral decomposition. These include whether the volume, shape and orientation of each mixture component vary between components or are constant across clusters (Banfield and Raftery, 1993; Celeux and Govaert, 1995).

Typically, the mixture parameters are estimated via maximum likelihood using the EM algorithm (Dempster *et al.*, 1977), and both the number of components K and the mixture form m are chosen using the Bayesian Information Criterion (BIC) (Schwarz, 1978) or other penalized likelihood criteria as the Integrated Completed Likelihood (ICL) criterion (Biernacki *et al.*, 2000) in the clustering context. Among the R packages which implement this methodology, we could mention the **mclust** software (<http://www.stat.washington.edu/mclust/>), and the **Rmixmod** software (<http://www.mixmod.org>).

2.2 Variable selection in model-based clustering

The SRUW model, as described in Maugis *et al.* (2009b), involves three possible roles for the variables: the relevant clustering variables (S), the redundant variables (U) and the independent variables (W). Moreover, the redundant variables U are explained by a subset R of the relevant variables S , while the variables W are assumed to be independent of the relevant variables. Thus the data density is assumed to be decomposed into three parts as follows:

$$f(\mathbf{y}_i | K, m, r, \ell, \mathbf{V}, \theta) = \sum_{k=1}^K \pi_k \phi(\mathbf{y}_i^S | \mu_k, \Sigma_{k(m)}) \times \phi(\mathbf{y}_i^U | a + \mathbf{y}_i^R b, \Omega_{(r)}) \times \phi(\mathbf{y}_i^W | \gamma, \tau_{(\ell)})$$

where $\theta = (\alpha, a, b, \Omega, \gamma, \tau)$ is the full parameter vector and $\mathbf{V} = (S, R, U, W)$. The form of the regression covariance matrix Ω is denoted by r ; it can be spherical, diagonal or general. The form of the covariance matrix τ of the independent variables W is denoted by ℓ and can be spherical or diagonal.

The SRUW model recasts the variable selection problem for model-based clustering as a model selection problem, where the model collection is indexed by $(K, m, r, \ell, S, R, U, W)$. This model selection problem is solved maximizing the following BIC-type criterion:

$$\text{crit}(K, m, r, \ell, \mathbf{V}) = \text{BIC}_{\text{clust}}(\mathbf{y}^S | K, m) + \text{BIC}_{\text{reg}}(\mathbf{y}^U | r, \mathbf{y}^R) + \text{BIC}_{\text{indep}}(\mathbf{y}^W | \ell), \quad (1)$$

where $\text{BIC}_{\text{clust}}$ represents the BIC criterion of the Gaussian mixture model with the variables S , BIC_{reg} represents the BIC criterion of the regression model of the variables U on the variables R and $\text{BIC}_{\text{indep}}$ represents the BIC criterion of the Gaussian model with the variables W .

Since the SRUW model collection is large, two embedded backward or forward stepwise algorithms for variable selection, one for the clustering and one for the linear regression, are considered to solve this model selection problem. A backward algorithm allows one to start with all variables in order to take variable dependencies into account. A forward procedure, starting with an empty clustering variable set or a small variable subset, could be preferred for numerical reasons if the number of variables is large. The method is implemented in the **SelvarClustIndep** software.¹ But in a high-dimensional setting, even the variable selection method with the two forward stepwise algorithms becomes painfully slow and alternative methods are desirable.

2.3 Variable selection in Classification

In the supervised classification framework, the labels of the training dataset are known and the variable selection problem is analogous but simpler than in the clustering framework. The training data are composed by n i.i.d vectors

$$(\mathbf{y}, \mathbf{z}) = \{(\mathbf{y}_1, z_1), \dots, (\mathbf{y}_n, z_n); \mathbf{y}_i \in \mathbb{R}^p, z_i \in \{1, \dots, K\}\},$$

where \mathbf{y}_i is the p -dimensional predictor and z_i is the class label of the i th observation. The number of classes K is known. The subset S is now the discriminant variable subset. Under a model (m, r, ℓ, \mathbf{V}) with $\mathbf{V} = (S, R, U, W)$, the distribution of the training sample is modeled by

$$\left\{ \begin{array}{l} f(\mathbf{y}_i | z_i = k, m, r, \ell, \mathbf{V}) = \phi(\mathbf{y}_i^S | \mu_k, \Sigma_{k(m)}) \phi(\mathbf{y}_i^U | a + \mathbf{y}_i^R \beta, \Omega_{(r)}) \phi(\mathbf{y}_i^W | \gamma, \tau_{(\ell)}) \\ (\mathbb{1}_{z_i=1}, \dots, \mathbb{1}_{z_i=K}) \sim \text{Multinomial}(1; \pi_1, \dots, \pi_K) \end{array} \right.$$

According to the assumed form of the covariances matrices involving their eigenvalue decomposition, a collection of more or less parsimonious mixture forms (m) is available as in the clustering context.

Considering the same model SRUW, the variable selection problem is solved by using the following model selection criterion

$$\text{crit}(m, r, \ell, \mathbf{V}) = \text{BIC}_{\text{clas}}(\mathbf{y}^S, \mathbf{z} | m) + \text{BIC}_{\text{reg}}(\mathbf{y}^U | r, \mathbf{y}^R) + \text{BIC}_{\text{indep}}(\mathbf{y}^W | \ell),$$

where BIC_{clas} denotes the BIC criterion for the Gaussian classification on the discriminant variable subset S . Maximizing this criterion is an easier task in this supervised context since there is no need to use the EM algorithm to derive the parameter estimates of the Gaussian classification model contrary to the model-based clustering situation. But, the variable selection procedure with two embedded stepwise procedures remain expensive and alternative procedures are desirable.

3 Variable selection through regularization

In order to avoid the highly CPU-time consuming of stepwise algorithms **SelvarClustIndep**, we propose an alternative variable selection procedure in two steps: First, the variables are ranked through a lasso-like procedure, and second, the variable roles are determined using the criterion (1) on these ranked variables. The procedure is detailed in the clustering framework, the simplifications for classification are presented in Section 3.3. This variable selection procedure is implemented in the R package **SelvarMix**.

¹ *SelvarClustIndep* is available at <http://www.math.univ-toulouse.fr/~maugis/>

3.1 Variable ranking by regularization

In the first step, the variables are ranked through the lasso-like procedure of Zhou *et al.* (2009). For any $K \in \mathbb{N}^*$, the criterion to be maximized is

$$\alpha \mapsto \sum_{i=1}^n \ln \left[\sum_{k=1}^K \pi_k \phi(\bar{\mathbf{y}}_i \mid \mu_k, \Sigma_k) \right] - \lambda \sum_{k=1}^K \|\mu_k\|_1 - \rho \sum_{k=1}^K \|\Sigma_k^{-1}\|_1, \quad (2)$$

where

$$\|\mu_k\|_1 = \sum_{j=1}^p |\mu_{kj}|, \quad \|\Sigma_k^{-1}\|_1 = \sum_{\substack{v,j=1 \\ v \neq j}}^p |(\Sigma_k^{-1})_{vj}|,$$

$\bar{\mathbf{y}}_i = (y_{ij} - \bar{y}_j)_{1 \leq j \leq p}$ with $\bar{y}_j = \frac{1}{n} \sum_{i=1}^n y_{ij}$ and where λ and ρ are two non negative regularization parameters defined on two grids of values \mathcal{G}_λ and \mathcal{G}_ρ respectively. The estimated mixture parameters for fixed tuning parameters λ and ρ ,

$$\hat{\alpha}(\lambda, \rho) = (\hat{\boldsymbol{\pi}}(\lambda, \rho), \hat{\boldsymbol{\mu}}_1(\lambda, \rho), \dots, \hat{\boldsymbol{\mu}}_K(\lambda, \rho), \hat{\boldsymbol{\Sigma}}_1(\lambda, \rho), \dots, \hat{\boldsymbol{\Sigma}}_K(\lambda, \rho))$$

are computed with the EM algorithm of Zhou *et al.* (2009). In particular, the glasso algorithm (Friedman *et al.*, 2007) using a coordinate descent procedure for the lasso is used to estimate sparse precision matrices Σ_k^{-1} . This procedure is reminded in Appendix A.1.

It is worth noting that this lasso-like criterion does not take into account the typology of the variables induced by the SRUW model. Strictly speaking, it only distinguishes two possible roles for the variables: a variable is declared related or independent of the clustering. A variable is declared independent for the clustering if for all $j = 1, \dots, p$, and $k = 1, \dots, K$, $\hat{\mu}_k(\lambda, \rho) = 0$. The variance matrices are not considered in this definition. Actually, their role is secondary in clustering and taking them into account would imply serious numerical difficulties.

Varying the regularization parameters (λ, ρ) in $\mathcal{G}_\lambda \times \mathcal{G}_\rho$, a score is defined for each variable $j \in \{1, \dots, p\}$ and for fixed K by

$$\mathcal{O}_K(j) = \sum_{(\lambda, \rho) \in \mathcal{G}_\lambda \times \mathcal{G}_\rho} \mathcal{B}_{(K, \rho, \lambda)}(j)$$

where

$$\mathcal{B}_{(K, \rho, \lambda)}(j) = \begin{cases} 0 & \text{if } \hat{\mu}_{1j}(\lambda, \rho) = \dots = \hat{\mu}_{Kj}(\lambda, \rho) = 0 \\ 1 & \text{else.} \end{cases}$$

The larger $\mathcal{O}_K(j)$ the more related for the clustering the variable j is expected to be. The variables are thus ranked by their decreasing values on $\mathcal{O}_K(j)$. This variable ranking is denoted $\mathcal{I}_K = (j_1, \dots, j_p)$ with $\mathcal{O}_K(j_1) > \mathcal{O}_K(j_2) > \dots > \mathcal{O}_K(j_p)$.

3.2 Determination of the variable roles

The relevant clustering variable set $\mathcal{S}_{(K, m)}$ is determined first. The variable set is scanned according to the \mathcal{I}_K order. One variable is added to $\mathcal{S}_{(K, m)}$ if

$$\begin{aligned} \text{BIC}_{\text{diff}}(j_v) &= \text{BIC}_{\text{clust}}(\mathbf{y}^{\mathcal{S}_{(K, m)}}, \mathbf{y}^{j_v} \mid K, m) \\ &\quad - \text{BIC}_{\text{clust}}(\mathbf{y}^{\mathcal{S}_{(K, m)}} \mid K, m) - \text{BIC}_{\text{reg}}(\mathbf{y}^{j_v} \mid \mathbf{y}^{R[j_v]}) \end{aligned} \quad (3)$$

is positive, $R[j_v]$ being the variables of $S_{(K,m)}$ required to linearly explain \mathbf{y}^{j_v} . This subset $R[j_v]$ is determined using the backward stepwise algorithm for the variable selection in the linear regression using in **SelvarClustIndep**. The scanning of \mathcal{I}_K is stopped as soon as c successive variables have a non positive BIC_{diff} value, c being a fixed positive integer. Once the relevant variable set $S_{(K,m)}$ is determined, the independent variable set W is determined as follows. Scanning the variable set according to the reverse order of I_K , a variable j_v is added to $W_{(K,m)}$ if the subset $R[j_v]$ of $S_{(K,m)}$ (derived from the backward stepwise algorithm) is empty. The algorithm stops as soon as c successive variables are not declared independent. The redundant variables are thus declared to be $U_{(K,m)} = \{1, \dots, p\} \setminus \{S_{(K,m)} \cup W_{(K,m)}\}$ and the subset $R_{(K,m,r)}$ of $S_{(K,m)}$ required to linearly explain $\mathbf{y}^{U_{(K,m)}}$ is derived from the backward stepwise algorithm, for each covariance shape r . The ideal position of the variable sets S , U and W in the variable ranking is schematised in Figure 1. Finally, the model $(\hat{K}, \hat{m}, \hat{r}, \hat{\ell})$ maximizing the criterion $\text{crit}(K, m, r, \ell, \mathbf{V}_{(K,m,r)})$ defined in Equation (1) with $\mathbf{V}_{(K,m,r)} = (S_{(K,m)}, R_{(K,m,r)}, U_{(K,m)}, W_{(K,m)})$ is selected.

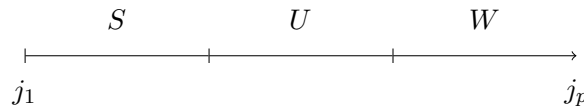


Figure 1: The ideal position of the sets S , U and W in the ranking of the variables

Some comments are in order:

- It is possible to use a lasso procedure instead of the stepwise variable selection algorithm in the linear regression step. However, this replacement is not expected to be highly beneficial since stepwise variable selection in linear regression are not too much expensive and, moreover, the number of variables in the set S is not expected to be high.
- There is no guarantee that the variable order designed in 3.1 would be in accordance with the ideal ranking of the variables displayed in Figure 1. In particular when the variables are highly correlated, lasso-like procedures could be expected to produce confusion between the sets S and U . This is the reason why we wait $c (> 1)$ few steps before deciding the variable roles: we give a chance to the procedure to catch more variables in S and in W .

3.3 Variable selection through regularization for classification

In the classification context, K is fixed and the regularization criterion to be maximized is

$$\sum_{i=1}^n \sum_{k=1}^K \mathbb{1}_{\{z_i=k\}} \ln \left[\pi_k \phi(\bar{\mathbf{y}}_i \mid \mu_k, \Sigma_k) \right] - \lambda \sum_{k=1}^K \|\mu_k\|_1 - \rho \sum_{k=1}^K \|\Sigma_k^{-1}\|_1, \quad (4)$$

with the same notation as Section 3.1. Assuming that the training dataset has been obtained according to the mixture sampling scheme, the proportions π_1, \dots, π_K are estimated by

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{z_i=k\}}, \quad k = 1, \dots, K.$$

RR n° ????

The maximization of the criterion (4) is done according to the procedure described in Appendix A.2. The only difference with the clustering context is that the labels z_i are known and no EM algorithm is required. Thus, a ranking \mathcal{L}_K of the variables is get and the procedure described in Section 3.2 for model-based clustering is adapted straightforwardly to the supervised classification context, where (3) is replaced by

$$\text{BIC}_{\text{diff}}(j_v) = \text{BIC}_{\text{clas}}(\mathbf{y}^{S(m)}, \mathbf{y}^{j_v}, \mathbf{z} \mid m) - \text{BIC}_{\text{clas}}(\mathbf{y}^{S(m)}, \mathbf{z} \mid m) - \text{BIC}_{\text{reg}}(\mathbf{y}^{j_v} \mid \mathbf{y}^{R[j_v]}).$$

4 Using the package SelvarMix

In this section, the **SelvarMix** functions are illustrated. The simulated example as shown in Maugis *et al.* (2009b) consists of $n = 2000$ observations described by $p = 14$ variables. On the subset of relevant clustering variables $S = \{1, 2\}$, data are distributed from a mixture of four equiprobable spherical Gaussian distributions. The subsets of redundant variables is $U = \{3 - 11\}$, regressed on the first two variables ($R = S$) and the last three variables are independent $W = \{12 - 14\}$. This example is detailed in Section 5.

```
R> data(scenarioCor)
R> data.cor <- scenarioCor[,1:14]
```

In the clustering framework, the variable selection procedure presented in Section 3 is implemented in the function **SelvarClustLasso**. This function requires a grid for both regularization parameters λ and ρ and a range of values for the number of clusters K . Other arguments can be modified by the user as the threshold c ("hybrid.size"), the Gaussian mixture form ("models") using the **mixmodGaussianModel** function of the **Rmixmod** software, the model selection criterion (BIC and/or ICL can be used; in the clustering context, $\text{BIC}_{\text{clust}}$ can be replaced by ICL in Equation (3)),...

```
R> lambda <- seq(0.1, 100, length = 25)
R> rho <- seq(1, 2, length=2)
R> nbCluster <- c(3,4)
R> simulate.clust <- SelvarClustLasso(data.cor, nbCluster, lambda, rho)
```

For each considered model selection criterion, the function returns the selected role of variables, the selected number of clusters, and the data clustering among other outputs:

```
R> simulate.clust$BIC$S
[1] 1 2

R> simulate.clust$BIC$R
[1] 1 2

R> simulate.clust$BIC$U
[1] 3 4 5 6 7 8 9 10 11

R> simulate.clust$BIC$W
[1] 12 13 14

R> simulate.clust$BIC$nbCluster
[1] 4
```

```
R> tabulate(simulate.clust$BIC$partition)
[1] 480 591 468 461
```

It is possible to run the `SortvarClust` function in order to simply get the variable ranking based on the lasso procedure described in Section 3.1. The user has to specify a grid for the regularization parameters λ and ρ , and the values for the number of clusters K . For each value of K , the variable ranking \mathcal{I}_K is provided.

```
R> lambda <- seq(0.1, 100, length = 50)
R> rho <- seq(1, 2, length=2)
R> nbCluster <- c(3,4)
R> SortvarClust(data.cor, nbCluster, lambda, rho)
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,]    1    4    6    7    9   10   11    3    8    5    2   12   14   13
[2,]    1    2    4    5    6    7    9   10   11    3    8   12   14   13
```

To illustrate the variable selection procedure in the classification framework, the previous dataset is split into a training sample ($n = 1900$ observations) and a test sample ($n = 100$ observations).

```
R> data(scenarioCor)
R> ## training sample : n = 1900 observations , p = 14 variables
R> data.learn <- scenarioCor[1:1900,1:14]
R> labels.learn <- scenarioCor[1:1900,15]
R> ## testing sample : n = 100 observations, p = 14 variables
R> data.test <- scenarioCor[1901:2000,1:14]
R> labels.test <- scenarioCor[1901:2000,15]
```

The variable selection procedure for model-based classification is available with the `SelvarLearnLasso` function. A grid for each regularization parameter (λ and ρ) is required.

```
R> lambda <- seq(0.1, 100, length = 25)
R> rho <- seq(1, 2, length=2)
R> simulate.da <- SelvarLearnLasso(data.learn, labels.learn, lambda, rho,
                                data.test, labels.test)
```

This function returns the selected role of variables as in the clustering framework. If the function is used with a training sample and a test sample, it returns the conditional probabilities of the classes (`simulate.da$proba`) and the classification of the test sample (`simulate.da$partition`) and the resulting classification error rate (`simulate.da$error`). If the function is only used with a training sample, these three outputs are not available.

As in the clustering framework, it is possible to only consider the variable ranking for the supervised classification using the `SortvarLearn` function.

```
R> SortvarLearn(data.learn, labels.learn, lambda, rho)
[1] 1 2 3 4 5 6 7 8 9 10 11 12 14 13
```

5 Numerical experiments

This section is devoted to comparing our R package `SelvarMix` with the forward/backward stepwise procedures of Maugis *et al.* (2009b, 2011) in both model-based clustering and classification settings.

RR n° ????

5.1 Model-based clustering

We consider one of the seven simulated datasets studied in Maugis *et al.* (2009b, section 6.1). The data consist of $n = 2000$ observations on $p = 14$ variables. On the first two variables ($S = \{1, 2\}$), data are distributed from an equiprobable mixture of four Gaussian distributions $\mathcal{N}(\mu_k, I_2)$ with $\mu_1 = (0, 0)$, $\mu_2 = (4, 0)$, $\mu_3 = (0, 2)$ and $\mu_4 = (4, 2)$. On the nine redundant variables ($U = \{3, \dots, 11\}$), data are simulated as follows: for $i = 1, \dots, n$,

$$\mathbf{y}_i^{\{3-11\}} = (0, 0, 0.4, 0.8, \dots, 2) + \mathbf{y}_i^{\{1,2\}}b + \varepsilon_i$$

where the regression coefficients are

$$b = ((0.5, 1)', (2, 0)', (0, 3)', (-1, 2)', (2, -4)', (0.5, 0)', (4, 0.5)', (3, 0)', (2, 1)')$$

and ε_i are i.i.i $\mathcal{N}(0_9, \Omega)$. The regression covariance matrix $\Omega = \text{diag}(I_3, 0.5I_2, \Omega_1, \Omega_2)$ is block diagonal with $\Omega_1 = \text{Rot}(\frac{\pi}{3})' \text{diag}(1, 3) \text{Rot}(\frac{\pi}{3})$ and $\Omega_2 = \text{Rot}(\frac{\pi}{6})' \text{diag}(2, 6) \text{Rot}(\frac{\pi}{6})$, where $\text{Rot}(\theta)$ is a plane rotation matrix with angle θ . The last three independent variables are standard Gaussian random variables $\mathbf{y}_i^{\{12-14\}} \sim \mathcal{N}(0_3, I_3)$.

In the first scenario, the function **SelvarClustLasso** of **SelvarMix** and the forward procedure of Maugis *et al.* (2009b) (**SelvarClustIndep**) are compared on 100 replications of the simulated dataset with $K = 4$ and spherical mixture components. We compare the CPU times of both procedures. The calculations are carried out on an 80 Intel Xeon 2.4 GHz processors machine and the variable ranking procedure (see Section 3.2) of **SelvarMix** is parallelized. In comparison to **SelvarMix**, the combinatorial nature of **SelvarClustIndep** makes it difficult to be parallelized. As a result, a significant improvement of the runtime with **SelvarMix** is obtained: **SelvarMix** takes 46.995(± 3.2519) CPU time whereas **SelvarClustIndep** needs 450.64(± 104.03). Figure 2 displays the distribution of the variable roles with **SelvarClustIndep** in left and **SelvarMix** in right. Globally, the true variable roles are well recovered. Surprisingly **SelvarMix** detects the relevant variables better than **SelvarClustIndep** which sometimes selects variables 5 and 9 instead of the first two variables.

In the second scenario, the 50 previous simulated datasets are considered but now the number of clusters K varies between 2 to 6 and the 28 Gaussian mixture forms m are considered. The true cluster number is always correctly selected by both procedures. The variable selection is similar with both procedures (see Figure 3): the true variable partition is selected 46 (resp. 48) times by **SelvarClustIndep** (resp. **SelvarMix**). The clustering performance is preserved with **SelvarMix** since the average of adjusted rand index (ARI) is 0.5996(± 0.0179) with **SelvarMix** and 0.5989(± 0.0154) with **SelvarClustIndep**.

We also applied the Zhou *et al.* (2009)'s lasso-like procedure on the 50 simulated datasets. As previously, the number of clusters varies from 2 to 6. The penalization parameters, including the means $\hat{\mu}_1, \dots, \hat{\mu}_K$, are estimated according to the penalized likelihood criterion (2) whereas the number of clusters K is selected using the BIC criterion. One variable j is declared relevant if there is at least one cluster k where $|\hat{\mu}_{kj}| > 10^{-1}$. For all 50 simulated datasets, the Zhou *et al.* (2009)'s procedure fails to select the true number of clusters ($K = 4$) and the true set of relevant variables ($S = \{1, 2\}$). It always selects $K = 6$, both relevant and redundant variables are declared relevant and the ARI is lower (0.4507(± 0.0253)).

In the third scenario, we consider 50 datasets consisting of $n = 400$ observations described by 100 variables. On the first eleven variables, data are distributed as previously. Next, 89 standard Gaussian variables are appended. As previously, the number of clusters varies from 2 to 6 and the 28 Gaussian mixtures forms m are considered. The selection of the number of clusters is less efficient: the true cluster number $K = 4$ is selected 23 times, the model with $K = 3$ is

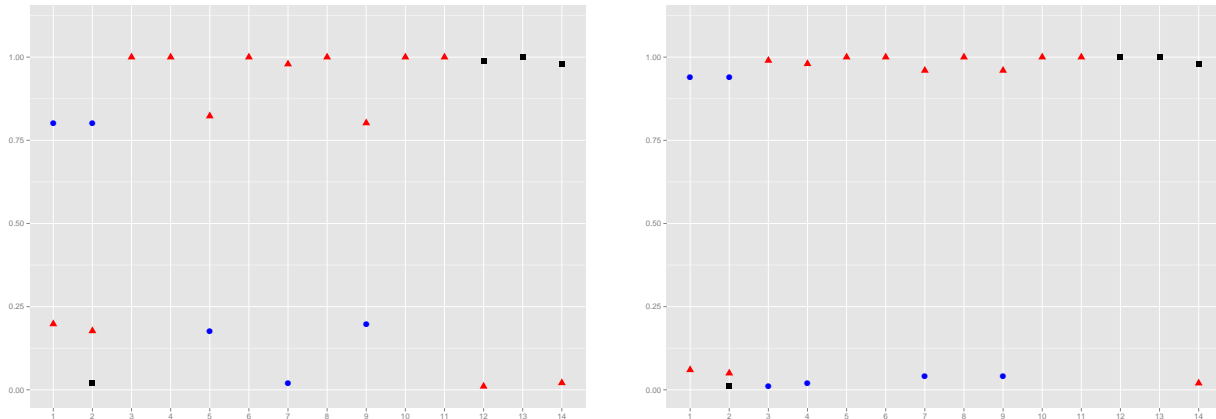


Figure 2: Proportion of times each variable was declared relevant (square), redundant (triangle) or independent (circle) by **SelvarClustIndep** (left) and **SelvarMix** (right) in the first scenario. Zero values are not shown.

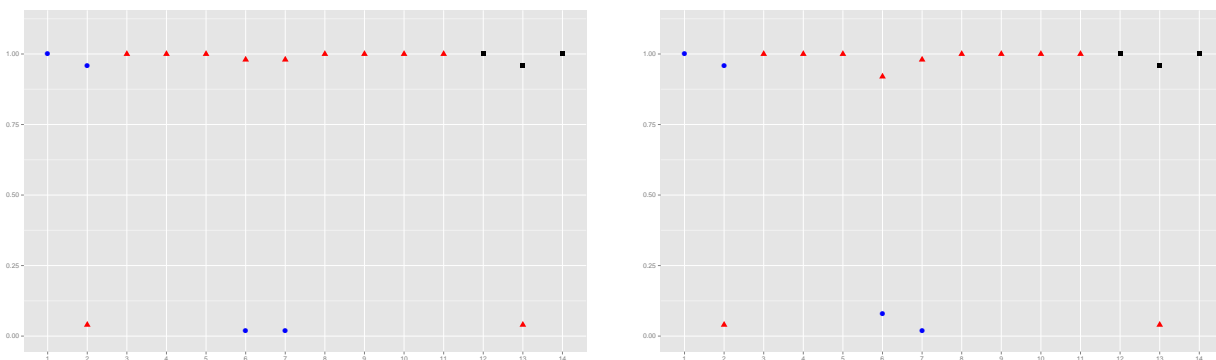


Figure 3: Proportion of times each variable was declared relevant (square), redundant (triangle) or independent (circle) by **SelvarClustIndep** (left) and **SelvarMix** (right) in the second scenario. Zero values are not shown.

RR n° ????

selected 10 times whereas the models with $K = 2, 5$ and 6 are respectively selected $6, 7$ and 4 times. Compared to the previous scenario in low dimension ($p = 14$), the variable selection using **SelvarMix** is slightly deteriorated (see Figure 4). The true relevant variable set $S = \{1, 2\}$ is selected 23 times. Sometimes, **SelvarMix** declares one of the redundant variables as relevant. One of the independent variables is declared as relevant seven times. Moreover, the clustering performance is slightly deteriorated: the ARI is $0.49(\pm 0.0993)$.

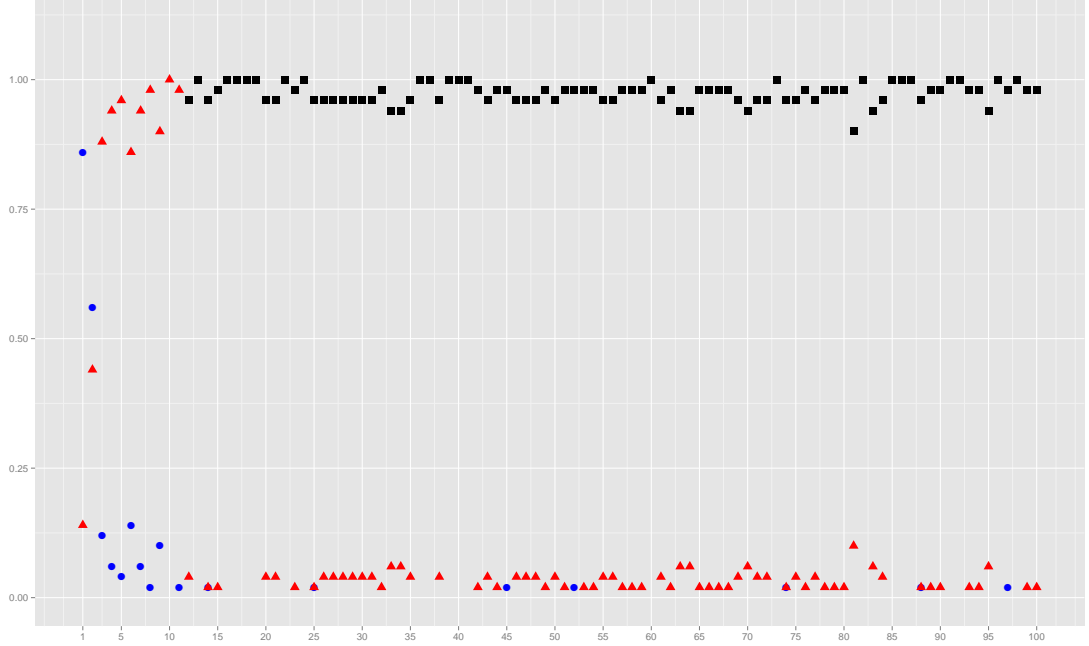


Figure 4: Proportion of times each variable was declared discriminant (square), redundant (triangle) or independent (circle) by **SelvarMix** in the clustering setting, for the datasets with $p = 100$ variables. Zero values are not shown.

5.2 Supervised Classification

We consider the same simulated example of Maugis *et al.* (2011). The samples are described by $p = 16$ variables. The prior probabilities of the four classes are $\boldsymbol{\pi} = (0.15, 0.3, 0.2, 0.35)$. On the three discriminant variables, data are distributed from $\mathbf{y}_i^{\{1-3\}} \mid z_i = k \sim \mathcal{N}(\mu_k, \Sigma_k)$ with means vectors $\mu_1 = (1.5, -1.5, 1.5)$, $\mu_2 = (-1.5, 1.5, 1.5)$, $\mu_3 = (1.5, -1.5, -1.5)$ and $\mu_4 = (-1.5, 1.5, -1.5)$. The covariance matrices are $\Sigma_k = \left(\rho_k^{|i-j|}\right)$ with $\rho_1 = 0.85, \rho_2 = 0.1, \rho_3 = 0.65$ and $\rho_4 = 0.5$. There are four redundant variables simulated from

$$\mathbf{y}_i^{\{4-7\}} \sim \mathcal{N}\left(\mathbf{y}_i^{\{1-3\}}\boldsymbol{\beta} = \begin{pmatrix} 1 & 0 & -1 & 2 \\ 0 & -2 & 2 & 1 \end{pmatrix}, I_4\right).$$

Nine independent variables are appended, sampled from $\mathbf{y}_i^{\{8-16\}} \sim \mathcal{N}(\boldsymbol{\gamma}, \tau)$ with

$$\boldsymbol{\gamma} = (-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2)$$

and the diagonal matrix

$$\tau = \text{diag}(0.5, 0.75, 1, 1.25, 1.5, 1.25, 1, 0.75, 0.5).$$

First, 100 simulation replications are considered where the training sample is composed of $n = 500$ observations and the same test sample with 50000 points is used. The fourteen forms m are considered. The two procedures **SelvarMix** and the version of **SelvarClustIndep** devoted to the variable selection in supervised classification (which is again called **SelvarClustIndep** in the sequel) are compared. Figure 5 shows the variable selection obtained with the procedure of Maugis *et al.* (2011) on the left and the **SelvarLearnLasso** function of **SelvarMix** on the right. **SelvarMix** sometimes declares Variables 6 and 7 as relevant with the first three relevant variables and has some tendency to declare redundant some independent variables, more often than **SelvarClustIndep**. For the prediction, both procedures have a similar behaviour: the misclassification error rate is 4.5% (± 0.19) for **SelvarMix** and 4.18% (± 0.06) for **SelvarClustIndep**.

Second, we consider 100 training samples composed of $n = 500$ observations described by $p = 100$ variables, 84 standard Gaussian variables are appended to the previous training samples. The test sample is similarly modified. As expected, **SelvarMix** allows us to quickly study such datasets where the number of variables is large. The level of prediction is preserved since the misclassification error rate is 4.34% (± 0.18) and the variable selection is similar as the first scenario as shown in Figure 6.

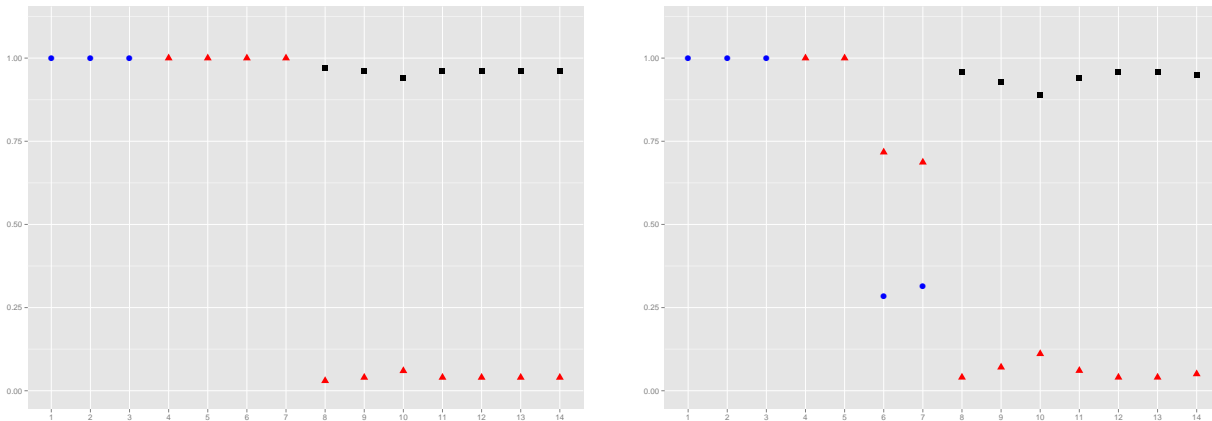


Figure 5: Proportion of times each variable was declared discriminant (square), redundant (triangle) or independent (circle) by **SelvarClustIndep** (left) and **SelvarMix** (right) in the classification setting, for the datasets with $p = 16$ variables. Zero values are not shown.

6 Discussion

The model SRUW is a powerful model to define the roles of variables in the Gaussian model-based clustering and classification contexts. However the diffusion of this model is slowed down by the stepwise selection algorithms used in its previous implementations. Despite they are sub-optimal these algorithms are highly CPU time demanding.

The regularization approach we propose allows us to avoid these stepwise procedures by designing an irreversible order on which the variables in S , U and W are chosen. The numerical

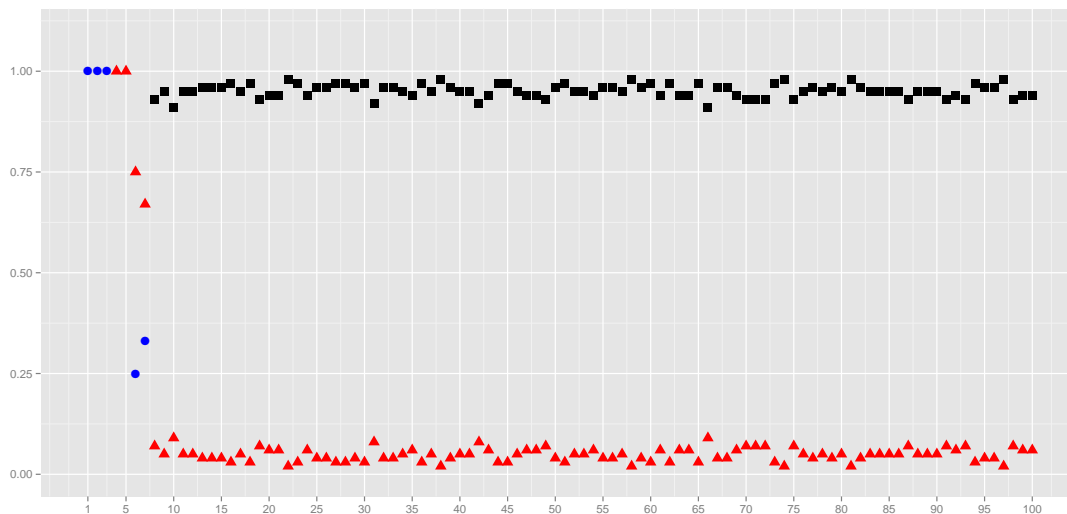


Figure 6: Proportion of times each variable was declared discriminant (square), redundant (triangle) or independent (circle) by **SelvarMix** in the classification setting, for the datasets with $p = 100$ variables. Zero values are not shown.

experiments performed with the resulting R package **SelvarMix** show quite encouraging performances. **SelvarMix** is highly faster than **SelvarClustIndep** while providing analogous (and sometimes better) performances than the reference **SelvarClustIndep** program.

In practice the influence of the number c for which the regularization algorithm provides the same role for c successive variables could be regarded as an important tuning parameter. In our numerical experiments, this parameter appears to be not too much sensitive and a default value $c = 3$ seems reasonable. However the influence of c needs to be further investigated in order to propose heuristic rules to choose it as a simple function of the total number of variables in a proper way to get stable selections.

Acknowledgments

The first author was supported by Fondation de Coopération Scientifique Campus Paris-Saclay-DIGITEO during his one year post-doc at INRIA Saclay Île-de-France. This work was partially supported by the French Agence Nationale de la Recherche (ANR), under grant MixStatSeq (ANR-13-JS01-0001-01).

A Procedures to maximize penalized empirical contrasts

A.1 The model-based clustering case

The EM algorithm for maximizing Criterion (2) is as follows (Zhou *et al.*, 2009). The penalized complete loglikelihood of the centered dataset $\bar{\mathbf{y}} = (\bar{\mathbf{y}}_1, \dots, \bar{\mathbf{y}}_n)'$ is given by

$$L_{c,(\lambda,\rho)}(\bar{\mathbf{y}}, \mathbf{z}, \alpha) = \sum_{i=1}^n \sum_{k=1}^K z_{ik} [\ln(\pi_k) + \ln \phi(\bar{\mathbf{y}}_i | \mu_k, \Sigma_k)] - \lambda \sum_{k=1}^K \|\mu_k\|_1 - \rho \sum_{k=1}^K \|\Theta_k\|_1 \quad (5)$$

where $\Theta_k = \Sigma_k^{-1}$ denotes the precision matrix of the k -th mixture component. The EM algorithm of Zhou *et al.* (2009) maximizes at each iteration the conditional expectation of (5) given $\bar{\mathbf{y}}$ and a current parameter vector $\alpha^{(s)}$: $\mathbb{E} \left[L_{c,(\lambda,\rho)}(\bar{\mathbf{y}}, \mathbf{z}, \alpha) | \bar{\mathbf{y}}, \alpha^{(s)} \right]$. The following two steps are repeated from an initial $\alpha^{(0)}$ until convergence. At the s -th iteration of the EM algorithm:

- **E-step:** The conditional probabilities $t_{ik}^{(s)}$ that the i -th observation belongs to the k -th cluster are computed for $i = 1, \dots, n$ and $k = 1, \dots, K$,

$$t_{ik}^{(s)} = \mathbb{P}(z_{ik} = 1 | \bar{\mathbf{y}}, \alpha^{(s)}) = \frac{\pi_k^{(s)} \phi(\bar{\mathbf{y}}_i | \mu_k^{(s)}, \Sigma_k^{(s)})}{\sum_{k'=1}^K \pi_{k'}^{(s)} \phi(\bar{\mathbf{y}}_i | \mu_{k'}^{(s)}, \Sigma_{k'}^{(s)})}.$$

- **M-step :** This step consists of maximizing the expected completed loglikelihood derived from the E-step. It leads to the following mixture parameter updates:

- The updated proportions are $\pi_k^{(s+1)} = \frac{1}{n} \sum_{i=1}^n t_{ik}^{(s)}$ for $k = 1, \dots, K$.
- Compute the updated means $\mu_1^{(s+1)}, \dots, \mu_K^{(s+1)}$ using the formulas (14) et (15) of Zhou *et al.* (2009): the j -th coordinate of $\mu_k^{(s+1)}$ is the solution of the following equations

$$\text{if } \left| \sum_{i=1}^n t_{ik}^{(s)} \left[\sum_{\substack{v=1 \\ v \neq j}}^p (\bar{y}_{ij} - \mu_{kv}^{(s)}) \Theta_{k,vj}^{(s)} + \bar{y}_{ij} \Theta_{k,jj}^{(s)} \right] \right| \leq \lambda \quad \text{then } \mu_{kj}^{(s+1)} = 0,$$

otherwise

$$\left[\sum_{i=1}^n t_{ik}^{(s)} \right] \mu_{kj}^{(s+1)} \Theta_{k,jj}^{(s)} + \lambda \text{sign}(\mu_{kj}^{(s+1)}) = \sum_{i=1}^n t_{ik}^{(s)} \sum_{v=1}^p \bar{y}_{iv} \Theta_{k,vj}^{(s)} - \left[\sum_{i=1}^n t_{ik}^{(s)} \right] \left[\left(\sum_{v=1}^p \mu_{kv}^{(s)} \Theta_{k,vj}^{(s)} \right) - \mu_{kj}^{(s)} \Theta_{k,jj}^{(s)} \right].$$

- For all $k = 1, \dots, K$, the covariance matrix $\Sigma_k^{(s+1)}$ is obtained via the precision matrix $\Theta_k^{(s+1)}$. The **glasso** algorithm available in the R package **glasso** of Friedman *et al.* (2011) is used to solve the following minimization problem on the set of symmetric positive definite matrices (denoted $\Theta \succ 0$):

$$\arg \min_{\Theta \succ 0} \left\{ -\ln \det(\Theta) + \text{trace} \left(S_k^{(s+1)} \Theta \right) + \rho_k^{(s+1)} \|\Theta\|_1 \right\},$$

RR n° ????

where $\rho_k^{(s+1)} = 2\rho \left(\sum_{i=1}^n t_{ik}^{(s)} \right)^{-1}$ and

$$S_k^{(s+1)} = \frac{\sum_{i=1}^n t_{ik}^{(s)} (\bar{\mathbf{y}}_i - \mu_k^{(s+1)}) (\bar{\mathbf{y}}_i - \mu_k^{(s+1)})'}{\sum_{i=1}^n t_{ik}^{(s)}}.$$

A.2 The classification case

The maximization of the regularized criterion (4) at μ_1, \dots, μ_K and $\Theta_1, \dots, \Theta_K$ is achieved using an algorithm similar to the one presented in Section A.1 when the labels z_i are known.

The j -th coordinate of the mean vector μ_k is the solution of the following equations

$$\text{if } \left\| \sum_{i=1}^n \mathbb{1}_{\{z_i=k\}} \left[\sum_{\substack{v=1 \\ v \neq j}}^p (\bar{y}_{ij} - \mu_{kv}) \Theta_{k,vj} + \bar{y}_{ij} \Theta_{k,jj} \right] \right\| \leq \lambda \quad \text{then } \mu_{kj} = 0,$$

otherwise

$$\left[\sum_{i=1}^n \mathbb{1}_{\{z_i=k\}} \right] \mu_{kj} \Theta_{k,jj} + \lambda \operatorname{sign}(\mu_{kj}) = \sum_{i=1}^n \mathbb{1}_{\{z_i=k\}} \sum_{v=1}^p \bar{y}_{iv} \Theta_{k,vj} - \left[\sum_{i=1}^n \mathbb{1}_{\{z_i=k\}} \right] \left[\left(\sum_{v=1}^p \mu_{kv} \Theta_{k,vj} \right) - \mu_{kj} \Theta_{k,jj} \right].$$

To estimate the sparse precision matrices $\Theta_1, \dots, \Theta_K$ from the dataset \mathbf{y} and the labels \mathbf{z} , we use the glasso algorithm to solve the following minimization problem on the set of symmetric positive definite matrices

$$\hat{\Theta}_k = \arg \min_{\Theta > 0} \left\{ -\ln \det(\Theta) + \operatorname{trace}(S_k \Theta) + \rho_k \|\Theta\|_1 \right\}, \quad (6)$$

for each $k = 1, \dots, K$. The ℓ_1 regularization parameter in (6) is given by $\rho_k = 2\rho \left(\sum_{i=1}^n \mathbb{1}_{\{z_i=k\}} \right)^{-1}$ and the empirical covariance matrix S_k is given by

$$S_k = \frac{\sum_{i=1}^n \mathbb{1}_{\{z_i=k\}} (\bar{\mathbf{y}}_i - \mu_k) (\bar{\mathbf{y}}_i - \mu_k)' }{\sum_{i=1}^n \mathbb{1}_{\{z_i=k\}}}.$$

And, alternate maximization in (μ_1, \dots, μ_K) and $(\Theta_1, \dots, \Theta_K)$ is achieved until convergence.

References

- Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**(3), 803–821.
- Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(7), 719–725.
- Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, **28**(5), 781–793.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society. Series B.*, **39**(1), 1–38.
- Friedman, J., Hastie, T., and Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**(3), 432–441.

- Friedman, J., Hastie, T., and Tibshirani, R. (2011). *glasso: Graphical lasso- estimation of Gaussian graphical models*.
- Law, M. H., Figueiredo, M. A. T., and Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(9), 1154–1166.
- Maugis, C., Celeux, G., and Martin-Magniette, M. (2009a). Variable selection for clustering with Gaussian mixture models. *Biometrics*, **65**(3), 701–709.
- Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009b). Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics and Data Analysis*, **53**, 3872–3882.
- Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2011). Variable selection in model-based discriminant analysis. *Journal of Multivariate Analysis*, **102**, 1374–1387.
- Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, **34**(3), 1436–1462.
- Meynet, C. and Maugis-Rabusseau, C. (2012). A sparse variable selection procedure in model-based clustering. Technical report.
- Murphy, T. B., Dean, N., and Raftery, A. E. (2010). Variable selection and updating in model-based discriminant analysis for high-dimensional data with food authenticity applications. *Annals of Applied Statistics*, **4**, 396–421.
- Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, **8**, 1145–1164.
- Raftery, A. E. and Dean, N. (2006). Variable Selection for Model-Based Clustering. *Journal of the American Statistical Association*, **101**(473), 168–178.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**(2), 461–464.
- Tadesse, M. G., Sha, N., and Vannucci, M. (2005). Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association*, **100**(470), 602–617.
- Wang, S. and Zhu, J. (2008). Variable Selection for Model-Based High-Dimensional Clustering and Its Application to Microarray Data. *Biometrics*, **64**(2), 440–448.
- Xie, B., Pan, W., and Shen, X. (2008). Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic journal of statistics*, **2**, 168–212.
- Zhou, H., Pan, W., and Shen, X. (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electronic Journal of Statistics*, **3**, 1473–1496.



**RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE**

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399