



# UNIVERSITÀ DI PARMA

## ARCHIVIO DELLA RICERCA

University of Parma Research Repository

Finding the Number of Normal Groups in Model-Based Clustering via Constrained Likelihoods

This is the peer reviewed version of the following article:

*Original*

Finding the Number of Normal Groups in Model-Based Clustering via Constrained Likelihoods / Cerioli, Andrea; García Escudero, Luis Angel; Mayo Iscar, Agustín; Riani, Marco. - In: JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS. - ISSN 1061-8600. - 27:2(2018), pp. 404-416. [10.1080/10618600.2017.1390469]

*Availability:*

This version is available at: 11381/2832548 since: 2021-11-09T15:20:22Z

*Publisher:*

American Statistical Association

*Published*

DOI:10.1080/10618600.2017.1390469

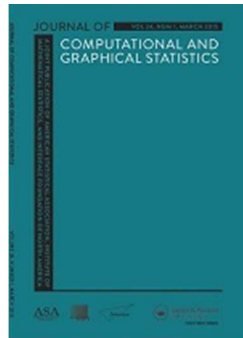
*Terms of use:*

Anyone can freely access the full text of works made available as "Open Access". Works made available

*Publisher copyright*

note finali coverpage

(Article begins on next page)



**Finding the Number of Normal Groups in Model-Based Clustering via Constrained Likelihoods**

Journal:	<i>Journal of Computational and Graphical Statistics</i>
Manuscript ID	JCGS-16-142.R3
Manuscript Type:	Original Article
Keywords:	Classification and Clustering, Exploratory Data Analysis, Number of groups
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
matlab_code.zip	

# Finding the Number of Normal Groups in Model-Based Clustering via Constrained Likelihoods

Andrea Cerioli

Dipart. di Scienze Economiche e Aziendali, Università di Parma

Luis Angel García-Escudero

Dpto. de Estadística e I.O. and IMUVA, Universidad de Valladolid

Agustín Mayo-Iscar

Dpto. de Estadística e I.O. and IMUVA, Universidad de Valladolid

and

Marco Riani

Dipart. di Scienze Economiche e Aziendali, Università di Parma

October 3, 2017

## Abstract

Deciding the number of clusters  $k$  is one of the most difficult problems in cluster analysis. For this purpose, complexity-penalized likelihood approaches have been introduced in model-based clustering, such as the well known BIC and ICL criteria. However, the classification/mixture likelihoods considered in these approaches are unbounded without any constraint on the cluster scatter matrices. Constraints also prevent traditional EM and CEM algorithms from being trapped in (spurious) local maxima. Controlling the maximal ratio between the eigenvalues of the scatter matrices to be smaller than a fixed constant  $c \geq 1$  is a sensible idea for setting such constraints. A new penalized likelihood criterion which takes into account the higher model complexity that a higher value of  $c$  entails, is proposed. Based on this criterion, a novel and fully automated procedure, leading to a small ranked list of optimal  $(k, c)$  couples is provided. A new plot called “car-bike” which provides a concise summary of the solutions is introduced. The performance of the procedure is assessed both in empirical examples and through a simulation study as a function of cluster overlap. Supplemental materials for the article are available online.

*Keywords:* Clustering, mixtures, EM algorithm, CEM algorithm, BIC, ICL.

# 1 Introduction

Cluster analysis is the art of clustering a data set into  $k$  groups of similar individuals. One of the main difficulties (and one of the most widely addressed problems) when using cluster analysis methods is how to decide the number of clusters  $k$  to be found. Sometimes  $k$  is known in advance because of the application in mind, but most of the times  $k$  is completely unknown and we want the data set itself to suggest a “sensible” number of groups. Several approaches for determining the number of clusters can be found in the literature (see, e.g., Milligan and Cooper, 1985; Rousseeuw, 1987; Tibshirani et al., 2001, among many others).

In this work we tackle the problem from a model-based perspective, with a normality assumption for the cluster components. Let  $x_1, \dots, x_n$  be the observations in  $\mathbb{R}^p$  to be clustered and let  $\phi(\cdot; \mu, \Sigma)$  denote the p.d.f. of the  $p$ -variate normal distribution with mean  $\mu$  and covariance matrix  $\Sigma$ . In model based clustering, there are two distinct approaches depending on whether the mixture or the classification likelihood function is used.

The first approach is based on maximization of the mixture log-likelihood (MIX)

$$L_k(\theta) = \sum_{i=1}^n \log \left[ \sum_{j=1}^k p_j \phi(x_i; m_j, S_j) \right],$$

where  $\theta = (p_1, \dots, p_k, m_1, \dots, m_k, S_1, \dots, S_k)$  is the set of parameters satisfying  $p_j \geq 0$  and  $\sum_{j=1}^k p_j = 1$ ,  $m_j \in \mathbb{R}^p$  and  $S_j$  a p.s.d. symmetric  $p \times p$  matrix. The optimal set of parameters based on this likelihood is

$$\hat{\theta}_{\text{Mixt},k} = \arg \max_{\theta} L_k(\theta). \quad (1)$$

Once  $\hat{\theta}_{\text{Mixt},k} = (\hat{p}_1, \dots, \hat{p}_k, \hat{m}_1, \dots, \hat{m}_k, \hat{S}_1, \dots, \hat{S}_k)$  is obtained, the observations in the sample are divided into  $k$  clusters by using posterior probabilities. That is, observation  $x_i$  is assigned to cluster  $j$  if  $j = \arg \max_l \hat{p}_l \phi(x_i; \hat{m}_l, \hat{S}_l)$ .

The second approach is based on maximization of the classification log-likelihood (CLA)

$$CL_k(\theta) = \sum_{i=1}^n \sum_{j=1}^k z_{ij}(\theta) \log (p_j \phi(x_i; m_j, S_j)),$$

where  $\theta = (p_1, \dots, p_k, m_1, \dots, m_k, S_1, \dots, S_j)$  and

$$z_{ij}(\theta) = \begin{cases} 1 & \text{if } j = \arg \max_l p_l \phi(x_i; m_l, S_l) \\ 0 & \text{otherwise .} \end{cases}$$

In this case, the optimal set of parameters is

$$\hat{\theta}_{\text{Clas},k} = \arg \max_{\theta} CL_k(\theta) \quad (2)$$

and observation  $x_i$  is now classified into cluster  $j$  if  $z_{ij}(\hat{\theta}_{\text{Clas},k}) = 1$ .

Based on the two different likelihood approaches (1) and (2), some proposals exist that lead to sensible ways for choosing the number of clusters. The basic idea is to maximize over  $k$  some complexity-penalized versions of these two likelihoods. Specifically, it is a common practice to add penalty terms depending on the number of free parameters in the model. Following this idea and taking the usual log-likelihood transformation, we envisage three possibilities:

$$\begin{aligned} \text{MIX-MIX} & : k_{\text{opt}} = \arg \min_k \left\{ -2L_k(\hat{\theta}_{\text{Mixt},k}) + v_k \right\} \\ \text{MIX-CLA} & : k_{\text{opt}} = \arg \min_k \left\{ -2CL_k(\hat{\theta}_{\text{Mixt},k}) + v_k \right\} \\ \text{CLA-CLA} & : k_{\text{opt}} = \arg \min_k \left\{ -2CL_k(\hat{\theta}_{\text{Clas},k}) + v_k \right\} \end{aligned}$$

where  $v_k$  is the penalty term counting the number of free parameters. This term is typically chosen as  $v_k = (kp + k - 1 + k(p + 1)p/2) \log(n)$ , if no particular constraints are posed on the scatter matrices  $S_1, \dots, S_k$ . In our notation, ‘‘MIX-MIX’’ corresponds to the use of the Bayesian Information Criterion (BIC) (see, e.g., Fraley and Raftery (2002); Fraley et al. (2017)), while ‘‘MIX-CLA’’ corresponds to the use of the Integrated Complete Likelihood (ICL) method proposed by Biernacki et al. (2000). The rationale behind the ICL criterion is that ‘‘mixture modeling’’ is a different problem from ‘‘clustering’’ and, thus, the number of groups obtained as a solution to these problems may not be the same. ‘‘CLA-CLA’’ is instead rooted in the crisp clustering framework of (2) and, to our knowledge, a novelty of this paper. The consideration of weights  $p_j$  in classification likelihoods, as in  $CL_k(\theta)$ , goes back to Symons (1981). Bryant (1991) mentioned the possible interest in classification likelihoods with weights to choose the number of groups in clustering, but without adding an extra penalty term for model complexity. It is important to note that the maximization of the classification and mixture likelihoods are both ill-posed problems because of their unboundedness without any constraint on the scatter matrices of the fitted normal components. To overcome this problem, we impose the fulfillment of a maximal ratio constraint

1  
2  
3 for all the eigenvalues of the scatter matrices. In addition, this eigenvalue ratio constraint  
4 is also useful to prevent traditional EM algorithms from being trapped in (non-interesting)  
5 spurious local maxima. This type of constraint depends on a fixed constant  $c$  in such a  
6 way that smaller  $c$  values favor cluster sphericity and homoscedasticity among groups. The  
7 use of penalized likelihood criteria under constraints, where the penalty term takes into  
8 account the higher model complexity entailed by a higher  $c$  value, suggest new criteria  
9 that will be denoted as  $MIX_c$ -MIX,  $MIX_c$ -CLA and  $CLA_c$ -CLA. Once the user is able to  
10 specify the type of cluster of interest through the specification of the constant  $c$ , these new  
11 criteria can be used to choose the number of clusters  $k$ . However, there are cases when that  
12 specification is not at all straightforward for the user. In those cases, as one of the main  
13 contributions of this work, we propose a fully automated procedure producing a small and  
14 ranked list of optimal  $(k, c)$ . All these “best ranked” solutions can be further examined, by  
15 applying validation tools in cluster analysis or taking into account the final user purposes,  
16 in order to choose the one that better fits the aims of analysis.  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27

28 The outline of our work is as follows. The need for constraints in model-based clustering  
29 is reviewed in Section 2 where the maximal eigenvalue ratio constraints are also presented.  
30 Section 3 shows how well-known criteria can be adapted in this constrained setting in such  
31 a way that a “sensible” number of clusters/components can be found when the constant  $c$   
32 is fixed in advance. Section 4 addresses the important problem of choosing simultaneously  
33 both  $k$  and  $c$ . Section 5 presents an automated procedure that returns a ranked small list of  
34 “optimal” cluster partitions and introduces a new plot which provides a concise summary  
35 of the best solutions. This procedure is illustrated in practice with both simulated and well-  
36 known real data sets. Section 7 describes a simulation study that shows the effectiveness of  
37 the proposed methodology under general settings. Finally, Section 8 concludes and provides  
38 some open lines for future research.  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49

## 50 2 Constrained clustering approaches

51  
52

53 The need for constraints on the scatter matrices arises because both (1) and (2) are un-  
54 bounded (just take  $\mu_1 = x_1$  and  $|\Sigma_1| \rightarrow 0$ ). Therefore, the associated maximization becomes  
55 a mathematically ill-posed problem (see, e.g., Day, 1969). Additionally, the lack of appro-  
56  
57  
58  
59  
60

1  
2  
3 appropriate constraints often leads the algorithms proposed for numerical maximization of (1)  
4 and (2) to be trapped in local maxima of the likelihood, associated to the detection of  
5 non-interesting “spurious solutions” (see, e.g., McLachlan and Peel, 2000).  
6  
7

8  
9 The lack of boundedness of (1) and (2) is often circumvented by resorting to “appropri-  
10 ate” initializations of the EM or CEM algorithms. Although this strategy is appealing, we  
11 note that, in this case, we would not be exactly trying to maximize the target functions in  
12 (1) and (2). In fact, it is known (see, e.g., Maitra (2009)) that the result of applying EM  
13 and CEM algorithms is strongly dependent on the chosen initialization, which may severely  
14 affect the value of the associated likelihoods and, consequently, the choice of  $k$  provided  
15 by MIX-MIX, MIX-CLA and CLA-CLA. For instance, we may have trouble with elon-  
16 gated parallel clusters when using the  $k$ -means method, or we can be affected by undesired  
17 “chaining effects” when considering single-linkage hierarchical clustering.  
18  
19

20  
21 Furthermore, it is important to note that cluster analysis is also not a well-defined  
22 problem from an applied viewpoint. There is nowadays a wide consensus about the fact  
23 that clustering techniques should always depend on the final data-analysis purpose, so that  
24 different goals require the use of different clustering approaches. Figure 1 in Hennig and  
25 Liao (2013) shows a toy example – to which we will return in Section 5.2.2 – with a data set  
26 obtained as a realization of a mixture of three well-separated bivariate normal components.  
27 Any clustering approach purely based on mixture modeling would determine the existence  
28 of three clusters. However, a “social stratification” framework, such as that exemplified  
29 in Hennig and Liao (2013), would clearly require the determination of more than three  
30 clusters. Similar conclusions could also hold in other important application fields, such as  
31 market research, where the construction of relevant clusters must often be coupled with  
32 subject matter aims. We thus argue that clustering should not be seen as a fully automatic  
33 task providing just one single solution and that the user always has to play an active role  
34 in it. The consideration of appropriate constraints on  $\theta$ , when maximizing (1) and (2), may  
35 allow the user to specify somehow the partitions of actual interest. This is another major  
36 reason that motivates our interest in introducing constraints in cluster analysis.  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52

53  
54 Some of the available solutions are based on imposing constraints on the elements of  
55 the decomposition of the scatter matrices. Different parameterizations of these scatter  
56  
57  
58  
59  
60

matrices allow us to obtain parsimonious variants of the unconstrained normal mixture model in such a way that (most of them) successfully serve to avoid spurious maximizers and convert the maximization of the likelihood into a well-defined problem. More details of these parsimonious parameterizations can be seen in Banfield and Raftery (1993) and Celeux and Govaert (1995). The resulting parameterizations can be easily addressed with the criteria described in Section 1 just by taking into account the number of free parameters. Another possibility, going back to Hathaway (1985), has been proposed and explored in Ingrassia and Rocci (2007) and García-Escudero et al. (2008, 2015). The approach is based on controlling the maximal ratio between the eigenvalues of the cluster scatter matrices. This implies maximizing the likelihoods (1) and (2), with  $\theta \in \Theta_c$  where  $\Theta_c = \{p_1, \dots, p_k \text{ with } \sum_{j=1}^k p_j = 1; m_1, \dots, m_k \text{ in } \mathbb{R}^p; S_1, \dots, S_k \text{ p.s.d. matrices with } \lambda_l(S_j) \leq c\lambda_q(S_h) \text{ for every } j, l, h, q\}$ . In the above,  $\{\lambda_l(S)\}_{l=1}^p$  stands for the set of eigenvalues for the scatter matrix  $S$ . Note that through the constant  $c \geq 1$  we are simultaneously controlling discrepancies from sphericity and differences among cluster scatters. The parameter  $c$  can be interpreted as the square root of the maximal ratio among the lengths of the equidensity ellipsoids defined by the  $\phi(\cdot; m_j, S_j)$  normal densities. Accordingly, we can define two constrained problems: the constrained mixture likelihood maximization (MIX<sub>c</sub>)

$$\hat{\theta}_{\text{Mixt},k}^c = \arg \max_{\theta \in \Theta_c} L_k(\theta), \quad (3)$$

and the constrained classification likelihood maximization (CLA<sub>c</sub>)

$$\hat{\theta}_{\text{Clas},k}^c = \arg \max_{\theta \in \Theta_c} CL_k(\theta). \quad (4)$$

The algorithms in Fritz et al. (2013) and in García-Escudero et al. (2014) can respectively be used to approximately solve these constrained maximizations. In these algorithms an eigenvalue truncation procedure is applied to enforce the eigenvalues ratio constraint in the EM and CEM steps. These algorithms always increase the likelihood functions throughout constrained EM and constrained CEM steps and so serve to find local maxima of the likelihood under the required constraints. Several random initializations are used when trying to find the global constrained maximum. Finally, an alternative approach can be found in Fraley and Raftery (2007) where semi-informative priors are placed on the eigenvalues of the covariance matrices. This results in a posterior mode, subsequently used in choosing the number of mixture components based on a BIC evaluated at that mode.

### 3 Penalized likelihoods in constrained clustering

We now define the  $MIX_c$ -MIX,  $MIX_c$ -CLA and  $CLA_c$ -CLA criteria for choosing the number of clusters when following the constrained maximization targets (3) and (4) for a fixed constant  $c \geq 1$ . This requires a modification of the “penalty term”, which should take into account the higher model complexity that a higher  $c$  value entails. We propose the use of a penalty term  $v_k^c$  defined as

$$v_k^c = \left( kp + k - 1 + \underbrace{k \frac{p(p-1)}{2}}_{\text{rotation par.}} + \underbrace{(kp-1) \left(1 - \frac{1}{c}\right) + 1}_{\text{eigenvalue par.}} \right) \log n. \quad (5)$$

Here we have distinguished the parameters related to orthogonal rotations of the scatter matrices – which are not affected by constraints – and those related to the eigenvalues. In the most constrained case ( $c = 1$ ), all the eigenvalues are equal, i.e. there is only one free extra parameter related to the eigenvalues. On the other hand, we recover  $kp(p+1)/2$  free parameters for the scatter matrices when we approximate the fully unconstrained case  $c \rightarrow \infty$ . A justification for this “soft” transition between the two extreme cases is as follows. If no constraints are posed on the whole set of eigenvalues of the scatter matrices, say  $\lambda_1, \dots, \lambda_D$  (with  $D = k \times p$ ), then we have the reference set  $A = \{(\lambda_1, \dots, \lambda_D) : 0 \leq \lambda_l\}$ . In the constrained case, we consider the set  $B = \{(\lambda_1, \dots, \lambda_D) : 0 \leq \lambda_l \leq c\lambda_q \text{ for every } l \neq q\}$ . A very simple idea is to consider the relative volume of set  $B$  with respect to  $A$  as a complexity measure. Of course, this ratio between volumes is not well-defined since neither  $A$  nor  $B$  are bounded sets. However, we can take into account that  $A = \bigcup_{t \geq 0} A_t$  and  $B = \bigcup_{t \geq 0} B_t$ , with  $A_t$  and  $B_t$  being sets defined as in the statement of Theorem 3.1.

**Theorem 3.1** *Let  $A_t = \{(\lambda_1, \dots, \lambda_D) : 0 \leq \lambda_l \leq t\}$  and  $B_t = \{(\lambda_1, \dots, \lambda_D) : 0 \leq \lambda_l \leq t; \lambda_l \leq c\lambda_q \text{ for every } l \neq q\}$ . Then*

$$\frac{Vol(B_t)}{Vol(A_t)} = \left(1 - \frac{1}{c}\right)^{D-1}.$$

The proof and a graphical illustration are left to the “online supplementary material”.

Theorem 3.1 is implicitly applied in our definition of the penalty term (5) by seeing that we have one “principal” eigenvalue and each of the remaining  $D - 1 = kp - 1$  eigenvalues

are “relatively” weighted by the multiplicative factor  $(1 - \frac{1}{c})$ . By considering the modified penalty term  $v_k^c$ , we have the following three new criteria for choosing the number of clusters depending on the maximal eigenvalue ratio  $c$ :

$$\begin{aligned}
 \text{MIX}_c\text{-MIX} : k_{\text{opt,MM}}(c) &= \arg \min_k \left\{ -2L_k(\hat{\theta}_{\text{Mixt},k}^c) + v_k^c \right\} \\
 &:= \arg \min_k F_{\text{MM}}(k, c) \\
 \text{MIX}_c\text{-CLA} : k_{\text{opt,MC}}(c) &= \arg \min_k \left\{ -2CL_k(\hat{\theta}_{\text{Mixt},k}^c) + v_k^c \right\} \\
 &:= \arg \min_k F_{\text{MC}}(k, c) \\
 \text{CLA}_c\text{-CLA} : k_{\text{opt,CC}}(c) &= \arg \min_k \left\{ -2CL_k(\hat{\theta}_{\text{Clas},k}^c) + v_k^c \right\} \\
 &:= \arg \min_k F_{\text{CC}}(k, c).
 \end{aligned}$$

Unlike the standard MIX-MIX, MIX-CLA and CLA-CLA criteria, here the use of our constrained proposals provides well-defined problems where the corresponding target functions  $F_m(k, c)$ ,  $m = \text{MM, MC or CC}$ , are bounded for a fixed  $1 \leq c < \infty$ . Moreover, spurious solutions are avoided provided that the supplied value  $c$  is not very large (García-Escudero et al., 2015). The specification of  $c$  may be seen as a sensible way for the user to declare the maximum allowed difference on cluster scatters that he/she is willing to admit. This choice then depends on the final clustering purpose in mind. For instance, the social stratification problem in Hennig and Liao (2013) would require the user to specify a value of  $c$  close to 1. This implies the search of almost spherical clusters with similar scatters or, analogously, the use of the Euclidean distance for clustering. Other problems would require larger values of  $c$ , and so the detection of less restricted clusters. Once the value of  $c$  has been fixed, the determination of  $k_{\text{opt},m}(c)$  is done by minimizing the previously introduced criteria with respect to  $k$  for a given method  $m$  where  $m = \text{MM, MC or CC}$ . It should also be noted that this approach is not affine equivariant due to the lack of equivariance of the chosen constraints. Therefore, standardizing the variables may be needed if, for instance, very different scales are involved.

To illustrate how the methodology can be applied, let us consider a simulated data set of size  $n = 100$  and dimension  $p = 2$  from a  $k = 3$  component mixture obtained by applying the `MixSim` method of Maitra and Melnykov (2010), as extended by Riani et al. (2015) and incorporated into the `FSDA` toolbox of Matlab (Riani et al., 2012). The

data set has been generated by imposing an average cluster overlap (defined as a sum of pairwise misclassification probabilities) equal to 0.04 and a maximum eigenvalue ratio for the scatters matrices equal to 5. Given two clusters  $j$  and  $l$  ( $j \neq l = 1, \dots, k$ ), indexed by  $\phi(x; m_j, S_j)$  and  $\phi(x; m_l, S_l)$ , with probabilities of occurrence  $p_j$  and  $p_l$ , the *overlap* between groups  $j$  and  $l$  is defined as sum of the two misclassification probabilities  $w_{jl} = w_{j|l} + w_{l|j}$  where  $w_{j|l} = Pr[p_l \phi(x; m_l, S_l) < p_j \phi(x; m_j, S_j)]$ . The average overlap is the sum of the off-diagonal elements of the matrix of the misclassification probabilities  $w_{j|l}$  divided by  $k(k-1)/2$ . Figure 1 shows two scatter plots of this simulated data set, without and with the “true” assignments labels. It is not perfectly clear by visual inspection, at least looking at the graph in the left panel, whether there are two or three clusters.

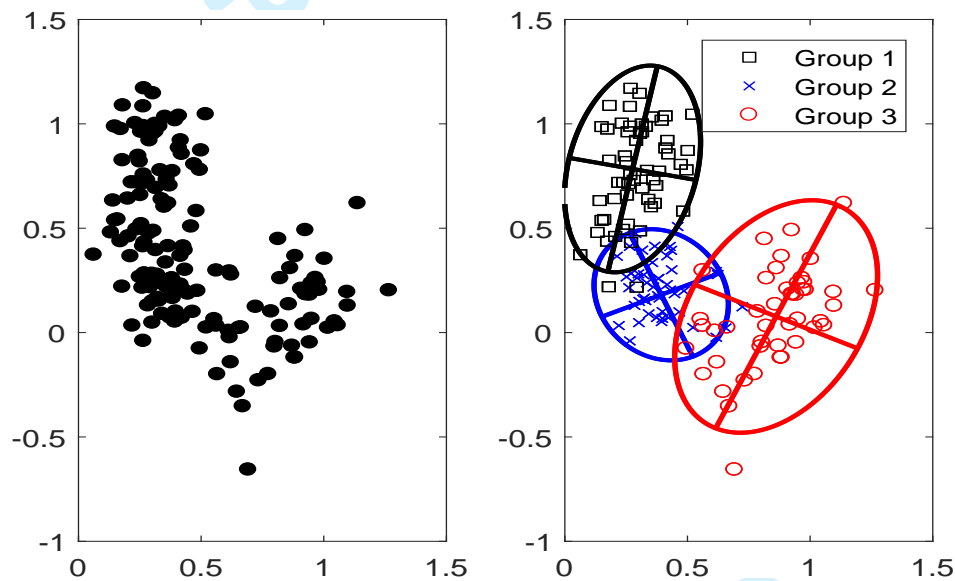


Figure 1: Simulated bivariate data set. The panel on the right shows the data set with the “true” labels and tolerance ellipsoids summarizing the three normal components.

Figure 2 shows the curves of our objective function that are obtained by monitoring  $F_{CC}(k, c)$  when  $c$  ranges over the interval  $[1, 128]$  and  $k$  ( $x$  axis) goes from 1 to 5. The large left panel shows all the 8 trajectories of  $F_{CC}(k, c)$  that are obtained by considering  $c = \{2^0, 2^1, 2^2, \dots, 2^7\}$ . The value of  $c$  for the lowest curve at each  $k$  is shown in the table below the caption of the Figure. For instance, when  $k = 2$  the lowest value is for  $c = 16$ ; for  $k = 3$  the lowest value is for  $c = 8$ ; etc.. Given that the eight trajectories strongly overlap, in the first five right panels of this figure we show what happens for the five smallest values

of  $c$  we have considered ( $c = 1, 2, 4, 8, 16$ ). The trajectories for the 3 largest values of  $c$  are very similar and, thus, they are all reported in the same final right panel.

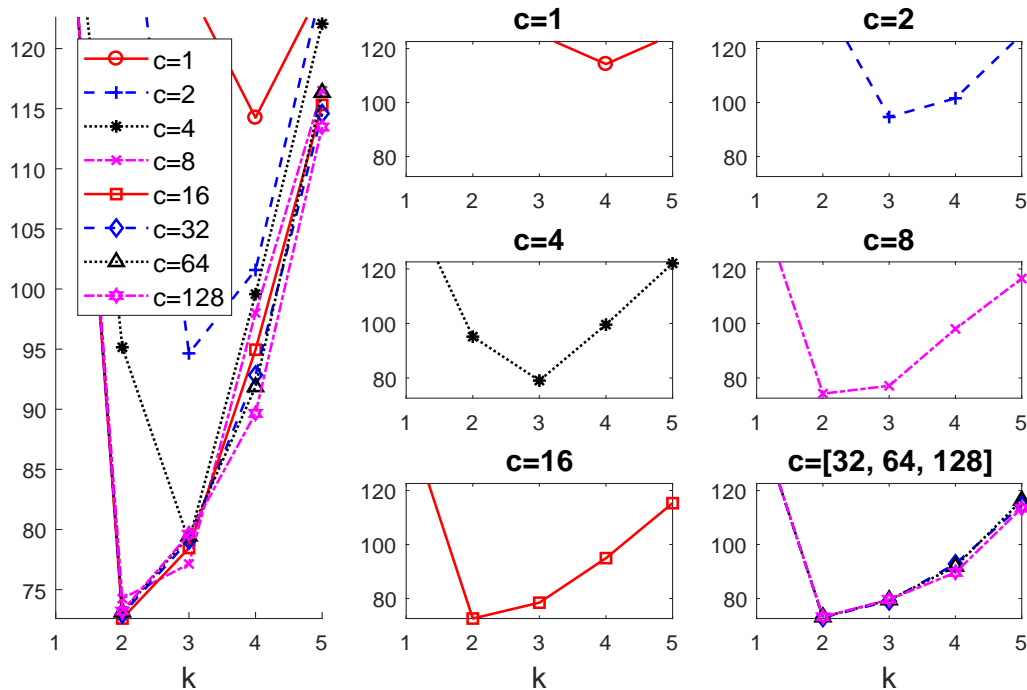


Figure 2: Analysis of the modified constrained criteria when using the  $CLA_c$ -CLA approach for the data set shown in Figure 1.

By using the curves plotted in Figure 2, we can see that the optimal values for the number of clusters are, for instance,  $k_{\text{opt},CC}(2) = 3$  (i.e., when  $c = 2$ ) or  $k_{\text{opt},CC}(16) = 2$  (i.e., when  $c = 16$ ). We thus obtain  $k = 3$ , which corresponds to the true number of components, when we are interested in neither very spherical nor homoscedastic clusters, but we find  $k = 2$  clusters when we allow for more elongated group structures. The latter also provides a sensible cluster partition, from a clustering point of view, since Group 1 and Group 2 (right panel of Figure 1) may be seen as being part of the same cluster if only the left panel of Figure 1 is visualized. Similar plots are given in Figure 3 when  $F_{MM}(k, c)$  is monitored. We can see that the use of an objective function more focused on “mixture modeling”, such as  $MIX_c$ -MIX, always suggests  $k_{\text{opt},MM}(c) = 3$  (i.e., the true number of mixture components) for every value of  $c > 1$  tried. A higher number of groups is only needed in the case  $c = 1$ , due to the strong assumption of homoscedasticity.

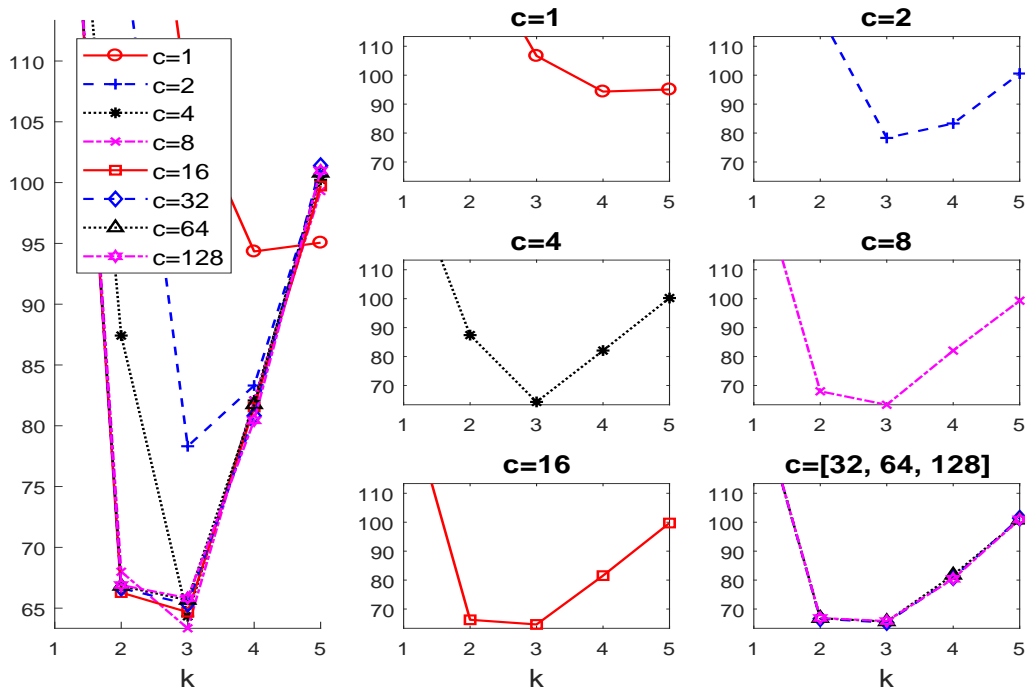


Figure 3: As Figure 2, but now for  $MIX_c$ -MIX.

## 4 Simultaneous choice of $k$ and $c$

Alternatively, we may know the number of groups  $k$  due to some economic, physical or operational reason, when our aim is that of obtaining a sensible value for  $c$ . Notice that in this case the user does not want to impose any particular structure on the clusters to be detected. This goal can be achieved by using the same penalized criteria as before, but now minimizing over  $c$ . Therefore, if  $k$  is assumed to be known, we take

$$c_{\text{opt},m}(k) = \arg \min_c F_m(k, c), \text{ for } m = \text{MM, MC and CC},$$

as our choice for the optimal value of  $c$ . This information is included in the “online supplementary material” for the data set shown in Figure 1.

In practice the most interesting case is when both the proper number of clusters  $k$  and the constraining factor  $c$  are unknown. We have argued before that a fully unsupervised choice of both parameters, depending only on the data set at hand, is very likely to be out of reach for most applications. Nevertheless, it would be helpful if we were able to reduce the space of all possible choices of the  $(k, c)$  parameter pairs to a small list of “sensible” ones,

1  
2  
3 in order to find more easily the pair that best fits the user’s main purpose in clustering.

4  
5 One might think that direct study of the functionals  $(k, c) \mapsto F_m(k, c)$ , for  $m =$   
6 MM, MC and CC, would provide valuable information about how to choose simultane-  
7 ously  $k$  and  $c$ . Contour plots that summarize the resulting monitoring process for the data  
8 set shown in Figure 1 are in the “online supplementary material”. Our experience is that  
9 these contour plots are not easily interpreted. Additionally, there are partitions obtained  
10 with different  $(k, c)$  parameters that correspond to essentially the same substantial groups,  
11 or that simply differ because of the inclusion of extra (non-interesting) spurious clusters.  
12  
13  
14  
15  
16  
17  
18  
19

## 20 **5 Automated selection of a list of “sensible” solutions**

### 21 **5.1 Methodology**

22  
23 In this section, we offer a fully automated procedure that leads to a small and ranked list  
24 of “optimal” choices for the pair  $(k, c)$ . The proposed methodology, based on our three  
25 constrained clustering criteria, relies on analysis of the stability of the cluster partitions  
26 through the Adjusted Rand Index (ARI). The ARI is a measure of the similarity between  
27 two data clusterings after “adjusting for chance”. The adjusted Rand index is thus ensured  
28 to have a value close to 0 for random labeling independently of the number of clusters and  
29 samples and exactly 1 when the clusterings are identical (up to a permutation). Specifically,  
30 the procedure first detects a list with  $L$  “plausible” partitions. Such “plausible” partitions  
31 may include some partitions that are essentially the same as others already detected, be-  
32 cause spurious clusters made up with few almost collinear or very concentrated data points  
33 are found. In a second step, the partitions including spurious clusters are discarded and  
34 we end up with a (typically very) reduced and ranked list with  $T$  “optimal” partitions.  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46

47 Given a pair  $(k, c)$ , let  $\mathcal{P}(k, c)$  denote the partition into  $k$  subsets which is obtained by  
48 solving the problem (3) or (4), with the given  $k$  and  $c$  and one of the suggested methods  
49  $m =$  MM, MC and CC. Let  $\text{ARI}(\mathcal{A}, \mathcal{B})$  denote the ARI between partitions  $\mathcal{A}$  and  $\mathcal{B}$ . We  
50 consider that two partitions  $\mathcal{A}$  and  $\mathcal{B}$  are “essentially the same” when  $\text{ARI}(\mathcal{A}, \mathcal{B}) \geq \varepsilon$ , for  
51 a fixed threshold  $\varepsilon$ . Clearly, the higher the value of the threshold the greater is the number  
52 of tentative different solutions which are considered.  
53  
54  
55  
56  
57  
58  
59  
60

Let us consider the sequence  $k = 1, \dots, K$ , where  $K$  is the maximal number of clusters, and a sequence  $c = c_1, \dots, c_C$  of  $C$  possible constraint values. For instance, the sequence of powers of 2,  $c_1 = 2^0, c_2 = 2^1, \dots, c_C = 2^{C-1}$  is recommended because it enables us to consider a sharp grid of values close to 1. By using this notation, the proposed automated procedure may be described as follows:

1. Obtain the list of “plausible” solutions:

1.1 *Initialize*: Start with  $K \times C$  possible  $(k, c)$  pairs to be explored. Let  $\mathcal{E}_0 = \{(k, c) : k = 1, \dots, K \text{ and } c = c_1, \dots, c_C\}$ .

1.2 *Iterate*: If  $\mathcal{E}_{l-1}$  is the set of pairs  $(k, c)$  not already explored at stage  $l-1$ , then:

1.2.1 Obtain  $(k_*^l, c_*^l) = \arg \min_{(k,c) \in \mathcal{E}_{l-1}} F_m(k, c)$ .

1.2.2 Remove all of the cluster partitions  $(k, c) \in \mathcal{E}_{l-1}$  with  $k = k_*^l$  and values of  $c$  which are adjacent to  $c_*^l$ , and such that they are very “similar” to partition  $\mathcal{P}(k_*^l, c_*^l)$  for the given threshold value  $\varepsilon$ , in the sense that

$$\text{ARI}(\mathcal{P}(k, c), \mathcal{P}(k_*^l, c_*^l)) \geq \varepsilon. \quad (6)$$

Take  $\mathcal{E}_l$  as the set  $\mathcal{E}_{l-1}$  after removing the pairs yielding “similar” partitions.

1.3 *Finalize*: The iterative procedure ends when  $\mathcal{E}_L = \emptyset$  (or when  $L$  is a positive prefixed integer number) and it returns  $\{(k_*^1, c_*^1), (k_*^2, c_*^2), \dots, (k_*^L, c_*^L)\}$  as a list with  $L$  “feasible” parameters combinations.

2. Obtain the list of “optimal” solutions:

2.1 *Initialize*: Start from  $\mathcal{I}_0 = \{1, \dots, L\}$  and the  $L \times L$  matrix  $(d_{r,s})_{r,s=1,\dots,L}$ , where

$$d_{r,s} = \text{ARI}(\mathcal{P}(k_*^r, c_*^r), \mathcal{P}(k_*^s, c_*^s)),$$

2.2 *Iterate*: Given  $\mathcal{I}_{t-1}$  the non discarded “plausible” solutions at stage  $t-1$ :

2.2.1 Take  $(k_{\text{opt}}^t, c_{\text{opt}}^t) = (k_*^{l_t}, c_*^{l_t})$  where  $l_t$  is the  $t$ -th element of  $\mathcal{I}_{t-1}$  (where the indexes in  $\mathcal{I}_{t-1}$  are sorted from lowest to highest).

2.2.2 Discard “spurious” solutions (i.e., those that are similar to the already detected “optimal” ones):  $\mathcal{I}_t = \mathcal{I}_{t-1} \setminus \{r : r \in \mathcal{I}_{t-1}, r > l_t \text{ and } d_{r,l_t} \geq \varepsilon\}$ .

2.3 *Finalize*: The iterative procedure ends when  $\mathcal{I}_T = \emptyset$ . It returns

$$\{(k_{\text{opt}}^1, c_{\text{opt}}^1), (k_{\text{opt}}^2, c_{\text{opt}}^2), \dots, (k_{\text{opt}}^T, c_{\text{opt}}^T)\}$$

as the “optimal” pairs.

To simplify notation, we have deleted the subscript  $m$  for the criterion used (i.e.,  $(k_{\text{opt}}^t, c_{\text{opt}}^t)$  should be  $(k_{\text{opt},m}^t, c_{\text{opt},m}^t)$  for  $m = \text{MM}, \text{MC}$  and  $\text{CC}$ ). Additionally, the complete automated procedure is hereinafter referred to *autMIXMIX*, *autMIXCLA* and *autCLACLA*.

For each “optimal” pair  $(k_{\text{opt}}^t, c_{\text{opt}}^t)$ , it is also informative to take into account the so-called “best interval”  $\mathcal{B}_t$  defined as

$$\mathcal{B}_t = \{c : F_m(k_{\text{opt}}^t, c_{\text{opt}}^t) \leq F_m(k_{\text{opt}}^t, c)\}, \quad (7)$$

and the so-called “stable interval” defined as

$$\mathcal{S}_t = \{c : \text{ARI}(\mathcal{P}(k_{\text{opt}}^t, c), \mathcal{P}(k_{\text{opt}}^t, c_{\text{opt}}^t)) \geq \varepsilon\}. \quad (8)$$

A large interval  $\mathcal{B}_t$  means that the number of clusters  $k_{\text{opt}}^t$  is “optimal”, in the sense of (7), for a wide range of  $c$  values. A large interval  $\mathcal{S}_t$  means that the solution is “stable”, in the sense of (8), because it does not essentially change when moving  $c$  in that interval.

## 5.2 Examples

Three examples have been included in the text to exemplify the performance of the proposed automated procedure. The first has to do with its application to the previously simulated data set and the second one considers a “toy example” which serves to illustrate how different cluster partitions may be needed depending on the final clustering purposes. The third example is based on a more complex “oil data set” with  $p = 8$  variables and a possible larger number of clusters  $k$ . The “online supplementary material” includes an additional example when the methodology is applied for the classical and well-known “Iris data set”.

### 5.2.1 Application to simulated data

We have applied the proposed automated procedure with an ARI to the simulated data set displayed in Figure 1. We start with the classification approach and the needed tables to

reproduce the analysis displayed in this section can be seen in the “online supplementary material”. These tables contains the values of  $CLA_c$ -CLA for all  $(k, c)$  pairs and we find the threshold given in equation (6) considering the matrix which contains the ARI indexes for two consecutive values of  $c$  given  $k$ . The first table there shows that (as anticipated in the previous section) the best value of  $k$  is 2 and the second best value of  $k$  is 3. The second table there, on the other hand, shows that for  $k = 2$  (overall best solution) there is essentially just one solution in the range  $c [2,128]$  (the values of ARI in this interval are all greater than 0.8). For  $k = 3$  (second overall best solution) the solution is stable in the interval  $c [1,128]$  with many solutions giving exactly the same partition. The situation seems to be more complex for  $k = 4$  where in the interval  $c [16,128]$  we virtually obtain the same solution. The ARI index between  $c = 8$  and  $c = 16$  when  $k = 4$  is 0.71 and suggests a moderate change in the classification, while the one between  $c = 4$  and  $c = 8$  goes down to 0.57. Finally, the solutions in the interval  $c [1,4]$  seem homogeneous. Our procedure combines in a fully automatic way the information in the two previous tables. The analysis of the the values of  $CLA_c$ -CLA for all  $(k, c)$  pairs shows that the first best solution is for  $k = 2$  and  $c = 16$  and that this solution remains the best in the interval  $c [8,128]$  and (if we consider a threshold of ARI equal to 0.8) we find that this solution is stable in the interval  $c [2,128]$ . In order to find the second solution, we remove from the table with all the values of  $CLA_c$ -CLA those  $(k, c)$  pairs in which the first solution was stable and check for the new minimum. The joint examination of the two tables, shows that the second solution takes place when  $k = 3$ , and  $c = 8$ , and that this solution is best and stable in the interval  $c [4,128]$  and that is stable in the interval  $c [1,128]$ . Given that the ARI index between the second solution ( $k = 3$  and  $c = 8$ ) and the first one ( $k = 2$  and  $c = 16$ ) is 0.38 (below the prefixed threshold of 0.8 we consider the second solution as non spurious). The procedure goes on until we find the the first (say) three or four non spurious solutions.

The corresponding four best-ranked solutions are shown in Figure 4. We see that we recover the true number of clusters  $k_{opt,CC}^2 = 3$  in the second solution. The solution with  $k_{opt,CC}^1 = 2$  makes perfect sense from the “pure” clustering point of view adopted by the  $CLA_c$ -CLA criterion and, thus, it is the first offered partition. The homoscedastic  $c = 1$  solution is shown as the ninth one and it proposes  $k_{opt,CC}^9 = 5$  clusters.

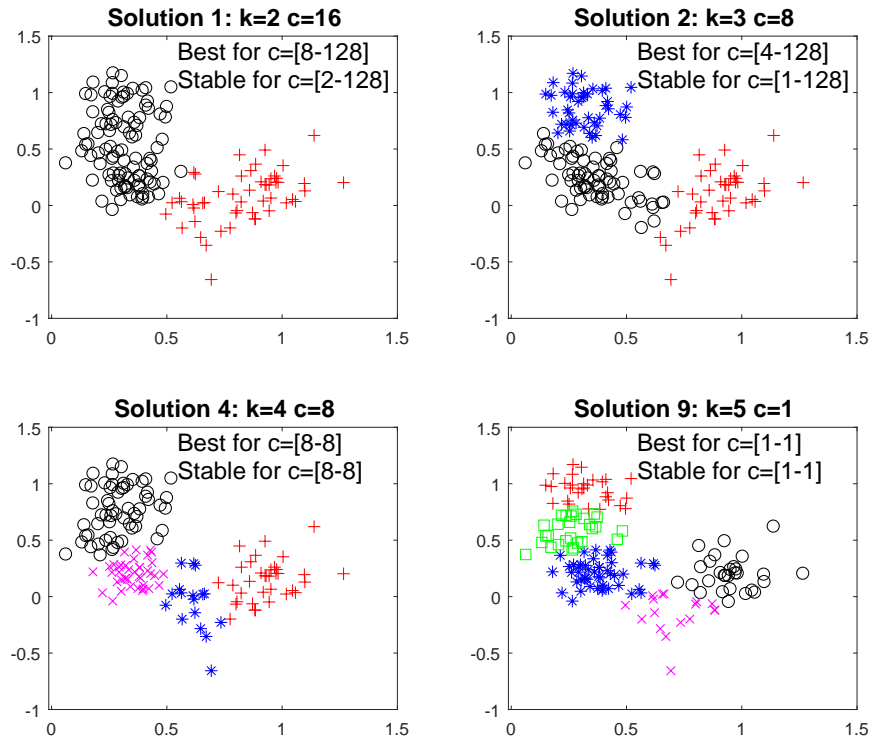


Figure 4: The  $T = 4$  best-ranked partitions when using the *autCLACLA* procedure for the simulated data set displayed in Figure 1.

In order to provide a concise summary of the non spurious solutions found and in order to better distinguish the most relevant solutions from those which are local, we propose a new graphical display which we call “car-bike”. This plot shows on the horizontal axis the value of  $c$  and on the vertical axis the value of  $k$ . For each solution we draw a rectangle for the interval of values for which the solution is best and stable and a horizontal line which departs from the rectangle for the values of  $c$  in which the solution is only stable. Finally, for the best value of  $c$  associated to the solution, we show a circle with two numbers, the first number indicates the ranked solution among those which are not spurious and the second one the ranked number including the spurious solutions. This plot has been baptized “car-bike”, because the first best ranked solutions (in general 2 or 3) are generally best and stable for a large number of values of  $c$  and therefore will have large rectangles. In addition, these solutions are likely to be stable for additional values of  $c$  and therefore are likely to have horizontal lines departing from the rectangles (from here the name “cars”).

Finally, local minor solutions (which are associated with particular values of  $c$  and  $k$ ) do not generally present rectangles or lines and are shown with circles (from here the name “bikes”). Figure 5 shows the car-bike plot coming from the *autCLACLA* procedure. This plot shows that there are two main solutions (“cars”) one with 2 groups and the second with 3 groups. In this case, the two cars are station wagons because the horizontal line just departs from the left side of the rectangles. The other two non-spurious solutions (“bikes”) are for  $k = 4$  and  $c = 8$ , and  $k = 5$  and  $c = 1$ . An additional bonus of this new graphical representation is that it allows us to immediately spot the area where the best values of  $c$  lie. If we examine the car-bike plot from a vertical perspective we can see that the optimal values of  $c$  are concentrated in the zone between  $c = 8$  and  $c = 16$ . In the car-bike plot, the height of the rectangles can be made proportional to the order of the ranked solution. Using this criterion the height of the rectangle of say solution 3 out of 8 is  $(8 - 3)/8$ -th that of the first. Careful examination of Figure 5 reveals that the height of the rectangle for the second best ranked solution is slightly smaller than that of the first one.

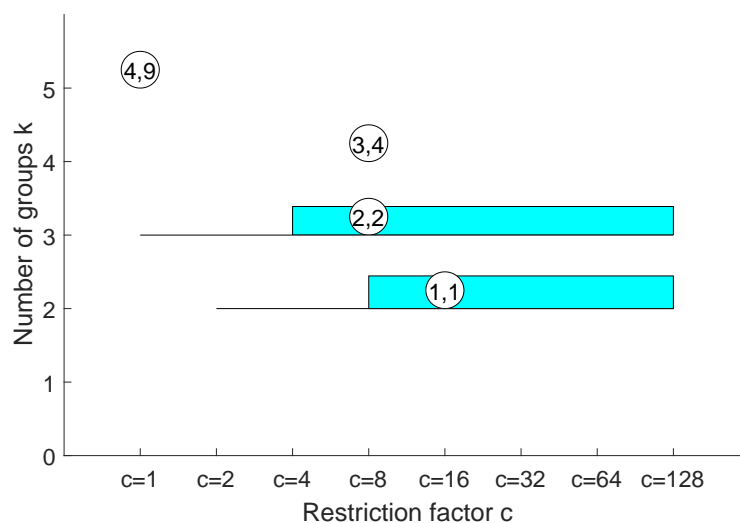


Figure 5: The “car-bike” plot when using using the *autCLACLA* procedure for the simulated data set displayed in Figure 1.

A figure showing the first 4 discarded “spurious” solutions for the simulated data set can be found in the “online supplementary material”. We can see there that these discarded solutions either include clusters made up with a few almost collinear or concentrated ob-

1  
2  
3 observations or correspond to solutions close to one already detected “optimal” partition.

4  
5 A referee suggested us to comment on the choice of the threshold of the ARI index. Our  
6  
7 experience is that the value of the threshold is important just to highlight/hide the local  
8  
9 solutions “bikes”, but plays no rule in the detection of the main solutions (“cars”). In the  
10  
11 example above, we have used a threshold of the Rand index equal to 0.8. If we consider  
12  
13 a threshold of 0.7 the only modification concerns the third non spurious solution (bike  
14  
15 associated with  $k = 4$  and  $c = 8$ ) which disappears because the third solution ( $c = 128$  and  
16  
17  $k = 4$  which is spurious and therefore is not pictured in the plot) with the new threshold is  
18  
19 considered stable in the interval  $c [8,128]$  (as seen in the “online supplementary material”).

20  
21 Figure 6 shows the ranked set of “optimal” solutions when using the *autMIXMIX*  
22  
23 procedure. Notice that, from a mixture modeling point of view, we obtain the correct  
24  
25 number of components ( $k_{\text{opt,MM}}^1 = 3$ ) in the first position. This result agrees with the  
26  
27 well known fact that mixture modeling is better suited to address cluster overlap than  
28  
29 “pure” clustering, which instead ideally assumes well-separated clusters. The car-bike plot  
30  
31 (not given here for lack of space) for *autMIXMIX* shows two big cars and and some bikes  
32  
33 scattered around showing once again the presence of just two relevant solutions.

### 34 **5.2.2 Application to Hennig and Liao’s type of data**

35  
36 Section 5 in Hennig and Liao (2013) includes a toy example to illustrate that there are  
37  
38 cases “where a mixture model is true and most people may have a natural intuition about  
39  
40 the true clusters” but these clusters “are not necessarily the clusters that a researcher is  
41  
42 interested in”. In the spirit of that toy example, we consider the simulated data set shown  
43  
44 in Figure 7. This data set corresponds to a realization of a mixture of three well-separated  
45  
46 bivariate normal components. Without knowledge of the underlying substantive problem,  
47  
48 one would then agree that  $k = 3$  is a sensible choice for  $k$ . However, let us assume (as  
49  
50 Hennig and Liao did) that we are facing a social stratification clustering problem and that  
51  
52 the two variables are, for instance, an income and a status indicator. By choosing  $k = 3$  and  
53  
54 very unrestricted scatter matrices, one cluster would contain both the poorest people with  
55  
56 lowest status and the richest people with the highest status. Therefore, in this particular  
57  
58 application, a higher number of (more homoscedastic) clusters is surely needed.  
59  
60

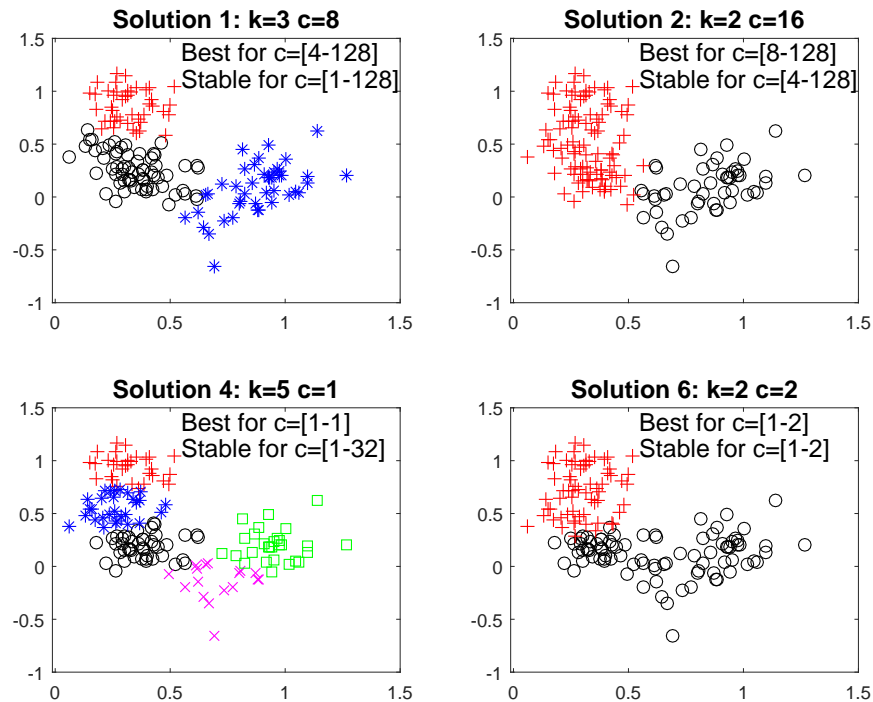


Figure 6: The  $T = 4$  “optimal” partitions when using the *autMIXMIX* procedure for the data set displayed in Figure 1.

Figure 7 shows  $T = 4$  “optimal” solutions when using the *autMIXMIX* procedure. We can see that the best-ranked partition is exactly the one which discovers the 3 bivariate normal components. The second and third best ranked partitions offer the user a more sensible clustering partition for that particular “social stratification” problem. The fourth best ranked solution offers a very peculiar partition where the two more concentrated normal components are surprisingly joined together. However, this more “exotic” solution just appears after the more “sensible” ones. In any case, we think that it is useful to reduce all the possible pairs  $(k, c)$  to such a type of small lists of best-ranked partitions, where the user can hopefully choose the one that better fits his/her clustering purposes. The car-bike plot for this data set is presented in the “online supplementary material”.

### 5.2.3 Olive oil data set

In the previous examples we have seen how our methodology works in the presence of a small number of clusters. The purpose of this section is to test our proposal on a more real

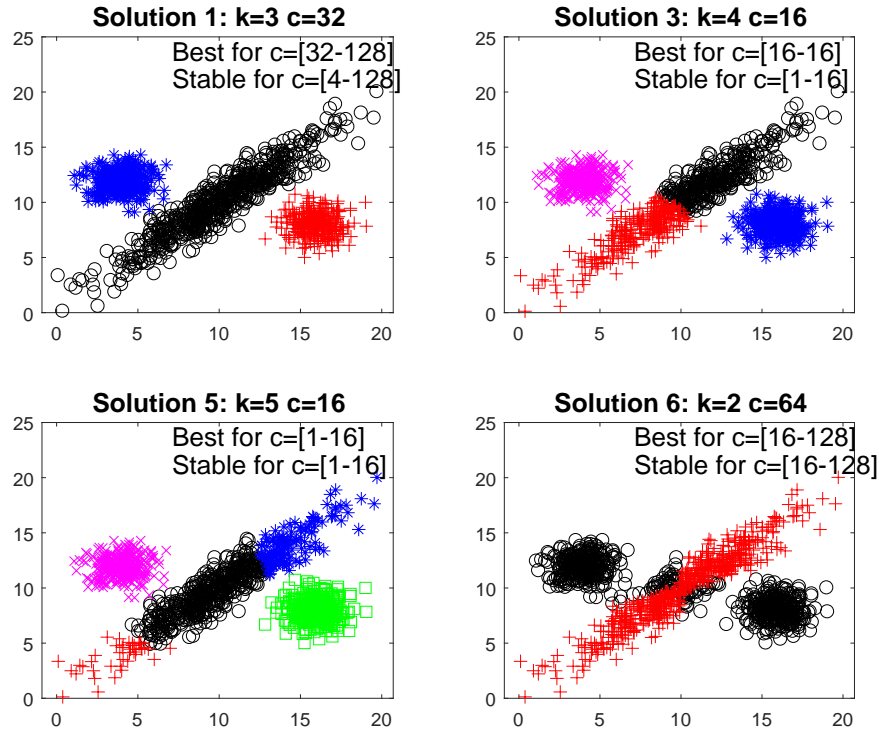


Figure 7: The  $T = 4$  best-ranked partitions when using the *autMIXMIX* for a data set similar to that in Hennig and Liao (2013).

and complex dataset which contains several ‘possibly’ overlapping groups.

The “olive oil” data set (Forina et al., 1983) contains  $p = 8$  chemical measurements on the acid components of  $n = 572$  olive oil specimen produced in various regions in Italy: (1) North Apulia, (2) Calabria, (3) South Apulia, (4) Sicily, (5) Inland Sardinia, (6) Coastal Sardinia, (7) East Liguria, (8) West Liguria, and (9) Umbria. This data set is available, for instance, from the *pgmm* package at CRAN (McNicholas et al., 2015). We standardize these variables and apply the *autMIXMIX* criterion with a maximal number of clusters  $K = 12$  and tentative constraining factors  $c = 1, 2, 4, \dots, 128$ . The three best ranked solutions obtained are  $(c = 128, k = 6)$ ,  $(c = 128, k = 7)$  and  $(c = 128, k = 5)$ . The use of the *mclust* package (Fraley and Raftery, 2002; Fraley et al., 2017) returns the following three best ranked models:  $(VVV, k = 5)$ ,  $(VVV, k = 9)$  and  $(VVV, k = 8)$  which are obtained by using the BIC criterion. “VVV” stands for ellipsoidal and varying volume, shape, and orientations. This is in concordance with the choice  $c = 128$ , i.e. scatter matrices as unconstrained as possible. Table 1 shows the ARI values obtained for the three

Table 1: ARI values of the 3 best ranked solutions with respect to the true regions in the “Olive oil” dataset. The proposed models appear within parenthesis.

	ARI values		
	First	Second	Third
<i>autMIXMIX</i>	0.8060 ( $c = 128, k = 6$ )	0.8468 ( $c = 128, k = 7$ )	0.7441 ( $c = 128, k = 5$ )
<i>mclust</i>	0.7763 (VVV, $k = 5$ )	0.6258 (VVV, $k = 9$ )	0.6601 (VVV, $k = 8$ )

best ranked solutions with respect to the true classification in the 9 regions.

In this case the three best partitions returned by *autMIXCLA* and *autCLACLA* are essentially the same (and in the same order) in this case.

Table 2 shows the confusion matrix for the second best solution (that with ARI value 0.8468). We can see the original oil production regions are quite well recovered in this suggested cluster partition. The only regions that we are not able to disentangle are regions 2 and 4 (Calabria and Sicily) and regions 5 and 6 (inland and costal Sardinia). A thorough examination of the scatter plot matrix reveals that oils from regions 2 and 4 take values on these fatty acid contents that are almost virtually impossible to distinguish. Regions 5 and 6, apart from belonging to same island, can only be slightly discriminated in just two out of the eight variables. Additionally, regions 5 and 6 constitute a rather compact and well-differentiated cluster with respect the other regions when joined together. Although the use of the traditional BIC criterion within *mclust* sometimes suggests  $k = 9$  clusters, the partition proposed with this number of clusters is far from the true one, mainly because the largest “South Apulia” cluster is split into two clusters.

## 6 Example: Road traffic data

In this example, we illustrate the proposed methodology in a road traffic problem. The data set comes from speed measurements collected in an expressway in Paris. To be more precise, data are from a fixed station that catches the average speed of cars passing through it resulting in 180 daily measurements from 5:00 a.m. to 23:00 p.m. This data set was used in García-Escudero and Gordaliza (2005) to illustrate the performance of a robust functional

Table 2: Confusion matrix for the second best partition obtained when using the *autMIXMIX* method.

Regions \ Clusters	1	2	3	4	5	6	7
1. North Apulia	0	24	0	0	1	0	0
2. Calabria	0	0	0	0	55	0	1
3. South Apulia	0	0	0	0	7	0	199
4. Sicily	0	18	0	0	16	0	2
5. Inland Sardinia	0	0	0	65	0	0	0
6. Coastal Sardinia	0	0	0	33	0	0	0
7. East Liguria	12	0	38	0	0	0	0
8. West Liguria	50	0	0	0	0	0	0
9. Umbria	0	0	3	0	0	48	0

clustering methodology. The highly rough speed curves were smoothed by projection onto a basis of cubic B-splines resulting from using 6 equispaced knots. Thus, each of these curves is converted into a  $p = 10$  dimensional vector. Although we start with data corresponding to 617 consecutive days, we only consider 493 curves surviving the trimming approach described in García-Escudero and Gordaliza (2005). Anomalous traffic days with slow traffic during very discontinuous and large time-periods, or days when the speed detector seems to provide wrong measurements, are discarded by using this trimming approach.

When applying the proposed methodology to this  $n \times p = 493 \times 10$  data set with a maximal number of clusters  $K = 10$  and constraining factors  $c = 1, 2, 4, \dots, 128$ , the three best ranked partitions are obtained for  $k = 3$ ,  $k = 4$  and  $k = 2$  (in this order) and  $c = 128$ . The proposed clustering partition is exactly the same regardless of whether the *autMIXMIX* or *autCLACLA* approach are used. The three best ranked solutions are presented in the “online supplementary material” by using the functional boxplots of Sun and Genton (2011). The first proposed solution detects 3 clusters: the first cluster is made up of days where speed remains almost constant and high throughout the whole day, the second day includes days where the average speed notably decreases in the evening and, finally, the third cluster contains those days where speed does not decrease in the evening,

1  
2  
3 but this decrease seems to happen in the morning. The second solution offered is similar to  
4 the first one but a fourth cluster is additionally detected. This fourth cluster includes days  
5 where the speed is not very high at any moment of the day and traffic jams arise both in  
6 the morning and in the evening. Finally, the third best ranked solution detects two clusters  
7 where the second one seems to include the days where the speed decreases in the evening.  
8  
9  
10  
11  
12

## 13 7 Simulation study

14  
15  
16  
17 The purpose of this section is to analyze the performance of the *autMIXMIX*, *autMIXCLA*  
18 and *autCLACLA* procedures as a function of the overlap between the groups. We have  
19 considered an example with clusters with true number of groups equal to 3, true eigenvalue  
20 ratio equal to 6,  $n = 150$ , and an average overlap which goes from 0.01 to 0.1, with step  
21 0.01. We have performed 100 simulations for each setting in dimensions  $p = 2$  and 6. In  
22 each simulation, with the aim of “visiting” as many as possible different  $\theta$  vectors, we have  
23 considered several random initializations (`nstarts=1000`) obtained from drawing  $k \times (p + 1)$   
24 observations that are arranged into  $k$  groups with  $p + 1$  observations. By using these  $k$   
25 groups, we obtain  $k$  initial  $m_j$  centers through their sample means and  $k$  initial scatter  
26 parameters  $S_j$  through their sample covariance matrices. In order to start with an initial  
27 admissible solution we have immediately applied the eigenvalue constraint. The values of  $c$   
28 which are considered go from 1 to 128 ( $c = \{2^0, 2^1, 2^2, \dots, 2^7\}$ ) and the values of  $k$  go from  
29 1 to 5. In order to avoid the randomness due to different starting points, both for mixture  
30 and classification likelihoods, for each simulation we have considered the same 1000 initial  
31 subsets for each value of  $c$ . For each simulation and each procedure, we have stored:  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44

- 45 1. The ARI between the true solution and the best-ranked solution found automatically;
- 46 47 2. The maximum ARI value between the true solution and the first two best-ranked  
48 solutions found automatically;
- 49 50 51 3. The maximum ARI value between the true solution and the first three best-ranked  
52 solutions found automatically.  
53  
54  
55  
56  
57  
58  
59  
60

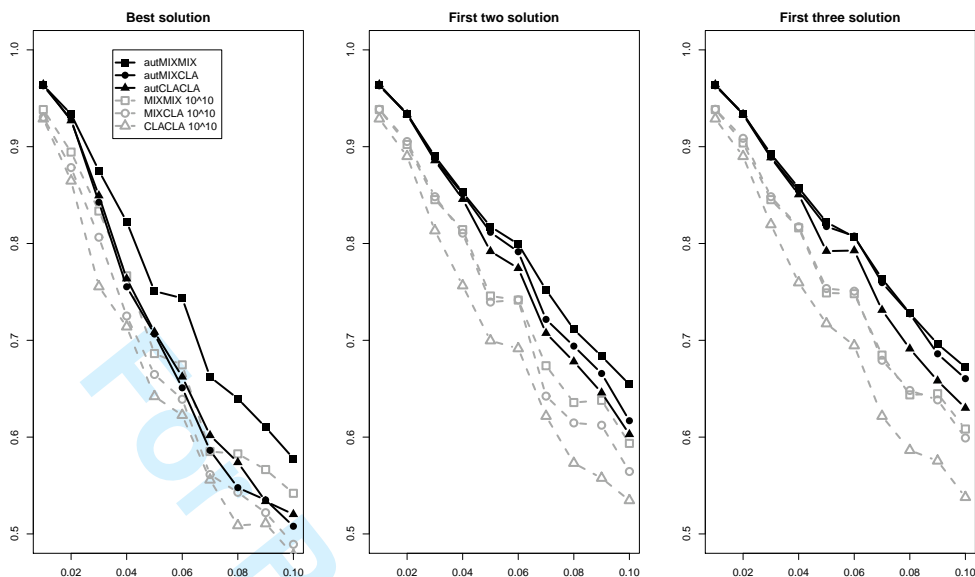


Figure 8: Average ARI index across 100 simulations as a function of cluster overlap when  $p = 2$ . The ARI indexes between the true solution and the best solution are shown in the *left panel*; with respect to the first two best-ranked solutions in the *central panel* and with respect to first three best-ranked ones in the *right panel*. The results of applying “traditional” ICL and BIC criteria (i.e., the use of MIX-MIX and MIX-CLA almost unconstrained with  $c = 10^{10}$ ) are shown in grey.

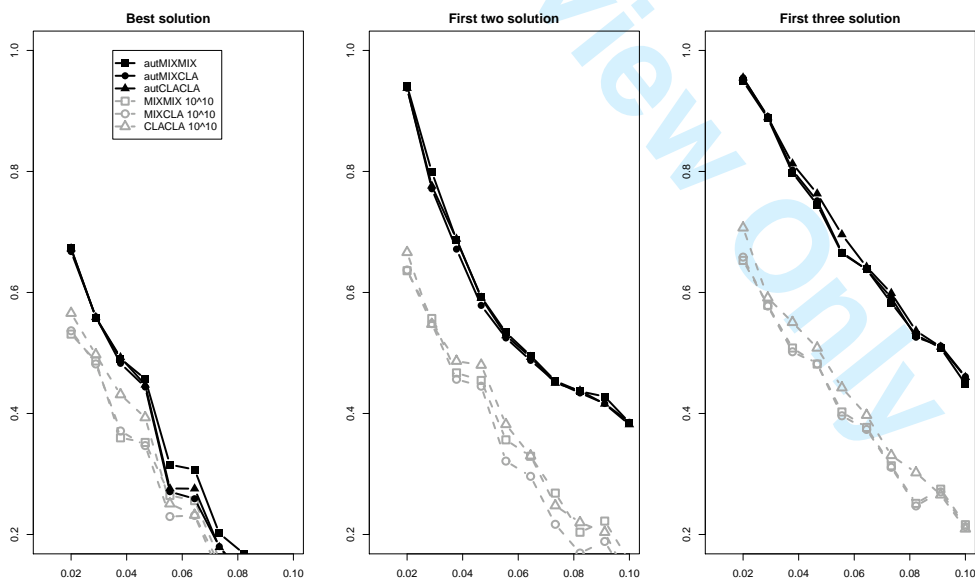


Figure 9: As Figure 8, but now with  $p = 6$ .

1  
2  
3 Figure 8 shows the average values of the above ARI over 100 simulations when the  
4 dimension of the simulated data set is  $p = 2$ . The left panel of the figure shows that  
5 as the average overlap increases the best performance is for the *autMIXMIX* procedure.  
6 More precisely, if the overlap is small the 3 information criteria give equivalent results.  
7 On the other hand, as the overlap increases, the gap between *autMIXMIX* and the other  
8 two information criteria increases. When we consider just the first solution, the curves for  
9 *autMIXCLA* and *autCLACLA* are virtually the same when the average overlap is smaller  
10 than 0.04 but the curve associated with *autMIXCLA* seems to be slightly higher than that  
11 of *autCLACLA* for high values of overlap. When we consider the first two solutions, the  
12 curve of *autMIXCLA* is always in between *autMIXMIX* and *autCLACLA*. Finally, when  
13 we consider the first three best solutions the curve of *autMIXCLA* is virtually equal to that  
14 of *autMIXMIX*, even if *autMIXMIX* still prevails for large overlap.

15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25 In order to show the effect of constraints, in Figure 8, we have also added the trajectories  
26 when we consider  $MIX_c$ -MIX,  $MIX_c$ -CLA and  $CLA_c$ -CLA with a very large  $c = 10^{10}$  value.  
27 This extreme  $c$  almost means that no constraint is imposed on the eigenvalue ratios of the  
28 scatter matrices. Therefore, these curves would essentially correspond to the traditional  
29 use of the BIC criteria (when using the MIX-MIX criterion) and the ICL (when using  
30 the MIX-CLA criterion). We can see that the constrained *autMIXMIX* procedure clearly  
31 outperforms traditional BIC and ICL criteria. Moreover, it appears that the gap between  
32 constrained and unconstrained curves seems to increase as the overlap increases and also if  
33 we increase the number of best possible solutions which are kept. Figure 9 also shows the  
34 average values of the above ARI over 100 simulations when the dimension of the simulated  
35 data sets is now increased to  $p = 6$ . Although this higher dimensional case yields smaller  
36 ARI values than those obtained when  $p = 2$ , we can see that the gap between constrained  
37 and unconstrained curves clearly increases in this new setting. Note also that very sensible  
38 ARI values are obtained, in spite of the higher problem dimensionality, when retaining the  
39 two and three best solutions returned from the proposed automated procedures. Finally, we  
40 can see that the observed differences from the application of the *autMIXMIX*, *autMIXCLA*  
41 and *autCLACLA* procedures are almost negligible here with  $p = 6$  case (especially in the  
42 central and right panels). The observed difference between the proposed methodology  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

and the traditional use of the BIC and ICL (unconstrained) criteria is likely to increase with the dimension  $p$  because spurious solutions are more likely to appear in these higher dimensional cases (see García-Escudero et al., 2014, 2015).

## 8 Conclusions and further directions

Three criteria for choosing the number of clusters in constrained model-based clustering have been proposed. Constraints make the associated (likelihood-based) target functions bounded and prevent the detection of non-interesting spurious solutions. Through our constraints we control the maximal ratio between the eigenvalues of the scatter matrices to be no greater than a fixed constant  $c$ , with  $c \geq 1$ . This constant serves to simultaneously control cluster departures from sphericity and heteroscedasticity among groups. In order to establish complexity-penalized criteria for choosing the number of clusters, we have taken into account the higher model complexity that a higher value of  $c$  entails. In our opinion, clustering should not be seen as a fully automatic task providing just one single solution and any user has to play an active role by specifying somehow the desired type of partitions. This specification can be done by fixing  $c$  depending on the clustering application. Additionally, a fully automated procedure producing a small and ranked list of optimal  $(k, c)$  pairs has been proposed and illustrated in a simulated data set and in three well-known real data examples. Our approach provides a trade off between the degree of automation of the clustering process and the user attitude towards a black-box output. If the user is prepared to look at more than one sensible solution, our procedure is still fully automatic. We have also added a new graphical display which enables us to clearly appreciate what are the most important solutions together with their stability.

A simulation study has also been carried out in order to validate the performance of our proposed methodology. The results of this simulation study have shown the importance of including constraints and have pointed out the general superiority of our proposal to other non-constrained penalized likelihood approaches, such as the BIC and the ICL criteria. Moreover, although with a small degree of overlap among the groups our three constrained criteria seem to give approximately the same results, the *autMIXMIX* criterion generally outperforms the other two when the overlap increases.

All the routines to obtain the results presented in this paper have been included in the FSDA toolbox for MATLAB which is freely downloadable from <http://www.riani.it/MATLAB> or from <http://fsda.jrc.ec.europa.eu>. More information about this routines can be found in the “online supplementary material”. There are some other research lines that deserve to be explored in the future. For instance, it will be interesting to extend this methodology to other clustering problems, such as clusterwise linear regression or mixtures of factor analyzers. We are also investigating how to apply this approach in robust clustering. Specifically, we are interested in extending the complexity-penalized likelihood approach described in this paper within the TCLUS framework (García-Escudero et al., 2008), in order to choose  $k$  and  $c$  together with the trimming level  $\alpha$ . This is not an easy problem since these three parameters,  $k$ ,  $c$  and  $\alpha$ , are clearly interrelated. For instance, a high value of  $\alpha$  could require a smaller  $k$  given that some small clusters may be completely trimmed. Besides, a high value of  $c$  may allow a certain fraction of background noise to be considered as an additional more scattered cluster and, thus, a higher  $k$  may be needed. Our feeling is that a reduced list of “sensible”  $(k, c, \alpha)$  triplets, where the user can choose the robust cluster partition that better fits his/her purposes, can also be automatically derived in an analogous way to that in Section 5. Neykov et al. (2007) and Li et al. (2016) have already considered trimmed version of the BIC in clustering and mixture modeling problems.

## Supplemental Materials

**Matlab Code:** The supplemental files for this article include Matlab programs which can be used to replicate the simulation study and the figures included in the article. File README contained in the zip file gives more details. (matlab\_code.zip, zip archive)

**Appendix:** The supplemental files include the Appendix which gives the proof of Theorem 3.1 and a graphical illustration of it. Additional tables and figures are given for the material presented in Sections 3, 4 and 5. The application of the proposed methodology to the well-known “Iris data set” is also given. The three best ranked solutions for the real data set example are summarized by using functional boxplots. An explanation about how to use the developed routines to carry out the proposed

methodology in the FSDA toolbox for MATLAB is also given (supplemental.pdf)

## Acknowledgments

García-Escudero's and Mayo-Isicar's research was supported in part by the Spanish Ministerio de Economía y Competitividad and FEDER, grant MTM2014-56235-C2-1-P, and by Consejería de Educación de la Junta de Castilla y León, grant VA212U13. Special thanks go to Anthony C. Atkinson, for helpful discussions and suggestions, and to Domenico Perrotta, for stimulating this research and for partially supporting it under the framework of the Automated Monitoring Tool (AMT) Project series. The authors thank the editor, the associate editor, and three anonymous referees for constructive comments.

## References

- Banfield, J. and Raftery, A. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.
- Biernacki, C., Celeux, G., and Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:719–725.
- Bryant, P. (1991). Large-sample results for optimization-based clustering methods. *J. Classif.*, 8:31–44.
- Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recogn.*, 28:781–793.
- Day, N. (1969). Estimating the components of a mixture of two normal distributions. *Biometrika*, 56:463–474.
- Forina, M., Armanino, C., Lanteri, S., and Tiscornia, E. (1983). Classification of olive oils from their fatty acid composition. In Martens, M. and Russwurm, H. J., editors, *Food Research and data Analysis*, pages 189–214. Applied Science Publishers, London.

- 1  
2  
3 Fraley, C. and Raftery, A. (2002). Model-based clustering, discriminant analysis, and  
4 density estimation. *J. Am. Stat. Assoc.*, 97:611–631.  
5  
6  
7  
8 Fraley, C. and Raftery, A. (2007). Bayesian regularization for normal mixture estimation  
9 and model-based clustering. *J. Classification*, 24:155–181.  
10  
11  
12 Fraley, C., Raftery, A., Scrucca, L., Murphy, T., and Fop, M. (2017). mclust version 5.3  
13 for R: Normal mixture modeling for model-based clustering, classification, and density  
14 estimation. available at <https://cran.r-project.org/web/packages/mclust/>.  
15  
16  
17  
18 Fritz, H., García-Escudero, L., and Mayo-Iscar, A. (2013). A fast algorithm for robust  
19 constrained clustering. *Comput. Stat. Data Anal.*, 61:124–136.  
20  
21  
22  
23 García-Escudero, L. and Gordaliza (2005). A proposal for robust curve clustering. *J.*  
24 *Classification*, 22:185–201.  
25  
26  
27 García-Escudero, L., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2008). A general  
28 trimming approach to robust cluster analysis. *Ann. Statist.*, 36:1324–1345.  
29  
30  
31 García-Escudero, L., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2015). Avoiding  
32 spurious local maximizers in mixture modeling. *Stat. Comput.*, 25:619–633.  
33  
34  
35  
36 García-Escudero, L., Gordaliza, A., and Mayo-Iscar, A. (2014). A constrained robust  
37 proposal for mixture modeling avoiding spurious solutions. *Adv. Data Anal. Classif.*,  
38 8:27–43.  
39  
40  
41  
42 Hathaway, R. (1985). A constrained formulation of maximum likelihood estimation for  
43 normal mixture distributions,. *Ann. Statist.*, 13:795–800.  
44  
45  
46  
47 Hennig, C. and Liao, T. (2013). How to find an appropriate clustering for mixed-type  
48 variables with application to socio-economic stratification,. *J. Roy. Statist. Soc. Ser. C*,  
49 62:309–369.  
50  
51  
52  
53 Ingrassia, S. and Rocci, R. (2007). Constrained monotone EM algorithms for finite mixture  
54 of multivariate gaussians,. *Comput. Stat. Data Anal.*, 51:5339–5351.  
55  
56  
57  
58  
59  
60

- 1  
2  
3 Li, M., Xiang, S., and Yao, W. (2016). Robust estimation of the number of components  
4 for mixtures of linear regression models,. *Computation. Stat.*, 31:1539–1555.  
5  
6  
7 Maitra, R. (2009). Initializing partition-optimization algorithms. *IEEE/ACM Trans. Com-*  
8 *put. Biol. Bioinf.*, 6:1447–15.  
9  
10  
11 Maitra, R. and Melnykov, V. (2010). Simulating data to study performance of finite mixture  
12 modeling and clustering algorithms. *J. Comput. Graph. Stat.*, 19:354– 376.  
13  
14  
15 McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. John Wiley Sons, Ltd.  
16  
17  
18 McNicholas, P., ElSherbiny, A., Jampani, K., McDaid, A., Murphy, T., and Banks, L.  
19 (2015). pgmm: Parsimonious gaussian mixture models. *available at [https://cran.r-](https://cran.r-project.org/web/packages/pgmm/)*  
20 *project.org/web/packages/pgmm/*.  
21  
22  
23  
24 Milligan, G. and Cooper, M. (1985). An examination of procedures for determining the  
25 number of clusters in a data set. *Psychometrika*, 50:159–179.  
26  
27  
28 Neykov, N., Filzmoser, P., Dimova, R., and Neytchev, P. (2007). Robust fitting of mixtures  
29 using the trimmed likelihood estimator. *Comput. Stat. Data Anal.*, 52:299–308.  
30  
31  
32  
33 Riani, M., Cerioli, A., Perrotta, D., and Torti, F. (2015). Simulating mixtures of multivari-  
34 ate data with fixed cluster overlap in FSDA library. *Adv. Data Anal. Classif.*, 9:2015.  
35  
36  
37 Riani, M., Perrotta, D., and Torti, F. (2012). FSDA: a matlab toolbox for robust analysis  
38 and interactive data exploration,. *Chemometr. Intell. Lab. Syst.*, 116:17–32.  
39  
40  
41  
42 Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of  
43 cluster analysis,. *J. Comput. Appl. Math.*, 20:53–65.  
44  
45  
46 Sun, Y. and Genton, M. G. (2011). Functional boxplots,. *J. Comput. Graph. Stat.*,  
47 20:316334.  
48  
49  
50  
51 Symons, M. (1981). Clustering criteria and multivariate normal mixtures. *Biometrics*,  
52 37:35–43.  
53  
54  
55 Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of data clusters  
56 via the GAP statistic. *J. Roy. Statist. Soc. Ser. B*, 63:411423.  
57  
58  
59  
60

## Appendix to “Finding the Number of Groups in Model-Based Clustering via Constrained Likelihoods”

Andrea Cerioli, Luis A. García-Escudero,  
Agustin Mayo-Iscar and Marco Riani

The materials in this document supplement the information presented in the manuscript “Finding the Number of Groups in Model-Based Clustering via Constrained Likelihoods”. Section A provides the proof of Theorem 3.1 and a graphical illustration of it. Section B gives the optimal  $c$  values for each  $k$ , when using  $CLA_c$ - $CLA$  and  $MIX_c$ - $MIX$ , for the data set in Figure 1 and the associated “contour plot” for the three constrained clustering criteria. Section C shows the tables that have been applied to obtain the results presented in Section 5.2.1 and a graph with the first 4 discarded “spurious” solutions for the data set considered in that section. Section D provides the car-bike plot for the Hennig and Liao’s type of data in Section 5.2.2. The application of the proposed methodology to the well-known “Iris data set” is given in Section E. Section F summarizes the three best ranked solutions obtained for the “road traffic data” in Section 6 by using functional boxplots. Finally, all the routines to obtain the results presented in this paper, and included in the FSDA toolbox for MATLAB, are briefly presented in Section G.

### A Proof of Theorem 3.1 and graphical illustration

**Proof of Theorem 3.1:** In order to prove that result, let us first consider

$$B_t^* = \{(\lambda_1, \dots, \lambda_D) : \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_D \leq c\lambda_1 \text{ and } 0 \leq \lambda_l \leq t\}.$$

We have

$$\begin{aligned} \text{Vol}(B_t^*) = & \int_0^{t/c} \int_{\lambda_1}^{c\lambda_1} \int_{\lambda_2}^{c\lambda_1} \dots \int_{\lambda_{D-1}}^{c\lambda_1} d\lambda_D d\lambda_{D-1} \dots d\lambda_2 d\lambda_1 \\ & + \int_{t/c}^t \int_{\lambda_1}^t \int_{\lambda_2}^c \dots \int_{\lambda_{D-1}}^c d\lambda_D d\lambda_{D-1} \dots d\lambda_2 d\lambda_1. \end{aligned}$$

Given that

$$\int_{\lambda_{D-q}}^t \dots \int_{\lambda_{D-1}}^t d\lambda_D d\lambda_{D-1} \dots d\lambda_{D-q+1} = \frac{(t - \lambda_{D-q})^q}{q!},$$

we can see that

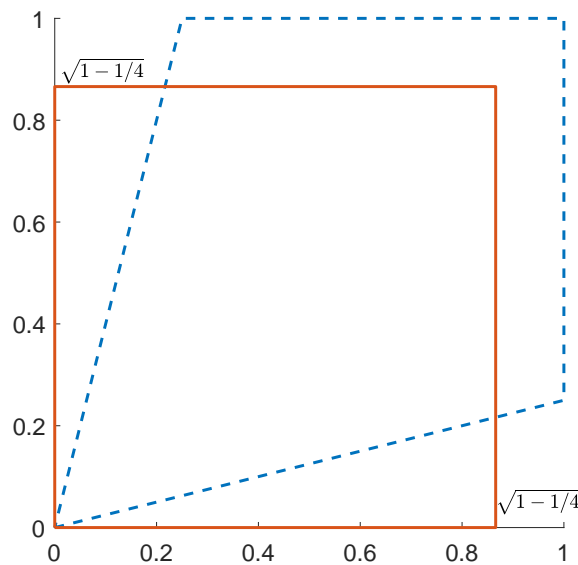
$$\begin{aligned} \text{Vol}(B_t^*) &= \int_0^{t/c} \frac{(c\lambda_1 - \lambda_1)^{D-1}}{(D-1)!} d\lambda_1 + \int_{t/c}^t \frac{(b - \lambda_1)^{D-1}}{(D-1)!} d\lambda_1 \\ &= \frac{(c-1)^{D-1} (t/c)^D}{D!} + \frac{(t - t/c)^{D-1}}{D!} = \frac{t^D}{D!} \left(1 - \frac{1}{c}\right)^{D-1}. \end{aligned}$$

There are  $D!$  different orderings of  $\lambda_1, \dots, \lambda_D$  and, thus, we have (by considering obvious symmetry arguments) that

$$\text{Vol}(B_t) = D! \times \text{Vol}(B_t^*) = t^D \left(1 - \frac{1}{c}\right)^{D-1}.$$

Thus, final result follows from the trivial fact that  $\text{Vol}(A_t) = t^D$ .  $\square$

Figure 1 shows a graphical interpretation when  $t = 1$ ,  $D = 2$  (that is one group of two-dimensional observations) and  $c = 4$ . In this case  $\text{Vol}(A_t) = 1$  and the ratio  $\text{Vol}(B_t)/\text{Vol}(A_t)$  equals the area of a square of side  $[0, \sqrt{1 - 1/c}]$ .



**Figure 1:** Illustration of Theorem 3.1 when  $t = 1$ ,  $D = 2$  and  $c = 4$ . The surface enclosed within dashed lines corresponds to  $B_1$ . Since  $\text{Vol}(A_1) = 1$ , the ratio  $\text{Vol}(B_1)/\text{Vol}(A_1)$  equals the area of the square  $[0, \sqrt{1 - 1/4}] \times [0, \sqrt{1 - 1/4}]$  shown with solid lines.

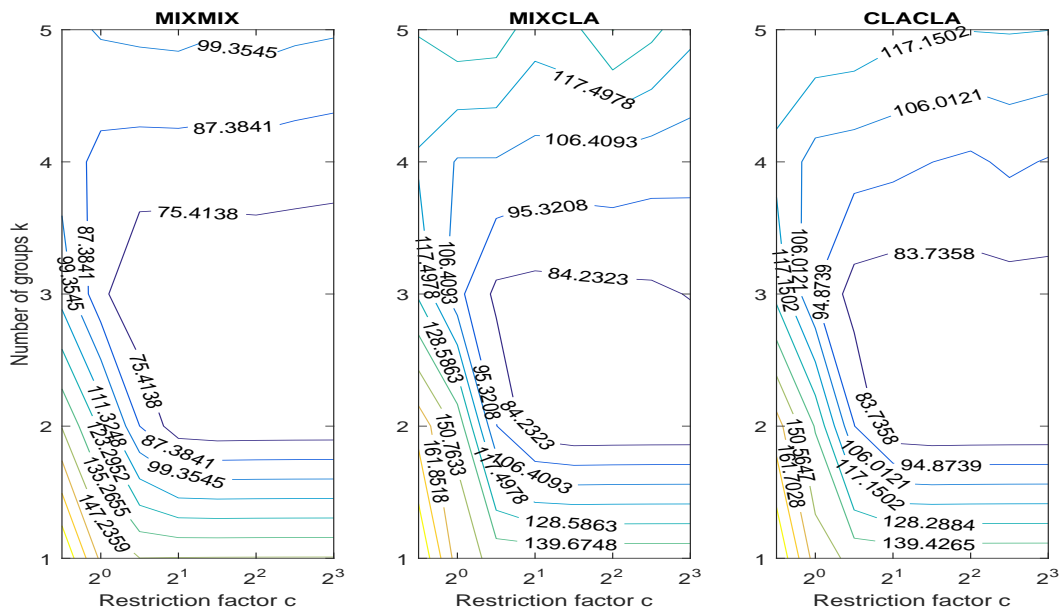
**Table 1:** Optimal  $c$  values for each  $k$  when using  $CLA_c$ -CLA and  $MIX_c$ -MIX for the data set shown in Figure 1.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
optimal $c$ for $CLA_c$ -CLA	4	16	8	128	1
optimal $c$ for $MIX_c$ -MIX	4	16	8	128	128

## B Optimal $c$ for each $k$ and “contour plot” for the data set in Figure 1

The optimal  $c$  for each  $k$  is shown for the data set in Figure 1 in the first line of Table 1 when using  $CLA_c$ -CLA and in the second line when using  $MIX_c$ -MIX.

Figure 2 shows the associated contour plots that summarize the resulting monitoring process for the data set shown in Figure 1 and for our three constrained clustering criteria.



**Figure 2:** Contour plots for the  $(k, c) \mapsto F_m(k, c)$  functions when the  $m =$  MM, MC and CC criteria are applied.

## C Tables for Section 5.2.1 and graph with the first 4 discarded “spurious” solutions

The starting point of the analysis done in Section 5.2.1 is the matrix which contains the values of  $CLA_c$ -CLA for all  $(k, c)$  pairs (given in Table 2). The

**Table 2:** Matrix of  $K \times C$  possible of  $\text{CLA}_c\text{-CLA}$  ( $k, c$ ) pairs to be explored.

	$c = 1$	$c = 2$	$c = 4$	$c = 8$	$c = 16$	$c = 32$	$c = 64$	$c = 128$
$k = 1$	195.12	156.58	147.35	147.70	147.87	147.95	147.98	148.00
$k = 2$	166.19	138.49	95.16	74.25	72.60	72.95	73.13	73.22
$k = 3$	125.06	94.65	79.05	77.13	78.46	79.12	79.45	79.61
$k = 4$	114.24	101.58	99.57	98.01	94.95	92.85	91.86	89.66
$k = 5$	125.08	124.92	122.06	116.50	114.54	112.13	111.87	109.24

**Table 3:** Matrix of  $K \times (C - 1)$  containing the values of the ARI for two consecutive values of  $c$  (for fixed  $k$ ).

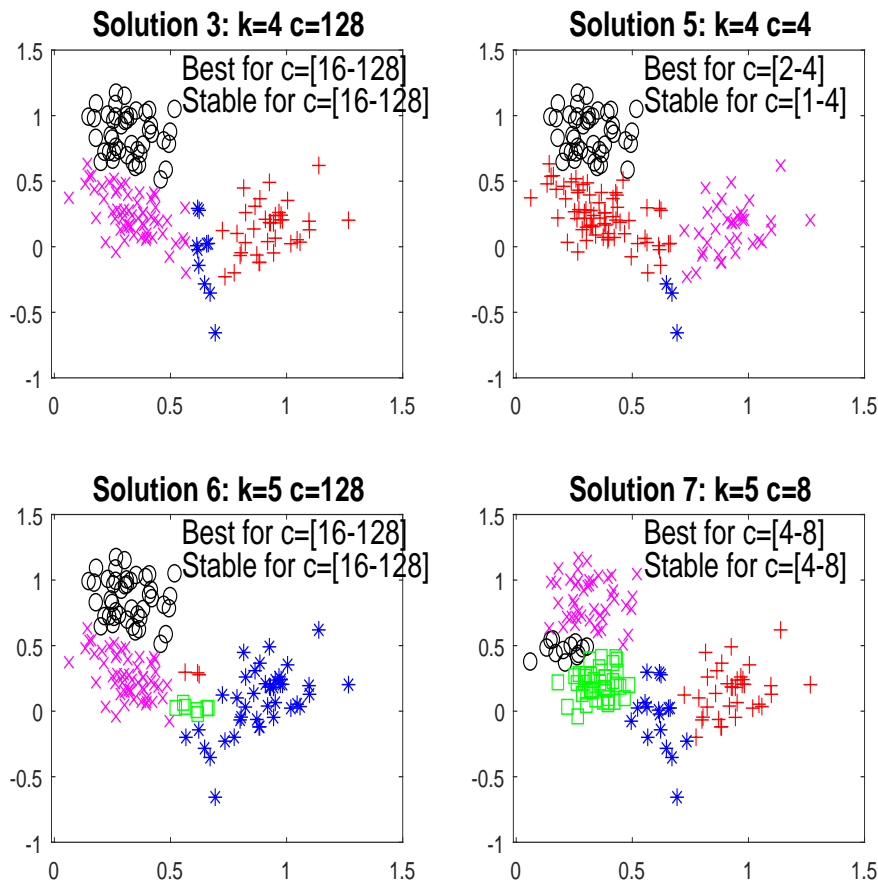
$c :$	1-2	2-4	4-8	8-16	16-32	32-64	64-128
$k = 1$	1	1	1	1	1	1	1
$k = 2$	0.17	0.89	1	0.84	1	1	1
$k = 3$	1	0.86	1	1	1	1	1
$k = 4$	0.87	0.93	0.57	0.71	0.99	0.96	0.99
$k = 5$	0.65	0.77	0.92	0.68	0.99	1	0.81

threshold given in equation (7) in the manuscript is obtained by considering the matrix which contains the ARI indexes for two consecutive values of  $c$  given  $k$  (see Table 3).

Figure 3 shows the first 4 discarded “spurious” solutions for the simulated data set in Section 5.2.1. We can see that these discarded solutions either include clusters made up with a few almost collinear or concentrated observations (solutions 3 and 5), or correspond to solutions close to one already detected “optimal” partition (solution 7).

## D Car-bike plot for the Hennig and Liao’s type of data

The car-bike plot (given in Figure 4) presents a nice summary of the solutions seen so far because it shows with a tall rectangle the first best ranked solution with 3 groups. The longest car is for the homoscedastic solution with 5 groups. The car-bike plot has the additional advantage of showing clearly that while the second best ranked solution with 4 groups is best just for a particular value of  $c$ , the homoscedastic solution is best in the interval  $c$   $[1, 16]$ . The height of the rectangle for the fourth best ranked solution is very small reflecting its low order in the ranking. The fifth best ranked solution is local and is shown as a “bike”.



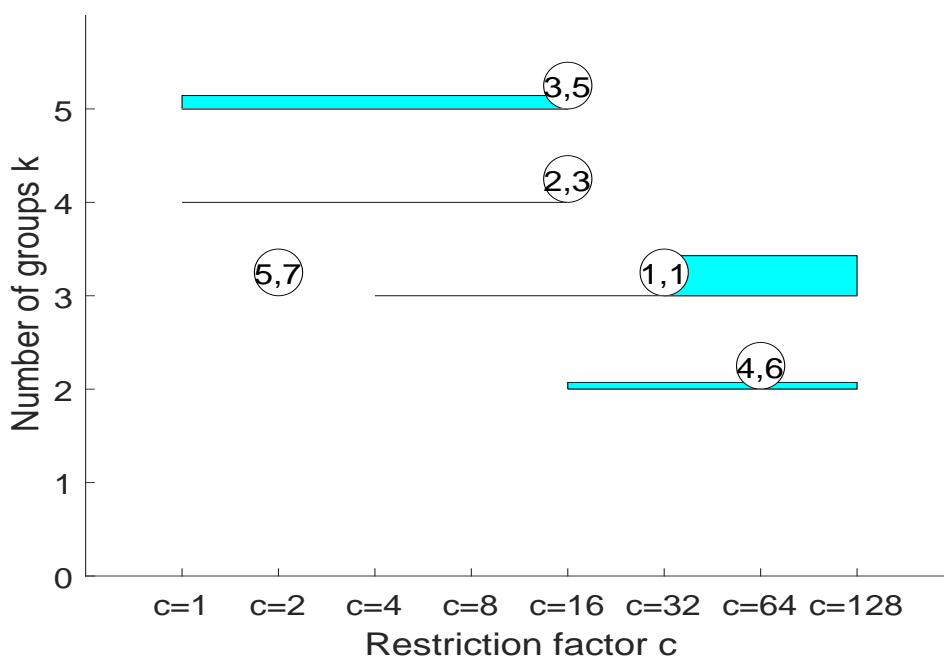
**Figure 3:** The first four discarded “spurious” solutions detected when using the *autCLA-CLA* procedure for the simulated data set displayed in Figure 1.

## E Application to the “Iris data set”

The “Iris data set”, originally collected by Anderson (1935) and first analyzed by Fisher (1936), is considered in this example. We have applied the proposed procedure to this well-known four-dimensional ( $p = 4$ ) data set. Figure 5 shows the ranked list of “sensible” cluster partitions which are automatically found when using the *autMIXMIX* procedure. For purposes of clarity we show just the scatter plots of sepal width (SW) vs sepal length (SL), petal length (PL) vs sepal width (SW) and petal width (PW) vs petal length (PL).

We can see that the most clear two-component partition is the first offered by our method. In this partition “Iris setosa” is well-separated from “Iris virginica” and “Iris versicolor” (that are not so easy to separate). The second proposed partition essentially coincides with the three actual species.

With respect to the third best ranked solution, we recall that this “Iris data set” was initially collected by Anderson with the aim of seeing whether

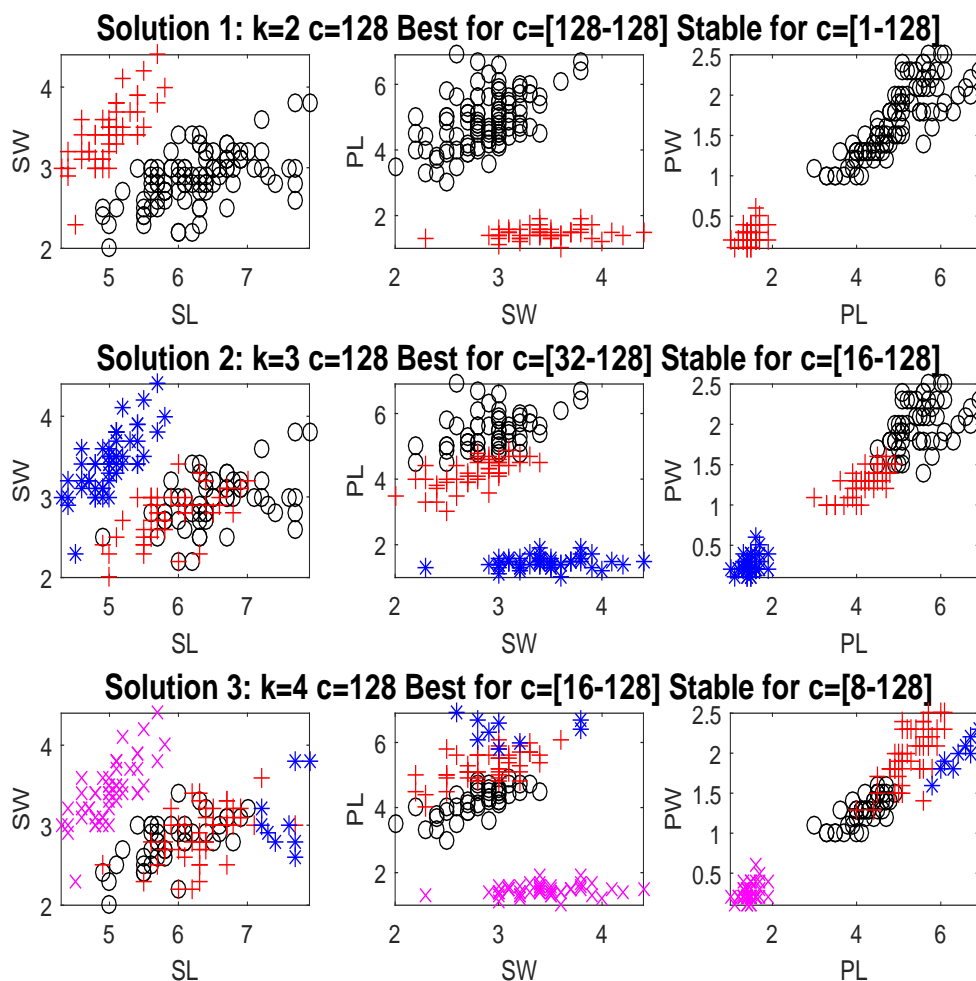


**Figure 4:** Car-bike plot for the when using the *autMIXMIX* for a data set similar to that in Hennig and Liao (2013).

there was “evidence of continuing evolution in any group of plants”. Thus, it is interesting to evaluate whether “virginica” species should be split into two subspecies or not. In their Section 3.11, McLachlan and Peel (2000) focused only on the 50 virginica iris data and fitted a mixture of  $k = 2$  normal components to them. They listed 15 possible local ML maximizers together with different quantities summarizing aspects as the separation between clusters, the size of the smallest cluster and the determinants of the scatter matrices corresponding to these solutions. After analyzing this information, the so-called “S1” solution is chosen as the most sensible one among the local ML maximizers. It is very nice to see that our third best ranked solution exactly detects a four-component partition where the “virginica” species is automatically split into 2 components in such a way that it coincides with the “S1” partition already proposed in McLachlan and Peel (2000).

## F Functional boxplots for the three best ranked solutions for the “road traffic data”

Figures 6, 7 and 8 summarize the three best ranked solutions by using functional boxplots as introduced in Sun and Genton (2011) (we consider the `fbplot` function in the `fda` package with its default values; see Ramsay et al. (2014)).



41 **Figure 5:** Best-ranked partitions when using *autMIXMIX* procedure criterion for the  
42 “Iris data set”. Only some few pairs plots are shown for each cluster partition.

## 43 G Computer code

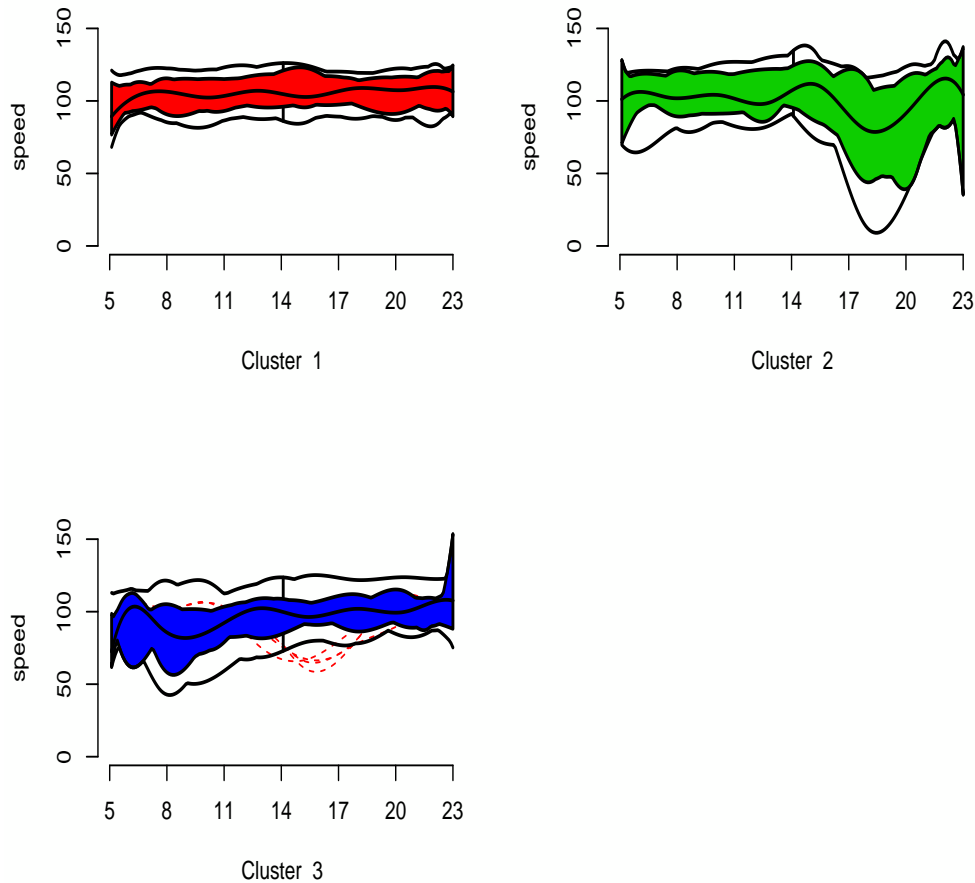
44 All the routines to obtain the results presented in this paper have been  
45 included in the FSDA toolbox for MATLAB which is freely downloadable  
46 from the web address

47 <http://www.riani.it/MATLAB>

48 or from

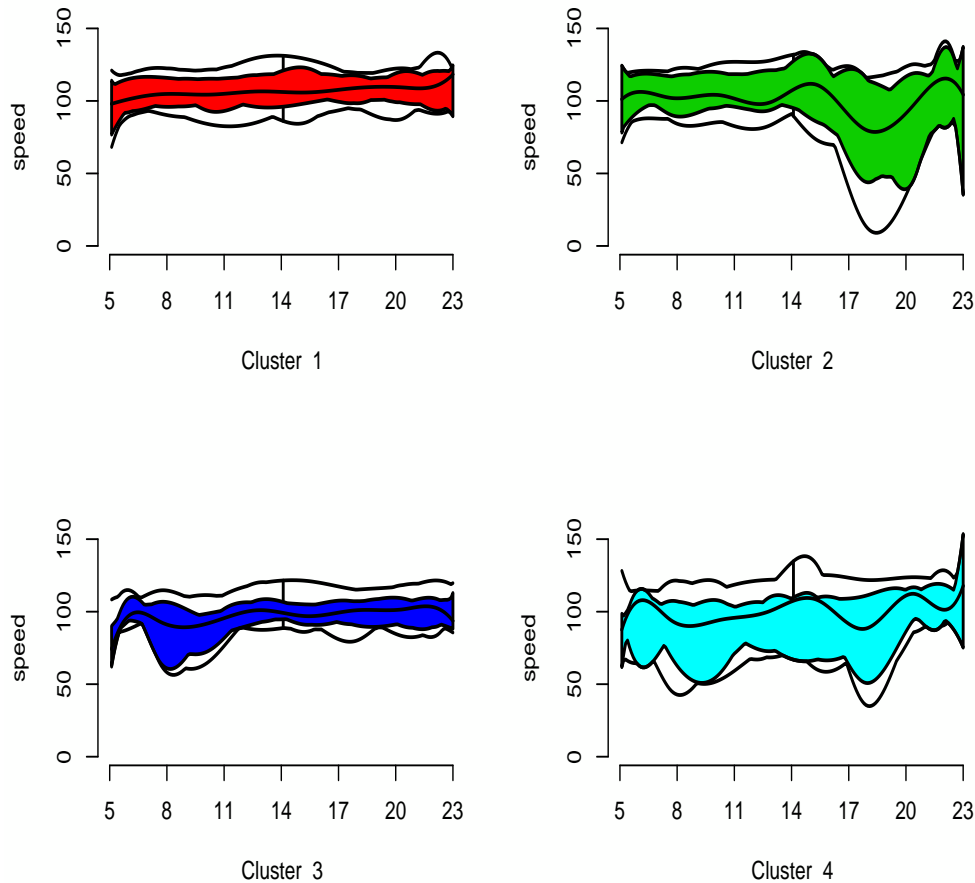
49 <http://fsda.jrc.ec.europa.eu> .

50 An explanation of the available routines is as follows:



**Figure 6:** First of the three best-ranked partitions when using the *autMIXMIX* for the “road traffic data” with  $k = 3$  and  $c = 128$  represented by using functional Box-plots.

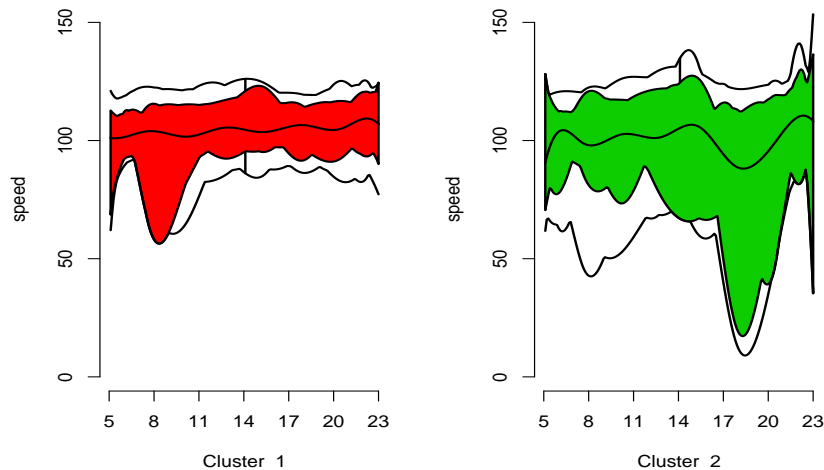
1. The routine `out=tclustIC(Y,varargin)` takes as input a data matrix containing  $n$  observations on  $p$  variables and computes the values of BIC (MIXMIX), ICL (MIXCLA) or CLA (CLACLA), for different values of  $k$  (number of groups) and different values of  $c$  (constraint factor). In `varargin` it is possible to specify the range of mixture components, the values of the constraint factor, the information criteria to use, the trimming level, the number of subsamples to extract, the number of refining iterations, the tolerance for the refining steps, the number of cores to use in parallel computing, and another series of small options. The output of this routine is a structure which contains a series of matrices which for each combination of values of  $k$  and  $c$



**Figure 7:** Second of the three best-ranked partitions when using the *autMIXMIX* for the “road traffic data” with  $k = 4$  and  $c = 128$ .

gives the associated information criterion.

- The routine `out = tclustICsol(IC,varargin)` takes as input the output of function `tclustIC` and extracts the first best solutions. In `varargin` it is possible to specify the information criterion to use, the number of solutions (`NumberOfBestSolutions`) to consider, the threshold to identify spurious solutions and another series of small options. The output of this routine is a structure which contains a MATLAB cell of size `NumberOfBestSolutions`- $\times$ -5 with the details of the best solutions and a matrix of adjusted Rand indexes among the best solutions associated with the requested information criteria.



**Figure 8:** Third of the three best-ranked partitions when using the *autMIXMIX* for the “road traffic data” with  $k = 2$  and  $c = 128$ .

3. The routine `tclustICplot(IC,varargin)` plots information criteria as a function of  $c$  and  $k$ . In other terms, `tclustICplot` takes as input the output of function `tclustIC` (that is a series of matrices which contain the values of the information criteria BIC/ICL/CLA) and plots them as a function of  $c$  or of  $k$ . Similarly to many of the other graphical routines included inside FSDA, the plot enables interaction in the sense that, if option `databrush` has been activated, it is possible to click on a point in the plot and to see the associated classification in the scatter plot matrix.

At the end of the preamble of each `.m` file (and also inside the corresponding `.html` file) there are a series of examples containing chunks of code which can reproduce all the figures shown in the current paper.

4. The routine `carbikeplot` takes as input the output of function `tclustICsol` and enables us to create the car-bike plot.

Finally, in agreement with all the other routines present inside FSDA toolbox, the above procedures have an extensive documentation both inside the `.m` file and in the corresponding `.html` file. The help system of the FSDA toolbox is completely integrated with that of MATLAB and is almost indistinguishable from that of the official toolboxes provided by Mathworks.

## References

- Anderson, E. (1935). The irises of the gaspe peninsula. *Bulletin of the American Iris Society*, 59:25.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188.
- Hennig, C. and Liao, T. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification,. *J. Roy. Statist. Soc. Ser. C*, 62:309–369.
- McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. John Wiley Sons, Ltd.
- Ramsay, J., Wickham, H., Graves, S., and Hooker, G. (2014). fda: Functional data analysis. available at <https://cran.r-project.org/web/packages/fda/>.
- Sun, Y. and Genton, M. G. (2011). Functional boxplots,. *J. Comput. Graph. Stat.*, 20:316334.