Statistics and Computing (2024) 34:100
https://doi.org/10.1007/s11222-024-10410-y

ORIGINAL PAPER

Improving model choice in classification: an approach based on clustering of covariance matrices

David Rodríguez-Vítores[1] · Carlos Matrán[1]

Received: 22 February 2023 / Accepted: 14 February 2024 / Published online: 19 March 2024

Abstract

This work introduces a refinement of the Parsimonious Model for fitting a Gaussian Mixture. The improvement is based on the consideration of clusters of the involved covariance matrices according to a criterion, such as sharing Principal Directions. This and other similarity criteria that arise from the spectral decomposition of a matrix are the bases of the Parsimonious Model. We show that such groupings of covariance matrices can be achieved through simple modifications of the CEM (Classification Expectation Maximization) algorithm. Our approach leads to propose Gaussian Mixture Models for model-based clustering and discriminant analysis, in which covariance matrices are clustered according to a parsimonious criterion, creating intermediate steps between the fourteen widely known parsimonious models. The added versatility not only allows us to obtain models with fewer parameters for fitting the data, but also provides greater interpretability. We show its usefulness for model-based clustering and discriminant analysis, providing algorithms to find approximate solutions verifying suitable size, shape and orientation constraints, and applying them to both simulation and real data examples.

Keywords Parsimonious model · Gaussian mixture model · Bayesian information criterion · Model-based classification · EM algorithm

1 Introduction

[1]In this paper we introduce methodological applications arising of cluster analysis of covariance matrices. Throughout, we will show that appropriate clustering criteria on these objects provide useful tools in the analysis of classic problems in Multivariate Analysis. The chosen framework is that of multivariate classification under a Gaussian Mixture Model, a setting where a suitable reduction of the involved parameters is a fundamental goal leading to the Parsimonious Model. We focus on this hierarchized model, designed to explain data with a minimum number of parameters, by introducing intermediate categories associated with clusters of covariance matrices.

Gaussian Mixture Models approaches to discriminant and cluster analysis are well-established and powerful tools in multivariate statistics. For a fixed number K, both methods aim to fit K multivariate Gaussian distributed components to a data set in $\mathbb{R}^d$, with the key difference that labels providing the source group of the data are known (supervised classification) or unknown (unsupervised classification). In the supervised problem, we handle a data set with N observations $y_1, \dots, y_N$ on $\mathbb{R}^d$ and associated labels $z_{i,k}$, $i = 1, \dots, N$, $k = 1, \dots, K$, where $z_{i,k} = 1$ if the observation y_i belongs to the group k and 0 otherwise. Denoting by $\phi(\cdot|\mu, \Sigma)$ the density of a multivariate Gaussian distribution on $\mathbb{R}^d$ with mean μ and covariance matrix Σ, we seek to maximize the complete log-likelihood function

$$CL\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\Big) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{i,k} \log\Big(\pi_k \phi(y_i|\mu_k, \Sigma_k)\Big), \qquad (1)$$

with respect to the weights $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ with $0 \leq \pi_k \leq 1$, $\sum_{k=1}^{K} \pi_k = 1$, the means $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)$ and the covariance matrices $\boldsymbol{\Sigma} = (\Sigma_1, \dots, \Sigma_K)$. In the unsupervised

[1] The research leading to these results received funding from MCIN/AEI/10.13039/501100011033/FEDER under Grant Agreement No PID2021-128314NB-I00.

✉ David Rodríguez-Vítores
david.rodriguez.vitores@uva.es

Carlos Matrán
carlos.matran@uva.es

[1] Department of Statistics and Operational Research and IMUVA, University of Valladolid, Paseo de Belén 7, Valladolid 47011, Spain

problem the labels $z_{i,k}$ are unknown, and fitting the model involves the maximization of the log-likelihood function

$$L\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\Big) = \sum_{i=1}^{N} \log\Bigg(\sum_{k=1}^{K} \pi_k \phi(y_i|\mu_k, \Sigma_k)\Bigg) \tag{2}$$

with respect to the same parameters. This maximization is more complex, and it is usually performed via the EM algorithm (Dempster et al. 1977), where we repeat iteratively the following two steps. The E step, which consists in computing the expected values of the unobserved variables $z_{i,k}$ given the current parameters, and the M step, in which we are looking for the parameters maximizing the complete log-likelihood (1) for the values $z_{i,k}$ computed in the E step. Therefore, both model-based techniques require the maximization of (1), for which optimal values of the weights and the mean are easily computed:

$$n_k = \sum_{i=1}^{N} z_{i,k} \quad \pi_k = \frac{n_k}{N} \quad \mu_k = \frac{\sum_{i=1}^{N} z_{i,k}\, y_i}{n_k}. \tag{3}$$

With these optimal values, if we denote $S_k = (1/n_k)\sum_{i=1}^{N} z_{i,k}$ $(y_i - \mu_k)(y_i - \mu_k)^T$, the problem of maximizing (1) with respect to $\Sigma_1, \ldots, \Sigma_K$ is equivalent to the problem of maximizing

$$(\Sigma_1, \ldots, \Sigma_K) \mapsto \sum_{k=1}^{K} \log\Big(W_d\big(n_k S_k|n_k, \Sigma_k\big)\Big) \tag{4}$$

where $W_d(\cdot|n_k, \Sigma_k)$ is the d-dimensional Wishart distribution with parameters n_k, Σ_k. For even moderate dimension d, the large number of involved parameters in relation with the size of the data set could result in a poor behavior of standard unrestricted methods. In order to improve the solutions, regularization techniques are often invoked. In particular, many authors have proposed estimating the maximum likelihood parameters under some additional constraints on the covariance matrices $\Sigma_1, \ldots, \Sigma_K$, which lead us to solve the maximization of (4) under these constraints. Between these proposals, a prominent place is occupied by the so called **Parsimonious Model**, a broad set of hierarchized constraints capable of adapting to conceptual situations that may occur in practice.

A common practice in multivariate statistics consists in assuming that covariance matrices share a common part of their structure. For example, if $\Sigma_1 = \ldots = \Sigma_K = I_d$, the clustering method described in (2) gives just the k-means. If we assume common covariance matrices $\Sigma_1 = \ldots = \Sigma_K = \Sigma$, the procedure coincides with linear discriminant analysis (LDA) in the supervised case (1), and with the method proposed in Friedman and Rubin (1967) in the unsupervised case (2). General theory to organize these relationships between covariance matrices is based on the spectral decomposition, beginning with the analysis of Common Principal Components (Flury 1984, 1988). In the discriminant analysis setting, the use of the spectral decomposition was first proposed in Flury et al. (1994), and in the clustering setting in Banfield and Raftery (1993). The term “Parsimonious model" and the fourteen levels given in Table 1 were introduced in Celeux and Govaert (1995) for the clustering setting and later, in Bensmail and Celeux (1996), for the discriminant setup.

Given a positive definite covariance matrix Σ_k, the spectral decomposition of reference is

$$\Sigma_k = \gamma_k \beta_k \Lambda_k \beta_k^T$$

where $\gamma_k = \det(\Sigma_k)^{1/d} > 0$ governs the size of the groups, Λ_k is a diagonal matrix with positive entries and determinant equal to 1 that controls the shape, and β_k is an orthogonal matrix that controls the orientation. Given K covariance matrices $\Sigma_1, \ldots, \Sigma_K$, the spectral decomposition enables to establish the fourteen different parsimonious levels in Table 1, allowing differences or not in the parameters associated to size, shape and orientation. To fit a Gaussian Mixture Model under a parsimonious level $\mathscr{M}$ in the Table 1, we must face the maximization of (4) under the parsimonious restriction. That is, we should find

$$\hat{\boldsymbol{\Sigma}} = \underset{\boldsymbol{\Sigma}\in\mathscr{M}}{\operatorname{argmax}} \sum_{k=1}^{K} \log\Big(W_d\big(n_k S_k|n_k, \Sigma_k\big)\Big), \tag{5}$$

where we say that $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_K) \in \mathscr{M}$ if the K covariance matrices verify the level. We should remark that the Common Principal Components model (Flury 1984, 1988) plays a key role in this hierarchy, which in any case is based on simple geometric interpretations.

Restrictions are also often used to solve a well-known problem that appears in model-based clustering, the unboundedness of the log-likelihood function (2). With no additional constraints, the problem of maximizing (2) is not even well defined, a fact that could lead to uninteresting spurious solutions, where some groups would be associated to a few, almost collinear, observations. Although we will also use these restrictions, we will not discuss on this line in this work. A review of approaches for dealing with this problem can be found in García-Escudero et al. (2017).

The aim of this paper is to introduce a generalization of equation (5), that allows us to give a likelihood-based classification associated to intermediate parsimonious levels. Let $G \in \{1, \ldots, K\}$ and $\boldsymbol{u} = (u_1, \ldots, u_K)$ be any vector in $\{1, \ldots, G\}^K$. Given a parsimonious level $\mathscr{M}$, we can formulate a model in which we assume that the theoretical covariance matrices $\Sigma_1, \ldots, \Sigma_K$ verify a parsimonious level $\mathscr{M}$ within each of the G classes defined by $\boldsymbol{u}$. For instance, let

Table 1 Parsimonious levels based on the spectral decomposition of $\Sigma_1, \dots, \Sigma_K$.

Name	Σ_k	Size	Shape	Orientation	Parameters
EII	γI	Equal	Spherical	–	1
VII	$\gamma_k I$	Variable	Spherical	–	K
EEI	$\gamma \Lambda$	Equal	Equal	Canonical	$1 + (d-1)$
EVI	$\gamma \Lambda_k$	Equal	Variable	Canonical	$1 + K(d-1)$
VEI	$\gamma_k \Lambda$	Variable	Equal	Canonical	$K + (d-1)$
VVI	$\gamma_k \Lambda_k$	Variable	Variable	Canonical	$K + K(d-1)$
EEE	$\gamma \beta \Lambda \beta^T$	Equal	Equal	Equal	$1 + (d-1) + d(d-1)/2$
EEV	$\gamma \beta_k \Lambda \beta_k^T$	Equal	Equal	Variable	$1 + (d-1) + Kd(d-1)/2$
EVE	$\gamma \beta \Lambda_k \beta^T$	Equal	Variable	Equal	$1 + K(d-1) + d(d-1)/2$
VEE	$\gamma_k \beta \Lambda \beta^T$	Variable	Equal	Equal	$K + (d-1) + d(d-1)/2$
VVE	$\gamma_k \beta \Lambda_k \beta^T$	Variable	Variable	Equal	$K + K(d-1) + d(d-1)/2$
EVV	$\gamma \beta_k \Lambda_k \beta_k^T$	Equal	Variable	Variable	$1 + K(d-1) + Kd(d-1)/2$
VEV	$\gamma_k \beta_k \Lambda \beta_k^T$	Variable	Equal	Variable	$K + (d-1) + Kd(d-1)/2$
VVV	$\gamma_k \beta_k \Lambda_k \beta_k^T$	Variable	Variable	Variable	$K(1 + (d-1) + d(d-1)/2)$

$K = 7$, $G = 3$, $\mathscr{M}$ = VVE and take $\boldsymbol{u} = (1, 1, 2, 3, 1, 2, 1)$. This implies

$$\Sigma_k = \gamma_k \beta_1 \Lambda_k \beta_1^T, \quad k = 1, 2, 5, 7,$$
$$\Sigma_k = \gamma_k \beta_2 \Lambda_k \beta_2^T, \quad k = 3, 6,$$
$$\Sigma_k = \gamma_k \beta_3 \Lambda_k \beta_3^T, \quad k = 4\,.$$

Following (5), the estimation of the original covariance matrices involves maximizing (4) within $\mathscr{M}_{\boldsymbol{u}}$, the set of covariance matrices satisfying $\{\Sigma_k : u_k = g\} \in \mathscr{M}$ for all $g = 1, \dots, G$. Using the maximized log-likelihood as a measure for the appropriateness of $\boldsymbol{u}$, the optimal $\hat{\boldsymbol{u}}$ would provide a classification for $S_1, \dots, S_K$ according to the level $\mathscr{M}$. Precise definitions will be provided in Sect. 2. We will present an iterative procedure to simultaneously compute the optimal classification and covariance matrix estimators through the modification of equation (5) given by

$$(\hat{\mathbf{u}}, \hat{\boldsymbol{\Sigma}}) = \underset{\boldsymbol{u}, \boldsymbol{\Sigma} \in \mathscr{M}_{\boldsymbol{u}}}{\operatorname{argmax}} \left(\sum_{g=1}^{G} \sum_{k : u_k = g} \log \Big(W_d(n_k S_k | n_k, \Sigma_k) \Big) \right). \tag{6}$$

Solving this equation will allow us to fit Gaussian Mixture Models with intermediate parsimonious levels, in which the common parameters of a parsimonious level will be shared within each of the G classes given by the vector of indexes $\hat{\boldsymbol{u}}$, but varying between the different classes. In the previous example, we obtain three classes of covariance matrices that share their principal directions within each class, resulting in a better interpretation of the final classification and allowing a considerable reduction of the number of parameters to be estimated. We will use these ideas for fitting Gaussian Mixture Models in discriminant analysis and cluster analysis. To avoid unboundedness of the objective function in the clustering framework, we will impose the determinant and shape constraints of García-Escudero et al. (2020), which are fully implemented in the MATLAB toolbox FSDA (Riani et al. 2012). We will analyze some examples where the proposed models result in less parameters and more interpretability fitting the data, being better suited when compared with the 14 parsimonious models. We point out that, as it is becoming usual in the literature, to carry out the comparisons between different models, we will use the Bayesian Information Criterion (BIC). This applies to all examples considered in the text. It has been noticed by many authors that BIC selection works properly in model based clustering, as well as in discriminant analysis. Fraley and Raftery (2002) includes a detailed justification for the use of BIC, based on previous references. A summary of the comparison of BIC with other techniques for model selection can also be found in Biernacki and Govaert (1999).

The paper is organized as follows. Section 2 approaches the problem of the parsimonious classification of covariance matrices given by equation (6), focusing on its computation for the most interesting restrictions in terms of dimensionality reduction and interpretability. Throughout, we will only work with models based on the parsimonious levels of proportionality (VEE) and common principal components (VVE), although the extension to other levels is straightforward. Section 3 applies the previous theory for the estimation of Gaussian Mixture Models in cluster analysis and discriminant analysis, including some simulation examples for their illustration. Section 4 includes real data examples, where we will see the gain in interpretability that can arise from these solutions. Some conclusions are outlined in Sect. 5. Finally, Appendix A includes theoretical results, Appendix B provides some additional simulation examples and Appendix C explains technical details about the algorithms. Additional graphical material is provided in the Online Supplementary Figures document.

2 Parsimonious classification of covariance matrices

Given $n_1, \ldots, n_K$ independent observations from K groups with different distributions, and $S_1, \ldots, S_K$ the sample covariance matrices, a group classification may be provided according to different similarity criteria. In the general case, given a similarity criterion f depending on the sample covariance matrices and the sample lengths, the problem of classifying K covariance matrices in G classes, $1 \le G \le K$, typically would consist in solving the equation

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u}\in\mathscr{H}}{\operatorname{argmax}} \sum_{g=1}^{G} f\Big(\big\{(S_k, n_k) : u_k = g\big\}\Big),$$

where $\mathscr{H} = \big\{\boldsymbol{u} = (u_1, \ldots, u_K) \in \{1, \ldots, G\}^K : \forall\, g = 1, \ldots, G \quad \exists\, k \text{ verifying } u_k = g\big\}$. In this work, we focus on the Gaussian case, proposing different similarity criteria based on the parsimonious levels that arise from the spectral decomposition of a covariance matrix.

Multivariate procedures based on parsimonious decompositions assume that the theoretical covariance matrices $\Sigma_1, \ldots, \Sigma_K$ jointly verify one level $\mathscr{M}$ out of the fourteen in Table 1. To elaborate on this idea, we include now some useful notation. In a parsimonious model $\mathscr{M}$, we write $(\Sigma_1, \ldots, \Sigma_K) \in \mathscr{M}$ if these matrices share some common parameters C, and they have variable parameters $\boldsymbol{V} = (V_1, \ldots, V_K)$ (specified in the model $\mathscr{M}$). We will denote by $\Sigma(V_k, C)$ the covariance matrix with the size, shape and orientation parameters associated to (V_k, C). Therefore, under the parsimonious level $\mathscr{M}$, we are assuming that

$$\Sigma_k = \Sigma(V_k, C) \qquad k = 1, \ldots, K.$$

If the n_k observations of group k are independent and arise from a distribution $N(\mu_k, \Sigma_k)$, according to the arguments in the introduction, it is natural to consider the maximized log-likelihood (5) under the parsimonious level $\mathscr{M}$ as a similarity criterion for the covariance matrices. This allows us to measure their resemblance in the features associated to the common part of the decomposition in the theoretical model. Thus, the similarity criterion for the parsimonious level $\mathscr{M}$ is

$$f_{\mathscr{M}}\Big(\big\{(S_k, n_k), k = 1, \ldots, r\big\}\Big) = \max_{V_1,\ldots,V_r,C} \sum_{k=1}^{r} \log\Big(W_d\big(n_k S_k | n_k, \Sigma(V_k, C)\big)\Big).$$

Consequently, given a level of parsimony $\mathscr{M}$, the covariance matrix classification problem in G classes consists in solving the equation

$$\begin{aligned}\hat{\boldsymbol{u}} &= \underset{\boldsymbol{u}\in\mathscr{H}}{\operatorname{argmax}} \sum_{g=1}^{G} f_{\mathscr{M}}\Big(\big\{(S_k, n_k) : u_k = g\big\}\Big) \\ &= \underset{\boldsymbol{u}\in\mathscr{H}}{\operatorname{argmax}}\Bigg(\max_{V_1,\ldots,V_K,C_1,\ldots,C_G} \sum_{g=1}^{G} \\ &\sum_{k:u_k=g} \log\Big(W_d\big(n_k S_k | n_k, \Sigma(V_k, C_g)\big)\Big)\Bigg). \end{aligned} \tag{7}$$

In order to avoid the combinatorial problem of maximizing within $\mathscr{H}$, denoting the variable parameters by $\boldsymbol{V} = (V_1, \ldots, V_K)$ and the common parameters by $\boldsymbol{C} = (C_1, \ldots, C_G)$, we focus on the problem of maximizing

$$W(\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}) = \sum_{g=1}^{G} \sum_{k:u_k=g} \log\Big(W_d\Big(n_k S_k | n_k, \Sigma(V_k, C_g)\Big)\Big),$$

since the value $\boldsymbol{u}$ maximizing this function agrees with the optimal $\hat{\boldsymbol{u}}$ in (7). This problem will be referred to as **Classification *G*-$\boldsymbol{\mathscr{M}}$**. From the expression of the d-dimensional Wishart density, we can see that maximizing W is equivalent to minimizing with respect to the same parameters the function

$$\sum_{g=1}^{G} \sum_{k:u_k=g} n_k\Big(\log\big(|\Sigma(V_k, C_g)|\big) + \operatorname{tr}\Big(\Sigma(V_k, C_g)^{-1} S_k\Big)\Big).$$

Maximization can be achieved through a simple modification of the CEM algorithm (Classification Expectation Maximization, introduced in Celeux and Govaert (1992)), for any of the fourteen parsimonious levels. A sketch of the algorithm is presented here:

Classification *G*-$\boldsymbol{\mathscr{M}}$: Starting from an initial estimation $\boldsymbol{C^0} = (C_1^0, \ldots, C_G^0)$ of the common parameters, which may be taken as the parameters of G different matrices S_k randomly chosen between $S_1, \ldots, S_K$, the m^{th} iteration consists of the following steps:

- **u-V step**: Given the common parameters $\boldsymbol{C^m} = (C_1^m, \ldots, C_G^m)$, we maximize with respect to the partition $\boldsymbol{u}$ and the variable parameters $\boldsymbol{V}$. For each $k = 1, \ldots, K$, we compute

  $$\tilde{V}_{k,g} = \operatorname*{argmax}_V W_d\Big(n_k S_k \big| n_k, \Sigma(V, C_g)\Big)$$

  for $1 \leq g \leq G$, and we define:

  $$u_k^{m+1} = \operatorname*{argmax}_{g \in \{1,\ldots,G\}} W_d\Big(n_k S_k \big| n_k, \Sigma(\tilde{V}_{k,g}, C_g)\Big).$$

- **V-C step:** Given the partition $\boldsymbol{u^{m+1}}$, we compute the values $(\boldsymbol{V^{m+1}}, \boldsymbol{C^{m+1}})$ maximizing $W(\boldsymbol{u^{m+1}}, \boldsymbol{V}, \boldsymbol{C})$. The maximization can be done individually for each of the groups created, by maximizing for each $g = 1, \ldots, G$ the function

  $$(\{V_k\}_{k:u_k=g}, C_g) \longmapsto \sum_{k:u_k=g} \log\Big(W_d\big(n_k S_k | n_k, \Sigma(V_k, C_g)\big)\Big),$$

  The maximization for each of the 14 parsimonious levels can be done, for instance, with the techniques in Celeux and Govaert (Celeux and Govaert 1995). The methodology proposed therein for common orientation models uses modifications of the Flury algorithm (Flury and Gautschi 1986). However, for these models we will use the algorithms subsequently developed by Browne and McNicholas (2014a, b), often implemented the software available for parsimonious model fitting, which allow more efficient estimation of the common orientation parameters.

For each of the fourteen parsimonious models, the variable parameters in the solution $\hat{\boldsymbol{V}}$ may be computed as a function of the parameters $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{C}})$, the sample covariance matrices $S_1, \ldots, S_K$ and the sample lengths $n_1, \ldots, n_K$. Therefore, the function W could be written as $W(\boldsymbol{u}, \boldsymbol{C})$, and the maximization could be seen as a particular case of the coordinate descent algorithm explained in Bezdek et al. (1987).

As it was already noted, we focus on the development of the algorithm only for two particular (the most interesting) parsimonious levels. First of all, we are going to keep models flexible enough to enable the solution of (6), when taking $G = K$ (no grouping is assumed), to coincide with the unrestricted solution, $\hat{\Sigma}_k = S_k$. The first six models do not verify this condition. For the last eight models, the numbers of parameters are

$$\delta_{\text{VOL}} \cdot 1 + \delta_{\text{SHAPE}} \cdot (d-1) + \delta_{\text{ORIENT}} \cdot \frac{d(d-1)}{2}$$

where δ_{VOL}, δ_{SHAPE} and δ_{ORIENT} take the value 1 if the given parameter is assumed to be common, and K if it is assumed to be variable between groups. When d and K are large, the main source of variation in the number of parameters is related to considering common or variable orientation, followed by considering common or variable shape. For example, if $d = 9, k = 6$, the number of parameters related to each constraint are detailed in Table 2.

Table 2 Number of parameters associated with each feature when $k = 6, d = 9$

	Size	Shape	Orientation
Common	1	8	36
Variable	6	48	216

Our primary motivation is exemplified through Table 2: to raise alternatives for the models with variable orientation. For that, we look for models with orientation varying in G classes, with $1 \leq G \leq K$. We consider the case where size and shape are variable across all groups (G different Common Principal Components, G-CPC) and also the case where shape parameters are additionally common within each of the G classes (proportionality to G different matrices, G-PROP). Apart from the parameter reduction, these models can provide an easier interpretation of the variables involved in the problem, which is often a hard task in multidimensional problems with several groups. We keep the size variable, since it does not cause a major increase in the number of parameters, and it is easy to interpret. Therefore, the models we are considering are:

- **Classification G-CPC**: We are looking for G orthogonal matrices $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_G)$ and a vector of indexes $\boldsymbol{u} = (u_1, \ldots, u_K) \in \mathscr{H}$ such that

  $$\Sigma_k = \gamma_k \beta_{u_k} \Lambda_k \beta_{u_k}^T \quad k = 1, \ldots, K$$

  where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_K)$ and $\boldsymbol{\Lambda} = (\Lambda_1, \ldots, \Lambda_K)$ are the variable size and shape parameters. The number of parameters is $K + K(d-1) + Gd(d-1)/2$. In the situation of Table 2, taking $G = 2$ the number of parameters is 126, while allowing for variable orientation it is 270. To solve (7), we have to find a vector of indexes $\hat{\boldsymbol{u}}$, G orthogonal matrices $\hat{\boldsymbol{\beta}}$ and variable parameters $\hat{\boldsymbol{\gamma}}$ and $\hat{\boldsymbol{\Lambda}}$ minimizing

  $$(\boldsymbol{u}, \boldsymbol{\Lambda}, \boldsymbol{\gamma}, \boldsymbol{\beta}) \longmapsto \sum_{g=1}^{G} \sum_{k:u_k=g} n_k \left(d \log(\gamma_k) + \frac{1}{\gamma_k} \operatorname{tr}\left(\Lambda_k^{-1} \beta_g^T S_k \beta_g\right)\right). \tag{8}$$

- **Classification G-PROP**: We are looking for G orthogonal matrices $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_G)$, G shape matrices $\boldsymbol{\Lambda} = (\Lambda_1, \ldots, \Lambda_G)$ and $\boldsymbol{u} = (u_1, \ldots, u_K) \in \mathscr{H}$ such that

  $$\Sigma_k = \gamma_k \beta_{u_k} \Lambda_{u_k} \beta_{u_k}^T \quad k = 1, \ldots, K$$

  where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_K)$ are the variable size parameters. The number of parameters is $K + G(d-1) + Gd(d-1)/2$. In the situation of Table 2, the number of parameters if we take $G = 2$ is 94. To solve (7), we have to find a vector of indexes $\hat{\boldsymbol{u}}$, G orthogonal matrices $\hat{\boldsymbol{\beta}}$, G shape matrices $\hat{\boldsymbol{\Lambda}}$ and the variable size parameters $\hat{\boldsymbol{\gamma}}$ minimizing

  $$(\boldsymbol{u}, \boldsymbol{\Lambda}, \boldsymbol{\gamma}, \boldsymbol{\beta}) \longmapsto \sum_{g=1}^{G} \sum_{k:u_k=g} n_k \left(d \log(\gamma_k) + \frac{1}{\gamma_k} \operatorname{tr}\left(\Lambda_g^{-1} \beta_g^T S_k \beta_g \right) \right). \tag{9}$$

Explicit algorithms for finding the minimum of (8) and (9) are given in Section C.2 in the Appendix. The results given by both algorithms are illustrated in the following example, where we have randomly created 100 covariance matrices $\Sigma_1, \ldots, \Sigma_{100}$ according:

$$\Sigma_k = X \Big(\mathrm{U}(\alpha)\, \mathrm{Diag}(1, Y)\, \mathrm{U}(\alpha)^T \Big) \quad k = 1, \ldots, 100$$

where $\mathrm{U}(\alpha)$ represents the rotation of angle α, $\mathrm{Diag}(1, Y)$ is the diagonal matrix with entries $1, Y$, and X, Y, α are uniformly distributed random variables with distributions:

$$X \sim U\big(0.5, 2\big) \;\; Y \sim U\big(0, 0.5\big) \;\; \alpha \sim U\big(0, \pi\big).$$

For each $k = 1, \ldots, 100$, we have taken S_k as the sample covariance matrix computed from 200 independent observations from a distribution $N(0, \Sigma_k)$, and we have applied 4-CPC and 4-PROP to obtain different classifications of $S_1, \ldots, S_{100}$. The partitions obtained by both methods allow us to classify the covariance matrices according to both criteria. Figure 1 shows the 95% confident ellipses representing the sample covariance matrices associated to each class (coloured lines) together with the estimations of the common axes or the common proportional matrix within each class (black lines).

3 Gaussian mixture models

In a Gaussian Mixture Model (GMM), data are assumed to be generated by a random vector with probability density function:

$$f(y) = \sum_{k=1}^{K} \pi_k \phi(y | \mu_k, \Sigma_k)$$

where $0 \leq \pi_k \leq 1$, $\sum_{k=1}^{K} \pi_k = 1$. The idea of introducing covariance matrix restrictions given by parsimonious decomposition in the estimation of GMMs has become a common tool for statisticians, and methods are implemented in the software R in many packages. In this paper we use for the comparison the results given by the package *mclust* (Fraley and Raftery 2002; Scrucca et al. 2016), although there exists many others widely known ($Rmixmod$: Lebret et al. (2015); $mixtools$: Benaglia et al. (2009)). The aim of this section is to explore how we can fit GMMs in different contexts with the intermediate parsimonious models explained in Sect. 2, allowing the common part of the covariance matrices in the decomposition to vary between G classes. That is, with the same notation as in Sect. 2, we want to study GMMs with density function

$$f(y) = \sum_{g=1}^{G} \sum_{k:u_k=g} \pi_k \phi\Big(y \big| \mu_k, \Sigma(V_k, C_g) \Big) \tag{10}$$

where $\boldsymbol{u} = (u_1, \ldots, u_K) \in \mathscr{H}$ is a fixed vector of indexes, $\boldsymbol{V} = (V_1, \ldots, V_K)$ are the variable parameters, $\boldsymbol{C} = (C_1, \ldots, C_G)$ are the common parameters among classes and $\Sigma(V_k, C_g)$ is the covariance matrix with the parameters given by (V_k, C_g). The following subsections exploit the potential of these particular GMMs for cluster analysis and discriminant analysis. A more general situation where only part of the labels are known could also be considered, following the same line as in Dean et al. (2006), but it will not be discussed in this work.

As already noted in the Introduction, the criterion we are going to use for model selection between all the estimated models is BIC (Bayesian Information Criterion), choosing the model with a higher value of the BIC approximation given by

$$\mathrm{BIC} = 2 \cdot \mathrm{loglikelihood} - \log(N) \cdot p$$

where N is the number of observations and p is the number of independent parameters to be estimated in the model. This criterion is used for the comparison of the intermediate models G-CPC and G-PROP with the fourteen parsimonious models estimated in the software R with the functions in the *mclust* package. In addition, within the framework of discriminant analysis, the quality of the classification given by the best models, in terms of BIC, is also compared using cross validation techniques.

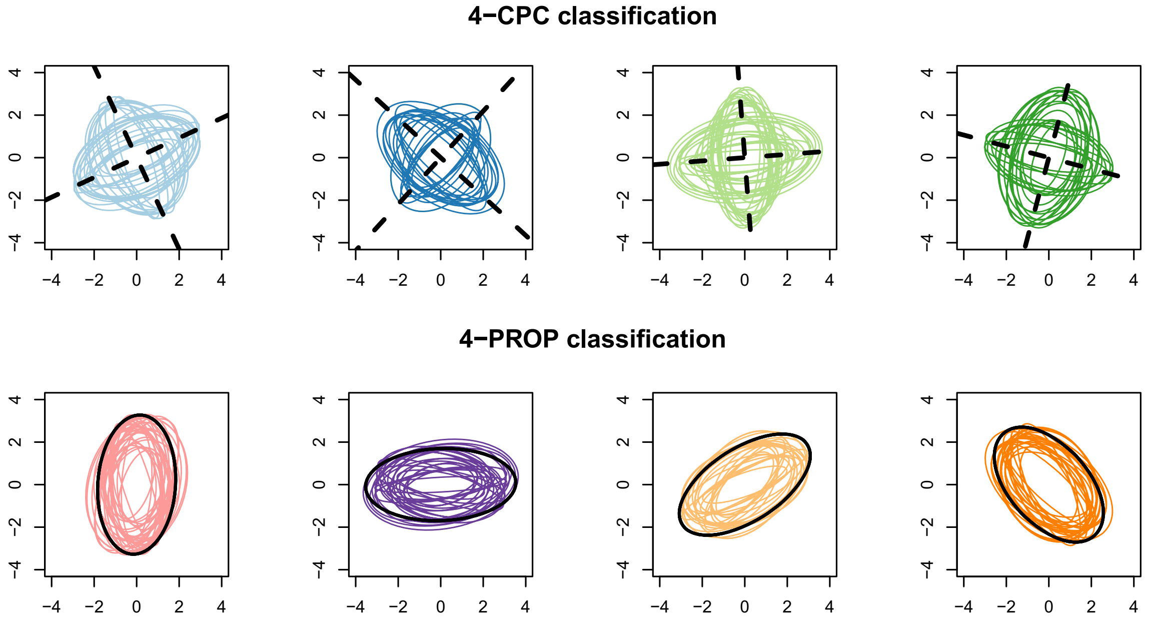

Fig. 1 Classification of $S_1, \dots, S_{100}$, represented by their 95% confidence ellipses. The first row shows the classes and axes estimations given by the 4-CPC model, and the second row shows the classes and proportional matrix estimations given by the 4-PROP model

3.1 Model-based clustering

Given $y_1, \dots, y_N$ independent observations of a d-dimensional random vector, clustering methods based on fitting a GMM with K groups seek to maximize the log-likelihood function (2). From the fourteen possible restrictions considered in Celeux and Govaert (1995), we can compute fourteen different maximum likelihood solutions in which size, shape and orientation are common or not between the K covariance matrices. For a particular level $\mathscr{M}$ in Table 1, the fitting requires the maximization of the log-likelihood

$$L\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{V}, C \Big| y_1, \dots, y_N\Big) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \phi\Big(y_i \Big| \mu_k, \Sigma(V_k, C)\Big)\right),$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ are the weights, with $0 \leq \pi_k \leq 1$, $\sum_{k=1}^{K} \pi_k = 1$, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)$ the means, $\boldsymbol{V} = (V_1, \dots, V_K)$ the variable parameters and C the common parameters. Estimation under the parsimonious restriction is performed via the EM algorithm. In the GMM context, we can see the complete data as pairs (y_i, z_i), where z_i is an unobserved random vector such that $z_{i,k} = 1$ if the observation y_i comes from distribution k, and $z_{i,k} = 0$ otherwise.

With the ideas of Sect. 2, we are going to fit Gaussian Mixture Models with parsimonious restrictions, but allowing the common parameters to vary between different classes. Assuming a parsimonious level of decomposition $\mathscr{M}$ and a number $G \in \{1, \dots, K\}$ of classes, we are supposing that our data are independent observations from a distribution with density function (10). The log-likelihood function given a fixed vector of indexes $\boldsymbol{u}$ is

$$L_{\boldsymbol{u}}\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{V}, \boldsymbol{C} \Big| y_1, \dots, y_N\Big) = \sum_{i=1}^{N} \log\left(\sum_{g=1}^{G} \sum_{k:u_k=g} \pi_k \phi\Big(y_i \Big| \mu_k, \Sigma(V_k, C_g)\Big)\right).$$

For each $\boldsymbol{u} \in \mathscr{H}$, we can fit a model. In order to choose the best value for the vector of indexes $\boldsymbol{u}$, we should compare the BIC values given by the different models estimated. As the number of parameters is the same, the best value for $\boldsymbol{u}$ can be obtained by taking

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u} \in \mathscr{H}}{\operatorname{argmax}} \left[\max_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{V}, \boldsymbol{C}} L_{\boldsymbol{u}}\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{V}, \boldsymbol{C} \Big| y_1, \dots, y_N\Big)\right].$$

In order to avoid the combinatorial problem of maximizing within $\mathscr{H}$, we can take $\boldsymbol{u}$ as if it were a parameter, and we are going to focus on the problem of maximizing

$$L\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C} \Big| y_1, \dots, y_N\Big) = \sum_{i=1}^{N} \log\left(\sum_{g=1}^{G} \sum_{k:u_k=g} \pi_k \phi\Big(y_i \Big| \mu_k, \Sigma(V_k, C_g)\Big)\right), \tag{11}$$

that will be referred to as **Clustering G-$\mathscr{M}$**. Therefore, given the unobserved variables $z_{i,k}$, for $k = 1, \dots, K$ and $i = 1, \dots, N$, the complete log-likelihood is

$$CL\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}\Big| y_1, \dots, y_N, z_{1,1}, \dots, z_{N,K}\Big) = \sum_{i=1}^{N} \left[\sum_{g=1}^{G} \sum_{k:u_k=g} z_{i,k} \log\Big(\pi_k \phi\Big(y_i \big| \mu_k, \Sigma(V_k, C_g)\Big)\Big) \right]. \tag{12}$$

The proposal of this section is to fit this model given a parsimonious level $\mathscr{M}$ and fixed values of K and $G \in \{1, \dots, K\}$, introducing also constraints to avoid the unboundedness of the log-likelihood function (11). For this purpose, we introduce the determinant and shape constraints studied in García-Escudero et al. (2020). For $k = 1, \dots, K$, denote by $(\lambda_{k,1}, \dots, \lambda_{k,d})$ the diagonal elements of the shape matrix Λ_k (which may be the same within classes). We impose K constraints controlling the shape of each group, in order to avoid solutions that are almost contained in a subspace of lower dimension, and a size constraint in order to avoid the presence of very small clusters. Given $c_{sh}, c_{vol} \geq 1$, we impose:

$$\frac{\max\limits_{l=1,\dots,d} \lambda_{k,l}}{\min\limits_{l=1,\dots,d} \lambda_{k,l}} \leq c_{sh}, \; k = 1, \dots, K, \qquad \frac{\max\limits_{k=1,\dots,K} \gamma_k}{\min\limits_{k=1,\dots,K} \gamma_k} \leq c_{vol} \tag{13}$$

Remark 1 With these restrictions, the theoretical problem of maximizing (11) is well defined. If Y is a random vector following a distribution $\mathbb{P}$, the problem consists in maximizing

$$\mathrm{E}\left[\log\Big(\sum_{g=1}^{G} \sum_{k:u_k=g} \pi_k \phi\Big(Y \big| \mu_k, \Sigma\big(V_k, C_g\big)\Big)\Big)\right] = \int \log\Big(\sum_{g=1}^{G} \sum_{k:u_k=g} \pi_k \phi\Big(y \big| \mu_k, \Sigma\big(V_k, C_g\big)\Big)\Big) \,\mathrm{d}\mathbb{P}(y) \tag{14}$$

with respect to $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}$, defined as above, and verifying (13). If $\mathbb{P}_N$ stands for the empirical measure $\mathbb{P}_N = (1/N)\sum_{i=1}^{N} \delta_{\{y_i\}}$, by replacing $\mathbb{P}$ by $\mathbb{P}_N$, we recover the original sample problem of maximizing (11) under the determinant and shape constraints (13). This approach guarantees that the objective function is bounded, allowing results to be stated in terms of existence and consistence of the solutions (see Section A in the Appendix).

Now, we are going to give a sketch of the EM algorithm used for the estimation of these intermediate parsimonious clustering models, for each of the fourteen levels.

Clustering G-$\mathscr{M}$: Starting from an initial solution of the parameters $\boldsymbol{\pi^0}, \boldsymbol{\mu^0}, \boldsymbol{u^0}, \boldsymbol{V^0}, \boldsymbol{C^0}$, we have to repeat the following steps until convergence:

- **E step**: Given the current values of the parameters $\boldsymbol{\pi^m}, \boldsymbol{\mu^m}, \boldsymbol{u^m}, \boldsymbol{V^m}, \boldsymbol{C^m}$, we compute the posterior probabilities

$$z_{i,k} = \frac{\pi_k^m \phi\Big(y_i | \mu_k^m, \Sigma\big(V_k^m, C_{u_k}^m\big)\Big)}{\sum_{l=1}^{K} \pi_l^m \phi\Big(y_i | \mu_l^m, \Sigma\big(V_l^m, C_{ul}^m\big)\Big)} \tag{15}$$

  for $k = 1, \dots, K, \; i = 1, \dots, N$.
- **M step**: In this step, we have to maximize (12) given the expected values $\{z_{i,k}\}_{i,k}$. The optimal values for $\boldsymbol{\pi^{m+1}}, \boldsymbol{\mu^{m+1}}$ are given by (3). With these optimal values, if we denote $S_k = (1/n_k)\sum_{i=1}^{N} z_{i,k}(y_i - \mu_k^{m+1})(y_i - \mu_k^{m+1})^T$, then we have to find the values $\boldsymbol{u^{m+1}}, \boldsymbol{V^{m+1}}, \boldsymbol{C^{m+1}}$ verifying the determinant and shape constraints (13) maximizing

$$(\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}) \longmapsto CL\Big(\boldsymbol{\pi^{m+1}}, \boldsymbol{\mu^{m+1}}, \boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}\Big| y_1, \dots, y_N, z_{1,1}, \dots, z_{N,K}\Big).$$

  If we remove the determinant and shape constraints, the solution of this maximization coincides with the classification problem presented in Sect. 2 for the computed values of $n_1, \dots, n_K$ and $S_1, \dots, S_K$. A simple modification of that algorithm, computing on each step the optimal size and shape constrained parameters (instead of the unconstrained version) with the *optimal truncation* algorithm presented in García-Escudero et al. (2020) allows the maximization to be completed. Determinant and shape constraints can be incorporated in the algorithms together with the parsimonious constraints following the lines developed in García-Escudero et al. (2022).

As already noted in Sect. 2, we keep only the clustering models G-CPC and G-PROP, the most interesting in terms of parameter reduction and interpretability. For these models, explicit algorithms are explained in Section C.3 in the Appendix. Now, we are going to illustrate the results of the algorithms in two simulation experiments:

- **Clustering G-CPC**: In this example, we simulate $n = 100$ observations from each of 6 Gaussian distributions, with means $\mu_1, \dots, \mu_6$ and covariance matrices verifying

$$\Sigma_k = \gamma_k \beta_1 \Lambda_k \beta_1^T, \quad k = 1, 2, 3,$$
$$\Sigma_k = \gamma_k \beta_2 \Lambda_k \beta_2^T, \quad k = 4, 5, 6\,.$$

In Fig. 2, we can see in the first plot the 95 % confidence ellipses of the six theoretical Gaussian distributions together with the 100 independent observations simulated from these distributions. The second plot represents the clusters created by the maximum likelihood solution for the 2-CPC model, taking $c_{sh} = c_{vol} = 100$. The numbers labeling the ellipses represent the class of covariance matrices sharing the orientation. Finally, the third plot represents the best solution estimated by *mclust* for $K = 6$, corresponding to the parsimonious model VEV, with equal shape and variable size and orientation. The BIC value in the 2-CPC model (31 d.f.) is -3937.08, whereas the best model VEV (30 d.f.) estimated with *mclust* has BIC value -3960.07. Therefore, the GMM estimated with the 2-CPC restriction has higher BIC than all the parsimonious models. Finally, the number of observations assigned to different clusters from the original ones is 82 for the 2-CPC model and 91 for the VEV model.

- **Clustering G-PROP**: In this example, we simulate $n = 100$ observations from each of 6 Gaussian distributions, with means $\mu_1, \dots, \mu_6$ and covariance matrices verifying:

$$\Sigma_k = \gamma_k A_1, \quad k = 1, 2, 3,$$
$$\Sigma_k = \gamma_k A_2, \quad k = 4, 5, 6\ .$$

Figure 3 is analogous to Fig. 2, but in the proportionality case. The BIC value for the 2-PROP model (27 d.f.) with $c_{sh} = c_{vol} = 100$ is -3873.127, whereas the BIC value for the best model fitted by *mclust* is -3919.796, which corresponds to the unrestricted model VVV (35 d.f.). Now, the number of observations wrongly assigned to the source groups is 64 for the 2-PROP model, while it is 71 for the VVV model.

Remark 2 Note that, by imposing appropriate constraints in the clustering problem, we can significantly decrease the number of parameters while keeping a good fit of the data. Figure 3 shows this effect. However, constraints also have a clear interpretation in cluster analysis problems, since we are looking for groups that are forced to have a particular shape. Therefore, different constraints can lead to clusters with different shapes. This is what happens in Fig. 2, where by introducing the right constraints we have managed to make the clusters created more similar to the original ones. Of course, in the absence of prior information, it is not possible to know the appropriate constraints, and the most reasonable approach is to select a model according to a criterion that penalizes the fit with the number of parameters such as the BIC.

Table 3 Proportions of times in which clustering 2-CPC or 2-PROP model improves the best *mclust* model in terms of BIC, for each sample size *n*

Example	n = 50	n = 100	n = 200
2-CPC	0.570	0.927	1.000
2-PROP	0.933	0.999	1.000

To evaluate the sensitivity of BIC for the detection of the true underlying model, we have used the models described in the two previous examples. Once a model and a particular sample size n (=50, 100, 200) have been chosen, the simulation planning produces a sample containing n random elements generated from each $N(\mu_k, \Sigma_k),\ k = 1, \dots, 6$. We repeated every simulation plan 1000 times, comparing for every sample the BIC obtained for the underlying clustering model vs the best parsimonious model estimated by *mclust*. Table 3 includes the proportions of times in which 2-CPC or 2-PROP model improves the best *mclust* model in terms of BIC for each value of n. Of course, the accuracy of the approach should depend on the dimension, the number of groups, the overlapping... However, even in the case of a large overlapping, as in the present examples, the proportions reported in Table 3 show that moderate values of n suffice to get very high proportions of success. Appendix B contains additional simulations supporting the suitability of BIC in this framework.

3.2 Discriminant analysis

The parsimonious model introduced in Bensmail and Celeux (1996) for discriminant analysis has been developed in conjunction with model-based clustering. The *R* package *mclust* (Fraley and Raftery 2002; Scrucca et al. 2016) also includes functions for fitting these models, denoted by EDDA (Eigenvalue Decomposition Discriminant Analysis). In this context, given a parsimonious level $\mathscr{M}$ and a number G of classes, we can also consider fitting an intermediate model for each fixed $\boldsymbol{u} \in \mathscr{H}$, by maximizing the complete log-likelihood

$$CL_{\boldsymbol{u}}\Big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{V}, \boldsymbol{C} \Big| y_1, \dots, y_N, z_{1,1}, \dots, z_{N,K}\Big) = \sum_{i=1}^{N} \left[\sum_{g=1}^{G} \sum_{k:u_k=g} z_{i,k} \log\Big(\pi_k \phi\Big(y_i | \mu_k, \Sigma(V_k, C_g)\Big)\Big) \right] . \tag{16}$$

Model comparison is done through BIC, and consequently we could try to choose $\boldsymbol{u}$ maximizing the log-likelihood (11). However, given that in the model fitting we are maximizing the complete log-likelihood (16), it is not unreasonable trying to find the value of $\boldsymbol{u}$ maximizing (16). Proceeding in this manner, we can think of $\boldsymbol{u}$ as a parameter, and the problem

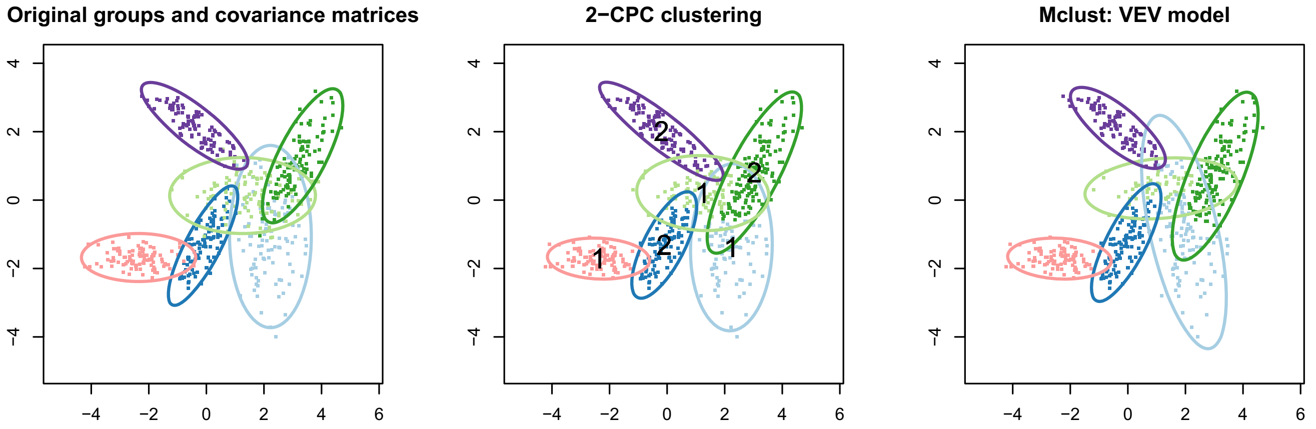

Fig. 2 From left to right: 1. Theoretical Gaussian distributions and observations simulated from each distribution. 2. Solution estimated by clustering through 2-CPC model. 3. Best clustering solution estimated by *mclust* in terms of BIC

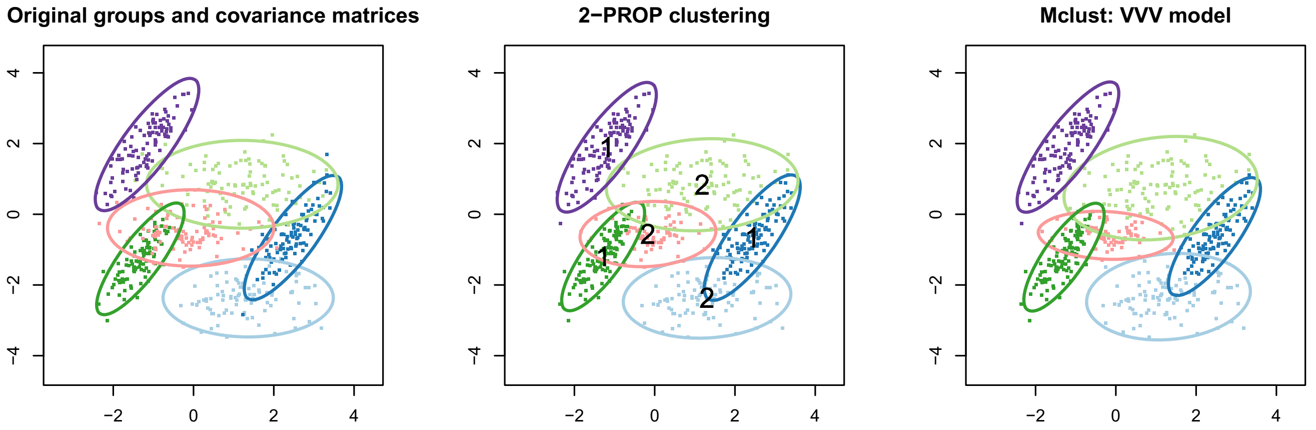

Fig. 3 From left to right: 1. Theoretical Gaussian distributions and observations simulated from each distribution. 2. Solution estimated by clustering through 2-PROP model. 3. Best clustering solution estimated by *mclust* in terms of BIC

consists in maximizing (12). Model estimation is simple from model-based clustering algorithms: with a single iteration of the M step, we can compute the values of the parameters. A new set of observations can be classified computing the posterior probabilities, with the formula (15) of the E step, and assigning each new observation to the group with higher posterior probability. Since the groups are known, the complete log-likelihood (12) is bounded under mild conditions, and it is not required to impose eigenvalue constraints, although it may be interesting in some examples with almost degenerated variables. To summarize the quality of the classification given by the best models (selected through BIC) in the different examples, other indicators based directly on classification errors are provided:

- **MM**: Model Misclassification, or training error. Proportion of observations misclassified by the model fitted with all observations.
- **LOO**: Leave One Out error.
- **CV(R,p):** Cross Validation error. Considering each observation as labeled or unlabeled with probability p and $1-p$, we compute the proportion of unlabeled observations misclassified by the model fitted with the labeled observations. The indicator CV(R,p) represents the mean of the proportions obtained in R repetitions of the process. When several classification methods are compared, the same R random partitions are used to compute the values of this indicator.

In the line of the previous section, only the discriminant analysis models G-CPC and G-PROP are considered. Table 4 and 5 show the results of applying these models to the simulation examples of Figs. 2, 3. In both situations, the classification obtained with our model slightly improves that given by *mclust*.

As we did in the clustering setting, in order to evaluate the sensitivity of BIC for the detection of the true underlying model, simulations have been repeated 1000 times, for each sample size n (=30, 50, 100, 200). Table 6 shows the propor-

Table 4 Classification results for data in Fig. 2 for the best *mclust* model and 2-CPC

Model	Loglik	df	BIC	MM	LOO	CV(300,0.9)
mclust: VVV	−1874.865	30	−3941.637	66/600	71/600	0.1187
2-CPC	−1874.74	26	−**3915.801**	65/600	69/600	0.1161

Table 5 Classification results for data in Fig. 3 for the best *mclust* model and 2-PROP

Model	Loglik	df	BIC	MM	LOO	CV(300,0.9)
mclust: VVV	−1852.765	30	−3897.439	62/600	69/600	0.1102
2-PROP	−1853.056	22	−**3846.845**	64/600	68/600	0.1083

Table 6 Proportions of times in which discriminant analysis 2-CPC or 2-PROP model improves the best *mclust* model in terms of BIC, for each sample size n

Example	n = 30	n = 50	n = 100	n = 200
2-CPC	0.443	0.782	0.975	1.000
2-PROP	0.971	1.000	1.000	1.000

tions of times in which 2-CPC or 2-PROP model improves the best *mclust* model in terms of BIC for each value of n.

Remark 3 In discriminant analysis, the weights $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ might not be considered as parameters. Model-based methods assume that observations from the k^{th} group follow a distribution with density function $f(\cdot, \theta_k)$. If π_k is the proportion of observations of group k, the classifier minimizing the expected misclassification rate is known as Bayes classifier, and it assigns an observation y to the group with higher posterior probability

$$P\big(y \in \text{Group } k\big) = \frac{\pi_k f(y, \theta_k)}{\sum_{l=1}^{K} \pi_l f(y, \theta_l)}. \tag{17}$$

The values of $\boldsymbol{\pi}, \theta_1, \ldots, \theta_K$ are usually unknown, and the classification is performed with estimations $\hat{\boldsymbol{\pi}}, \hat{\theta}_1, \ldots, \hat{\theta}_K$. Whereas $\hat{\theta}_1, \ldots, \hat{\theta}_K$ are always parameters estimated from the sample, the values of $\hat{\boldsymbol{\pi}}$ may be seen as part of the classification rule, if we think that they represent a characteristic of a particular sample we are classifying, or real parameters, if we assume that the observations (z_i, y_i) arise from a GMM such that

$$z_i \sim \text{mult}\Big(1, \{1, \ldots, K\}, \{\pi_1, \ldots, \pi_K\}\Big)$$
$$y_i \big| z_i \sim f\left(\cdot, \theta_{z_i}\right),$$

where $mult()$ denotes the multinomial distribution, and the weights verify $0 \le \pi_k \le 1$, $\sum_{k=1}^{K} \pi_k = 1$. In accordance with *mclust*, for model comparison we are not considering $\boldsymbol{\pi}$ as parameters, although its consideration would only mean adding a constant to all BIC values computed. However, in order to define the theoretical problem, the situation where we are considering $\boldsymbol{\pi}$ as a parameter is more interesting. If (Z, Y) is a random vector following a distribution $\mathbb{P}$ in $\{1, \ldots, K\} \times \mathbb{R}^d$, the theoretical problem consists in maximizing

$$\mathrm{E}\Bigg[\sum_{g=1}^{G} \sum_{k:u_k=g} \mathrm{I}(Z = k) \log\Big(\pi_k \phi\Big(Y | \mu_k, \Sigma(V_k, C_g)\Big)\Big)\Bigg] =$$
$$\int \sum_{g=1}^{G} \sum_{k:u_k=g} \mathrm{I}(z = k) \log\Big(\pi_k \phi\Big(y | \mu_k, \Sigma(V_k, C_g)\Big)\Big) \mathrm{d}\mathbb{P}(z, y) \tag{18}$$

with respect to the parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}$. Given N observations (z_i, y_i), $i = 1, \ldots, N$ of $\mathbb{P}$, the problem of maximizing (18) agrees with the sample problem presented above the remark when taking the empirical measure $\mathbb{P}_N$, with the obvious relation $z_{i,k} = \mathrm{I}(z_i = k)$. Arguments like those presented in Section A in the Appendix for the cluster analysis problem would give existence and consistency of solutions also in this setting.

4 Real data examples

To illustrate the usefulness of the G-CPC and G-PROP models in both settings, we show four real data examples in which our models outperform the best parsimonious models fitted by *mclust*, in terms of BIC. The two first examples are intended to illustrate the methods in simple and well-known data sets, while the latter involve greater complexity.

4.1 Cluster analysis: IRIS

Here we revisit the famous *Iris data set*, which consists of observations of four features (length and width of sepals and petals) of 50 samples of three species of Iris (setosa, versicolor and virginica), and is available in the base package of R. We apply the functions of package *mclust* for model-based clustering, letting the number of clusters to search equal to 3, to obtain the best parsimonious model in terms of BIC value. Table 7 compares this model with the models 2-CPC

Table 7 Iris data solutions for clustering with *mclust*, 2-CPC and 2-PROP

Model	Loglik	df	BIC	MM
mclust: VEV	−186.074	38	−562.550	5/150
2-CPC	−185.538	38	−561.480	5/150
2-PROP	−192.177	35	**−559.727**	4/150

and 2-PROP, fitted with $c_{sh} = c_{vol} = 100$. With some abuse of notation, we include in the table the Model Misclassification (MM), representing here the number of observations assigned to different clusters than the originals, after identifying the clusters created with the originals in a logical manner.

From Table 7 we can appreciate that the best clustering model in terms of BIC is the 2-PROP model. In Fig. 4 we can see the clusters created by this model. These clusters coincide with the real groups, except for four observations. From this example, we can also see the advantage of the intermediate models G-CPC and G-PROP in terms of interpretability. In the solution found with G-PROP the covariance matrices associated to two of the three clusters are proportional. Each cluster represents a group of individuals with similar features, which in absence of labels, we could see as a subclassification within the Iris specie. In this subclassification associated to the groups with proportional covariance matrices, both groups share not only the principal directions, but also the same proportion of variability between the directions. In many biological studies, principal components are of great importance. When working with phenotypic variables, principal components may be interpreted as “growing directions" (see e.g. Thorpe 1983). From the estimated model, we can conclude that in the Iris data, it is reasonable to think that there are three groups, two of them with similar “growing pattern", since not only the principal components are the same, but also the shape is common. However, this biological interpretation will become even more evident in the following example.

4.2 Discriminant analysis: CRABS

The data set consists of measures of 5 features over a set of 200 crabs from two species, orange and blue, and from both sexes, and it is available in the *R* package *MASS* (Venables and Ripley 2002). For each specie and sex (labeled OF, OM, BF, BM) there are 50 observations. The variables are measures in mm of the following features: frontal lobe (FL), rear width (RW), carapace length (CL), carapace width (CW) and body depth (BD). Applying the classification function of the *mclust* library, the best parsimonious model in terms of BIC is EEV. Table 8 shows the result for the EEV model, together with the discriminant analysis models 2-CPC and 2-PROP, with $c_{sh} = c_{vol} = 100000$ (with these values, the solutions agrees with the unrestricted solutions).

The results show that the comparison given by BIC can differ from those obtained by cross validation techniques, partially because BIC mainly measures the fit of the data to the model. However, in the parsimonious context, model selection is usually performed via BIC, in order to avoid the very time-consuming process of evaluating every possible model with cross validation techniques.

Figure 1 in the Online Supplementary Figures represents the solution estimated by 2-PROP model. The solution given by this model allows for a better biological interpretation than the one given by the parsimonious model EEV, where orientation varies along the 4 groups, making the comparison quite complex. In the 2-PROP model, the groups of males of both species share proportional matrices, and the same is true for the females. Returning to the biological interpretation of the previous example, under the 2-PROP model, we can state that crabs of the same sex have the same “growing pattern”, despite of being from different species.

4.3 Cluster analysis: gene expression cancer

In this example, we work with the *Gene expression cancer RNA-Seq Data Set*, which can be downloaded from the UCI Machine Learning Repository. This data set is part of the data collected by “The Cancer Genome Atlas Pan-Cancer analysis project"" (Weinstein et al. 2013). The considered data set consists of a random extraction of gene expressions of patients having different types of tumor: BRCA (breast carcinoma), KIRC (kidney renal clear-cell carcinoma), COAD (colon adenocarcinoma), LUAD (lung squamous carcinoma) and PRAD (prostate adenocarcinoma). In total, the data set contains the information of 801 patients, and for each patient we have information of 20531 variables, which are the RNA sequencing values of 20531 genes. To reduce the dimensionality and to apply model-based clustering algorithms, we have removed the genes with almost zero sum of squares ($< 10^{-5}$) and applied PCA to the remaining genes. We have taken the first 14 principal components, the minimum number of components retaining more than 50 % of the total variance. Applying model-based clustering methods looking for 5 groups to this reduced data set, we have found that 3-CPC, fitted with $c_{sh} = c_{vol} = 1000$, improves the BIC value obtained by the best parsimonious model estimated by *mclust*. The results obtained from 3-CPC, presented in Table 9, significantly improve the assignment error made by *mclust*. Figure 2 in the Online Supplementary Figures shows the projection of the solution obtained by 3-CPC onto the first six principal components computed in the preprocessing steps.

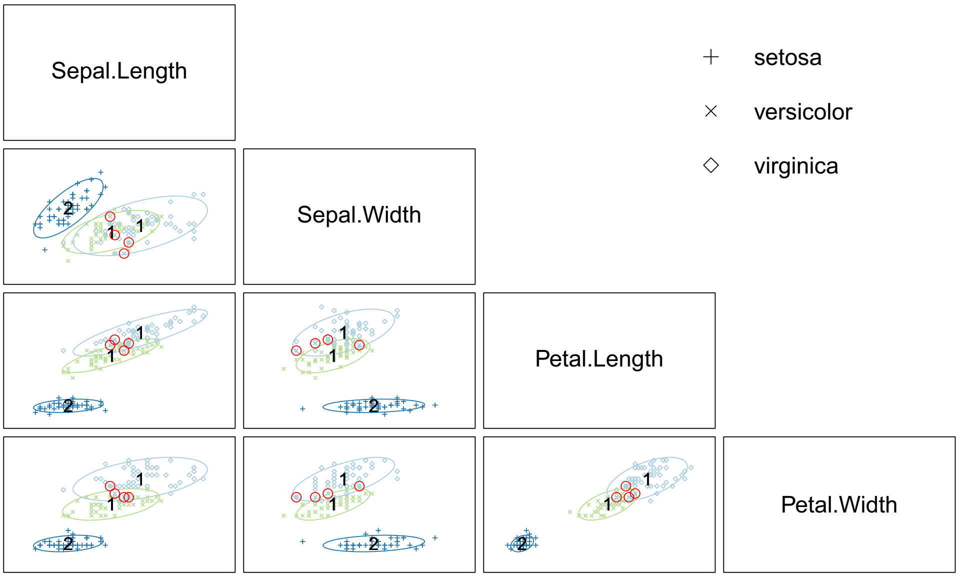

Fig. 4 Clustering obtained from 2-PROP model in the Iris data set. Color represents the clusters created. The ellipses are the contours of the estimated mixture densities, grouped into the classes given by indexes in black. Point shapes represent the original groups. Observations lying on different clusters from the originals are marked with red circles

Table 8 Crabs data solutions for discriminant analysis with *mclust*, 2-CPC and 2-PROP

Model	Loglik	df	BIC	MM	LOO	CV(300,0.8)	CV(300,0.95)
mclust: EEV	−1247.693	65	−2839.776	8/200	9/200	0.0513	0.0521
2-CPC	−1271.470	60	−2860.839	7/200	9/200	0.0536	0.0514
2-PROP	−1278.906	52	**−2833.324**	8/200	11/200	0.0546	0.0613

Table 9 Cancer data solutions for clustering with *mclust* and 3-CPC

Model	Loglik	df	BIC	MM
mclust: VVV	−44121.24	599	−92247.32	64/801
3-CPC	−44561.12	417	**−91910.25**	6/801

4.4 Discriminant analysis: Italian olive oil

The data set contains information about the composition in percentage of eight fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic and eicosenoic) found in the lipid fraction of 572 Italian olive oils, and it is available in the *R* package *pdfCluster* (Azzalini and Menardi 2014). The olive oils are labeled according to a two level classification: 9 different areas that are grouped at the same time in three different regions.

- SOUTH: Apulia North, Calabria, Apulia South, Sicily.
- SARDINIA: Sardinia inland, Sardinia coast.
- CENTRE-NORTH: Umbria, Liguria east, Liguria west.

In this example, we have evaluated the performance of different discriminant analysis models, for the problem of classifying the olive oils between areas. The best parsimonious model fitted with *mclust* is the VVE model, with variable size and shape and equal orientation. Note that due to the dimension $d = 8$, there is a significant difference in the number of parameters between models with common or variable orientation. Therefore, BIC selection will tend to choose models with common orientation, despite the fact that this hypothesis might not be very precise. This suggests that intermediate models could be of great interest also in this example. Given that the last variable *eicosenoic* is almost degenerated in some areas, we fit the models with $c_{sh} = c_{vol} = 10000$, and the shape constraints are effective

Table 10 Olive oil discriminant analysis with *mclust*, 2-CPC, 3-CPC and 3-PROP

Model	Loglik	df	BIC	MM	LOO	CV(300,0.8)	CV(300,0.95)
mclust: VVE	−20595.49	172	−42283.03	12/572	20/572	0.0375	0.0363
2-CPC	−20452.64	200	−42175.11	10/572	18/572	0.0369	0.0281
3-CPC	−20332.93	228	**−42113.47**	9/572	16/572	0.0365	0.0278
3-PROP	−20521.33	186	−42223.60	16/172	27/572	0.0464	0.0463

in some groups. We have found 3 different intermediate models improving the BIC value obtained with *mclust*. Results are displayed in Table 10.

The best solution found in terms of BIC is given by the 3-CPC model, which is also the solution with the best values for the other indicators. The classification of the areas in classes given in this solution is:

- CLASS 1: Umbria.
- CLASS 2: Apulia North, Calabria, Apulia South, Sicily.
- CLASS 3: Sardinia inland, Sardinia coast, Liguria east, Liguria west.

Note that areas in class 2 exactly agree with areas from the South Region. This classification coincides with the separation in classes given by 3-PROP, whereas 2-PROP model grouped together class 1 and class 3. These facts support that our intermediate models have been able to take advantage of the apparent difference in the structure of the covariance matrices from the South region and the others. When we are looking for a three-class separation, instead of splitting the areas from the Centre-North and Sardinia into these two regions, all Centre-North and Sardinia areas are grouped together, except Umbria, which forms a group alone. Figure 3 in the Online Supplementary Figures represents the solution in the principal components of the group Umbria, and we can appreciate the characteristics of this area. The plot corresponding to the second and third variables allows us to see clear differences in some of its principal components. Additionally, we can see that it is also the area with less variability in many directions. In conclusion, a different behavior of the variability in the olive oils from this area seems to be clear. This could be related to the geographical situation of Umbria (the only non-insular and non-coastal area under consideration).

5 Conclusions and further directions

Cluster analysis of structured data opens up interesting research prospects. This fact is widely known and used in applications where the data themselves share some common structure, and thus clustering techniques are a key tool in functional data analysis. More recently, the underlying structures of the data have increased in complexity, leading, for example, to consider probability distributions as data, and to use innovative metrics, such as earth-mover or Wasserstein distances. This configuration has been used in cluster analysis, for example, in del Barrio et al. (2019), from a classical perspective, but also including new perspectives: meta-analysis of procedures, aggregation facilities.... Nevertheless, to the best of our knowledge, this is the first occasion in which a clustering procedure is used as a selection (of an intermediate model) step in an estimation problem. Our proposal allows improvements in the estimation process and, arguably, often a gain in the interpretability of the estimation thanks to the chosen framework: Classification through the Gaussian Mixture Model.

The presented methodology enhances the so-called parsimonious model leading to the inclusion of intermediate models. They are linked to geometrical considerations on the ellipsoids associated to the covariance matrices of the underlying populations that compose the mixture. These considerations are precisely the essence of the parsimonious model. The intermediate models arise from clustering covariance matrices, considered as structured data, and using a similarity measure based in the likelihood. The consideration of clustering these objects through other similarities could be appropriate looking for tools for different goals. In particular, we emphasize on the possibility of clustering based on metrics like the Bures–Wasserstein distance. The role played here by the BIC would have to be tested in the corresponding configurations or, alternatively, replaced by appropriate penalties for choosing between other hierarchical models.

Feasibility of the proposal is an essential requirement for a serious essay of a statistical tool. The algorithms considered in the paper are simple adaptations of Classification Expectation Maximization algorithm, but we think that they could be still improved. We will pursuit on this challenge, looking also for feasible computations for similarities associated to new pre-established objectives.

In summary, through the paper we have used clustering to explore similarities between groups according to predetermined patterns. In this wider setup, clustering is not a goal in itself, it can be an important tool for specialized analyses.

6 Supplementary material

Supplementary figures: Online document with additional graphs for the real data examples. Repository: Github repository containing the *R* scripts with the algorithms and workflow necessary to reproduce the results of this work. Simulation data of the examples are also included. (https://github.com/rvitores/ImprovingModelChoice).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-024-10410-y.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Appendix A Theoretical results

In this section we are going to further on the comments of Remark 1. Given a parsimonious model $\mathscr{M}$ and fixed values of K, G, c_{vol} and c_{sh}, the problem consists in maximizing the function (14) in $\Theta^{\mathscr{M},G}_{c_{vol},c_{sh}}$, the set of parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}$ associated with the clustering model G-$\mathscr{M}$ verifying the size and shape constraints (13). Using the same notation as in García-Escudero et al. (2020), denote

$$\Theta_{c_{vol},c_{sh}} = \Big\{(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \in [0,1]^K \times \mathbb{R}^{dK} \times (\mathbb{S}^d_{>0})^K \\ \text{verifying the constraints (13)}\}$$

where $\mathbb{S}^d_{>0}$ is the set of positive definite symmetric real matrices. If we define the map

$$\begin{array}{ccc} \Theta^{\mathscr{M},G}_{c_{vol},c_{sh}} & \xrightarrow{T} & [0,1]^K \times \mathbb{R}^{dK} \times (\mathbb{S}^d_{>0})^K \\ \big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}\big) & \longmapsto & \big(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\big(\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}\big)\big) \end{array}$$

where $\boldsymbol{\Sigma}\big(\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}\big)$ is the collection of K covariance matrices created from the parameters $\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{C}$, it is obvious that $T(\Theta^{\mathscr{M},G}_{c_{vol},c_{sh}}) \subset \Theta_{c_{vol},c_{sh}}$. This and Lemma 1 in García-Escudero et al. (2020) allow us to replicate the proofs of Proposition 1 and Proposition 2 in García-Escudero et al. (2015) to prove the following theorems on the existence and consistence of the solutions.

Theorem 1 *If $\mathbb{P}$ is a probability that is not concentrated on K points, and $\mathrm{E}_{\mathbb{P}} \|\cdot\|^2 < \infty$, the maximum of (14) is achieved at some $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{V}}, \hat{\boldsymbol{C}}) \in \Theta^{\mathscr{M},G}_{c_{vol},c_{sh}}$.*

Given $\{y_i\}_{i=1}^{\infty}$ independent observations of the distribution $\mathbb{P}$, for each N we can define the empirical distribution $\mathbb{P}_N = (1/N)\sum_{i=1}^N \delta_{\{y_i\}}$. The sample problem of maximizing (14) under the constraint (13) coincides with the distributional problem presented here, when we take the probability $\mathbb{P}_N$. Therefore, Theorem 1 also guarantees the existence of the solution of the empirical problem corresponding to large enough samples drawn from an absolutely continuous distribution.

We use the notation θ_0 for any constrained maximizer of the theoretical problem for the underlying distribution $\mathbb{P}$, and let

$$\theta_n = \big(\boldsymbol{\pi^n}, \boldsymbol{\mu^n}, \boldsymbol{u^n}, \boldsymbol{V^n}, \boldsymbol{C^n}\big)$$

be a sequence of empirical solutions for the sequence of empirical sample distributions $\{\mathbb{P}_N\}_{N=1}^{\infty}$. The following result states consistency under similar assumptions as in Theorem 1 if the maximizer of the theoretical problem is assumed to be unique.

Theorem 2 *Let us assume that $\mathbb{P}$ is not concentrated on K points, $\mathrm{E}_{\mathbb{P}} \|\cdot\|^2 < \infty$ and that $\theta_0 \in \Theta^{\mathscr{M},G}_{c_{vol},c_{sh}}$ is the unique constrained maximizer of (14) for $\mathbb{P}$. If $\{\theta_n\}_{n=1}^{\infty}$ is a sequence of empirical maximizers of (14) with $\theta_n \in \Theta^{\mathscr{M},G}_{c_{vol},c_{sh}}$, then $\theta_n \longrightarrow \theta_0$ almost surely.*

Appendix B Additional simulations

At the suggestion of a reviewer, we present two additional simulation examples that reinforce the ideas presented in Sect. 3.1. For the sake of brevity, we only give the results for the more involved clustering problem. We point out two basic ideas. Since we have introduced a broader family of models, model selection will be more challenging than within the fourteen parsimonious models. This is clearly seen in the former example, but with a sufficiently large sample size, BIC is still able to select the true model. In the latter example, we emphasize that our extension of the parsimonious model is not redundant.

First, we repeat the two-dimensional simulation experiment described in Sect. 3.1, but assuming the VVE model:

$$\Sigma_k = \gamma_k \beta \Lambda_k \beta^T, \qquad k = 1, \ldots, 6.$$

This example allows us to deal with two different situations. The true underlying model verifies the VVE (1-CPC) model, so it also verifies the 2-CPC model, but it does not verify the 2-PROP model. For a sample with $n = 50$ observations from each group, we compute the VVE, 2-CPC and 2-PROP solutions for clustering. Results are shown in Fig. 5, where we can appreciate that both VVE and 2-CPC models fit the data perfectly, while the constraint of 2-PROP does not allow a good fitting of the data. This is also reflected in Table 11, where the BIC values are computed. The best model in terms of BIC is VVE, but 2-CPC is also competitive. 2-PROP gives much worse BIC values.

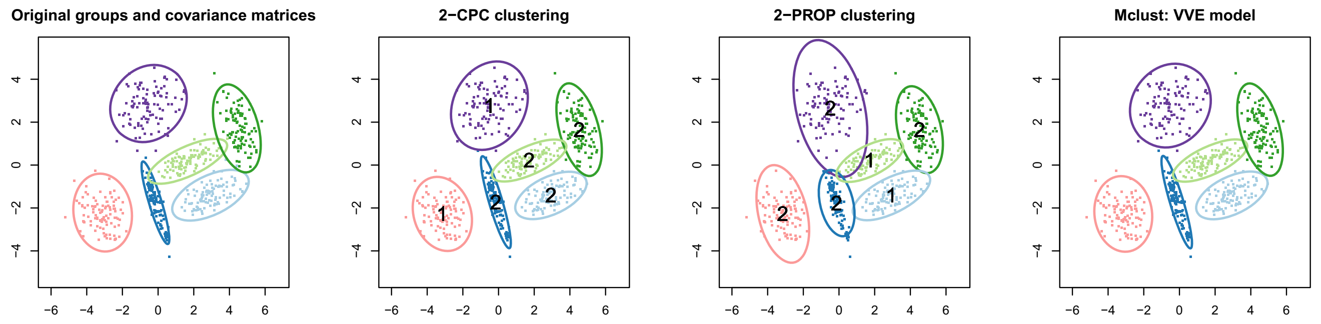

Fig. 5 From left to right: 1. Theoretical Gaussian distributions and observations simulated from each distribution in the VVE example. 2. Solution estimated by clustering through 2-CPC model. 3. Solution estimated by clustering through 2-PROP model. 4. VVE clustering solution estimated with *mclust*

Table 11 Clustering results in the VVE example for VVE, 2-CPC and 2-PROP models

model	loglik	df	BIC
mclust: VVE	−2116.478	30	**−4424.864**
2-CPC	−2115.219	31	−4428.742
2-PROP	−2212.407	27	−4597.531

Table 12 Proportions of times in which clustering 2-CPC or 2-PROP model improves the model VVE in terms of BIC, for each sample size n

Example	n = 50	n = 100	n = 500
2-CPC	0.208	0.141	0.031
2-PROP	0.001	0	0

Table 13 Clustering results in the 10-dimensional example for the best *mclust* model and 2-CPC

Model	Loglik	df	BIC
mclust: VVE	−7310.521	170	−15708.52
2-CPC	−6714.553	215	**−14804.45**

Finally, as we did in Table 3, simulations have been repeated 1000 times, for different sample sizes n. In each simulation, we are comparing the BIC value obtained for 2-CPC and 2-PROP with the BIC value obtained for the true underlying model VVE. Results are shown in Table 12.

The results are consistent with the ideas set out above. Since 2-PROP model is not verified, the clustering models fitted with this constraint give lower BIC value than VVE. 2-CPC model is verified, it is more flexible than VVE, and the difference in the number of parameters is only one. Thus, this is a rather complicated setting for model selection. Even in this case, if the sample size n is large enough, BIC is able to select the true model in almost all cases.

The second example is similar to the 2-CPC example in 3.1, but now in dimension $d = 10$. We consider $K = 6$ distributions, with $G = 2$ classes given by

$$\Sigma_k = \gamma_k \beta_1 \Lambda_k \beta_1^T, \quad k = 1, 2, 3,$$
$$\Sigma_k = \gamma_k \beta_2 \Lambda_k \beta_2^T, \quad k = 4, 5, 6\,.$$

Parameters were created so that we get a favorable but not trivial situation for applying clustering algorithms. Figure 4 in the Online Supplementary Figures shows a sample created with $n = 100$ observations from each group. For this sample, we fit the clustering model 2-CPC, and we compare it with the best model estimated by *mclust*. The results of this simulation are given in Table 13.

The main advantage of considering our intermediate models against the 14 parsimonious models estimated by *mclust* in this particular example is that *mclust* is selecting the model VVE, which it is not exactly verified, because the VVV model involves a substantially larger number of parameters (395 for clustering, 390 for discriminant analysis). This leads to a significant improvement in the BIC value of the 2-CPC model. As a result of this, when we repeated the simulation 1000 times with different sample sizes $n(= 50, 100, 200)$, our model 2-CPC improved in terms of BIC the best model estimated by *mclust* in 100% of the simulations, for all the values of n considered.

Appendix C Algorithms

C.1 Optimal truncation

In the algorithms presented, we will repeatedly use the *optimal truncation* algorithm explained in Section 3.1 in García-Escudero et al. (2020), which was introduced in Fritz et al. (2013).

Given $d \geq 0$ and a fixed restriction constant $c \geq 1$, the m-truncated value is defined by

$$d^m = \begin{cases} d & \text{if } d \in [m, cm] \\ m & \text{if } d < m \\ cm & \text{if } d > cm \end{cases} .$$

Given $\left\{n_j\right\}_{j=1}^J \in \mathbb{R}_{>0}^J$ and $\left\{d_{j1}, \dots, d_{jL}\right\}_{j=1}^J \in [0, \infty)^{J \times L}$, we define the operator

$$\text{OT}_c \left(\left\{n_j\right\}_{j=1}^J ; \left\{d_{j1}, \dots, d_{jL}\right\}_{j=1}^J \right)$$

which returns $\left\{d_{j1}^*, \dots, d_{jL}^*\right\}_{j=1}^J \in [0, \infty)^{J \times L}$ with $d_{jl}^* = d_{jl}^{m_{opt}}$ for m_{opt} being the optimal threshold value obtained as

$$m_{\text{opt}} = \underset{m}{\text{argmin}} \sum_{j=1}^J n_j \sum_{l=1}^L \left(\log \left(d_{jl}^m \right) + \frac{d_{jl}}{d_{jl}^m} \right) .$$

Obtaining that optimal threshold value only requires the maximization of a real-valued function and m_{opt} can be efficiently obtained by performing only $2 \cdot J \cdot L + 1$ evaluations through a procedure which can be fully vectorized (Fritz et al. 2013).

In the algorithms of the following sections, when working with proportionality models, we will minimize in several situations a function of the type

$$(\beta, \gamma_1, \dots, \gamma_r, \lambda_1, \dots, \lambda_d) \longmapsto \sum_{k=1}^r n_k \sum_{l=1}^d \left(\log(\gamma_k \lambda_l) + \frac{\beta_l^T S_k \beta_l}{\gamma_k \lambda_l} \right) ,$$

being β an orthogonal matrix and $\beta_l, l = 1, \dots, d$ its columns, $\gamma_1, \dots, \gamma_r$ size parameters verifying the size constraint for c_{vol} and $\lambda_1, \dots, \lambda_d$ the common shape parameters verifying the shape constraint for c_{sh} and $\prod_{l=1}^d \lambda_l = 1$. In this situation, the minimization can be made iteratively, taking into account that:

- Fixed the sizes and shapes, the minimization with respect to β can be done with the algorithms proposed in Browne and McNicholas (2014b).
- Fixed the orientation and shapes, the optimal unconstrained values of the size are

  $$\gamma_k^{opt} = \frac{1}{d} \sum_{l=1}^d \frac{\beta_l^T S_k \beta_l}{\lambda_l} \quad k = 1, \dots, r.$$

  Therefore, the optimal restricted values for the size are $\text{OT}_{c_{vol}} \left(\{n_k\}_{k=1}^r ; \{\gamma_k^{opt}\}_{k=1}^r \right)$.
- Fixed the orientation and sizes, the optimal unconstrained values of the shapes are:

  $$\lambda_l^{opt} = \frac{1}{N} \sum_{k=1}^r n_k \frac{\beta_l^T S_k \beta_l}{\gamma_k} \quad l = 1, \dots, d.$$

  The optimal values verifying the constraint c_{sh} are $\text{OT}_{c_{sh}} \left(\{1\} ; \{\lambda_1^{opt}, \dots, \lambda_d^{opt}\} \right)$, and because of the reasoning in Section 3.3 in García-Escudero et al. (2020), the optimal values verifying also $\prod_{l=1}^d \lambda_l = 1$ are obtained normalizing the result of the optimal truncation operator.

When working with CPC models, many times we will come to the conclusion that we have to minimize a slightly different type of function:

$$(\beta, \gamma_1, \dots, \gamma_r, \lambda_{1,1}, \dots, \lambda_{1,d}, \dots, \lambda_{r,d}) \longmapsto \sum_{k=1}^r n_k \sum_{l=1}^d \left(\log(\gamma_k \lambda_{k,l}) + \frac{\beta_l^T S_k \beta_l}{\gamma_k \lambda_{k,l}} \right) .$$

In this case, we can repeat analogous comments for the minimization with respect to the sizes and the orientation matrix. For the shape matrices:

- Fixed the orientation and sizes, the optimal unconstrained values of the shapes are

  $$\lambda_{k,l}^{opt} = \frac{\beta_l^T S_k \beta_l}{\gamma_k} \quad k = 1, \dots, r, \; l = 1, \dots, d.$$

  For each $k = 1, \dots, r$, the optimal values verifying the constraint c_{sh} are the result of the operator $\text{OT}_{c_{sh}} \left(\{1\} ; \{\lambda_{k,1}^{opt} \dots, \lambda_{k,d}^{opt}\} \right)$, and the optimal values verifying also $\prod_{l=1}^d \lambda_l = 1$ are obtained normalizing the result of that truncation.

C.2 Classification G-CPC/G-PROP

In this section we are going to develop the algorithms for the covariance matrices classification models G-CPC and G-PROP minimizing (8) and (9). Since these algorithms are included in the algorithms for cluster analysis, determinant and shape constraints are also included. When focusing on the original problem of Sect. 2, these constraints should be omitted, which can be done taking $c_{vol} = c_{sh} = \infty$. The input of the algorithm is

Classification G-CPC/PROP

$$\left(S_1, \dots, S_K, n_1, \dots, n_K, G, c_{sh}, c_{vol}, nstart_1 \right) ,$$

where $S_1, \dots, S_K$ are the sample covariance matrices, $n_1, \dots, n_K$ the sample lengths, G the number of classes, c_{sh}, c_{vol} the values of the constants for the determinant and shape constraints and $nstart_1$ the number of random initializations. The parameters of the minimization are $\boldsymbol{u} = (u_1, \dots, u_K)$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_K)$ $\boldsymbol{\beta} = (\beta_1, \dots, \beta_G)$ and $\boldsymbol{\Lambda} = (\Lambda_1, \dots, \Lambda_s)$, where $s = K$ in G-CPC and $s = G$ in G-PROP, and they are also the output of the algorithm. A detailed presentation of the algorithm is given as follows:

1. **Initialization:** We start taking a random vector of indexes $\boldsymbol{u^0} \in \mathscr{H}$. Then we take:

$\boldsymbol{\beta^0}$: For each $g = 1, \dots, G$, we take k such that $u_k^0 = g$, and we define β_g as the eigenvectors of S_k.

$\boldsymbol{\Lambda^0}$: → G-PROP: For each $g = 1, \dots, K$, taking the same k as before,

$$\Lambda_g^0 = \mathrm{OT}_{c_{sh}}\Big(\{1\}; \mathrm{diag}(\beta_g^T S_k \beta_g)\Big),$$
$$\Lambda_g^0 = \frac{\Lambda_g^0}{\mathrm{prod}(\Lambda_g^0)^{1/d}} \ .$$

→ G-CPC: For each $k = 1, \dots, K$,

$$\Lambda_k^0 = \mathrm{OT}_{c_{sh}}\Big(\{1\}; \mathrm{diag}\big(\beta_{u_k^0}^T S_k \beta_{u_k^0}\big)\Big),$$
$$\Lambda_k^0 = \frac{\Lambda_k^0}{\mathrm{prod}(\Lambda_k^0)^{1/d}} \ .$$

$\boldsymbol{\gamma^0}$: For each $k = 1, \dots, K$,

$$\rightarrow \text{G-PROP:} \qquad \gamma_k^0 = \frac{1}{d}\,\mathrm{tr}\Big((\Lambda_{u_k^0}^0)^{-1} \beta_{u_k^0}^T S_k \beta_{u_k^0}\Big)$$
$$\rightarrow \text{G-CPC:} \qquad \gamma_k^0 = \frac{1}{d}\,\mathrm{tr}\Big((\Lambda_k^0)^{-1} \beta_{u_k^0}^T S_k \beta_{u_k^0}\Big) \ .$$

Constrained values:

$$(\gamma_1^0, \dots, \gamma_d^0) = \mathrm{OT}_{c_{vol}}\Big(\{n_k\}_{k=1}^K; \{\gamma_k^0\}_{k=1}^K\Big).$$

2. **Iterations:** The following steps are repeated until convergence:

u-V step: Based on the current parameters $\boldsymbol{u^m}, \boldsymbol{\gamma^m}, \boldsymbol{\beta^m}, \boldsymbol{\Lambda^m}$, we are going to optimize with respect to $\boldsymbol{u}$ and the variable parameters of each parsimonious model. The variable parameters will be also optimized in the following step, thus its value will not be updated here. Size parameters $\boldsymbol{\gamma}$ don't affect the selection of the best $\boldsymbol{u}$, thus it is enough to find for each $k = 1, \dots, K$ the value of u_k for which taking the common parameters C_{u_k} we obtain a lower value in the minimization with respect to the variable parameters of

$$R(\beta, \Lambda) = \sum_{l=1}^{d} \frac{\beta_l^T S_k \beta_l}{\lambda_l}.$$

→ G-PROP: The parameters $\boldsymbol{\Lambda}, \boldsymbol{\beta}$ are common, we are only minimizing with respect to $\boldsymbol{u}$. For each $k = 1, \dots, K$,

$$u_k^{m+1} = \underset{g \in \{1,\dots,G\}}{\mathrm{argmin}} \quad R(\beta_g^m, \Lambda_g^m) \ .$$

→ G-CPC: The parameters $\boldsymbol{\beta}$ are common. For each $k = 1, \dots, K$,

$$\tilde{\Lambda}_{k,g} = \mathrm{OT}_{c_{sh}}\Big(\{1\}; \mathrm{diag}\big((\beta_g^m)^T S_k \beta_g^m\big)\Big) \ ,$$
$$\tilde{\Lambda}_{k,g} = \frac{\tilde{\Lambda}_{k,g}}{\mathrm{prod}(\tilde{\Lambda}_{k,g})^{1/d}} \ ,$$
$$u_k^{m+1} = \underset{g \in \{1,\dots,G\}}{\mathrm{argmin}} \quad R(\beta_g^m, \tilde{\Lambda}_{k,g}) \ .$$

V-C step: Based on the current parameters $\boldsymbol{u^{m+1}}, \boldsymbol{\gamma^m}, \boldsymbol{\beta^m}, \boldsymbol{\Lambda^m}$, we are going to optimize with respect to $\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\Lambda}$. This optimization requires iterations. Setting $s = 0$, and considering the initial solutions

$$\bar{\boldsymbol{\gamma}}^{\boldsymbol{0}} = \boldsymbol{\gamma^m} \qquad \bar{\boldsymbol{\beta}}^{\boldsymbol{0}} = \boldsymbol{\beta^m} \qquad \bar{\boldsymbol{\Lambda}}^{\boldsymbol{0}} = \boldsymbol{\Lambda^m}$$

the following steps are repeated until convergence:

$\boldsymbol{s}$: $s = s + 1$

$\bar{\boldsymbol{\gamma}}^{\boldsymbol{s}}$: Update the size parameters. For each $k = 1, \dots, K$,
→ G-PROP:

$$\bar{\gamma}_k^s = \frac{1}{d}\,\mathrm{tr}\Big((\bar{\Lambda}_{u_k^{m+1}}^{s-1})^{-1} (\bar{\beta}_{u_k^{m+1}}^{s-1})^T S_k \bar{\beta}_{u_k^{m+1}}^{s-1}\Big) \ .$$

→ G-CPC:

$$\bar{\gamma}_k^s = \frac{1}{d}\,\mathrm{tr}\Big((\bar{\Lambda}_k^{s-1})^{-1} (\bar{\beta}_{u_k^{m+1}}^{s-1})^T S_k \bar{\beta}_{u_k^{m+1}}^{s-1}\Big) \ .$$

Then we apply the size constraint:

$$\bar{\boldsymbol{\gamma}}^{\boldsymbol{s}} = \mathrm{OT}_{c_{vol}}\Big(\{n_k\}_{k=1}^r; \{\bar{\gamma}_k^s\}_{k=1}^K\Big).$$

$\bar{\boldsymbol{\Lambda}}^{\boldsymbol{s}}$: Update the shape parameters.

→ G-PROP: For each $g = 1, \dots, G$,

$$\bar{\Lambda}_g^s = \mathrm{OT}_{c_{sh}}\Big(\{1\};$$

$$\mathrm{diag}\Big(\frac{1}{N}\sum_{k:u_k^{m+1}=g} n_k \frac{(\bar{\beta}_g^{s-1})^T S_k \bar{\beta}_g^{s-1}}{\bar{\gamma}_k^s}\Big)\Big),$$

$$\bar{\Lambda}_g^s = \frac{\bar{\Lambda}_g^s}{\mathrm{prod}(\bar{\Lambda}_g^s)^{1/d}} .$$

→ G-CPC: For each $k = 1, \dots, K$,

$$\bar{\Lambda}_k^s = \mathrm{OT}_{c_{sh}}\left(\{1\}; \mathrm{diag}\Big(\frac{(\bar{\beta}_{u_k{}^{m+1}}^{s-1})^T S_k \bar{\beta}_{u_k{}^{m+1}}^{s-1}}{\bar{\gamma}_k^s}\Big)\right),$$

$$\bar{\Lambda}_k^s = \frac{\bar{\Lambda}_k^s}{\det(\bar{\Lambda}_k^s)^{1/d}}.$$

$\bar{\boldsymbol{\beta}}^{\boldsymbol{s}}$: Update the rotation parameters. For each $g = 1, \dots, G$, the algorithms in Browne and McNicholas (2014b) allow us to find, for each $g = 1, \dots, G$, a rotation matrix $\bar{\beta}_g^s$ minimizing:

$$\to \text{G-PROP: } \beta \mapsto \sum_{k:u_k^{m+1}=g} n_k \sum_{l=1}^{d} \frac{\beta_l^T S_k \beta_l}{\bar{\gamma}_k^s \bar{\lambda}_{g,l}^s} .$$

$$\to \text{G-CPC: } \quad \beta \mapsto \sum_{k:u_k^{m+1}=g} n_k \sum_{l=1}^{d} \frac{\beta_l^T S_k \beta_l}{\bar{\gamma}_k^s \bar{\lambda}_{k,l}^s} .$$

Once the iterations have finished, we update the parameters

$$\boldsymbol{\gamma}^{\boldsymbol{m+1}} = \bar{\boldsymbol{\gamma}}^{\boldsymbol{s}} \qquad \boldsymbol{\beta}^{\boldsymbol{m+1}} = \bar{\boldsymbol{\beta}}^{\boldsymbol{s}} \qquad \boldsymbol{\Lambda}^{\boldsymbol{m+1}} = \bar{\boldsymbol{\Lambda}}^{\boldsymbol{s}}.$$

3. **Evaluate the target function**: Steps 1 and 2 are repeated $nstart_1$ times. At each step, we evaluate the target function (8) or (9), and we keep the parameters estimated in the iteration with the best value of the target function.

C.3 Clustering G-CPC/G-PROP

In this section we are going to give a detailed explanation of the algorithms for model-based clustering G-CPC and G-PROP presented in Sect. 3.1 for minimizing (11). The algorithms for fitting the corresponding discriminant analysis models can be easily deduced from these. The input of the clustering algorithm is:

clustering G-CPC/PROP

$$\Big(X, G, K, c_{sh}, c_{vol}, nstart_1, nstart_2\Big) ,$$

where X is the matrix with N observations of d variables, G is the number of classes, K is the number of clusters, c_{sh}, c_{vol} are the values for the determinant and shape constraints, $nstart_1$ is the number of random initializations in the classification algorithm, and $nstart_2$ is the number of random initialization in the clustering algorithm. The parameters of the minimization are $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)$, $\boldsymbol{u} = (u_1, \dots, u_K)$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_K)$ $\boldsymbol{\beta} = (\beta_1, \dots, \beta_G)$ and $\boldsymbol{\Lambda} = (\Lambda_1, \dots, \Lambda_s)$, where $s = K$ in G-CPC and $s = G$ in G-PROP. A detailed presentation of the algorithm is given as follows:

1. **Initialization:** We start taking a random vector of indexes $\boldsymbol{u^0} \in \mathscr{H}$. Then we take:

   $\boldsymbol{\pi^0}$: Equal weights: $\pi_k^0 = \frac{1}{K} \quad k = 1, \dots, K$.

   $\boldsymbol{\mu^0}$: Denote by $(\bar{\mu}_1, \dots, \bar{\mu}_K)$ the solution obtained by the R function $tclust$ (García-Escudero et al. 2008; Fritz et al. 2012) for a random sample of length $N/2$ of X, number of groups K, eigenvalue constraint given by $c = \min\{c_{sh}, c_{vol}\}$ and a suitable number of starts. We are considering as initial solution a random perturbation of the values obtained. If $S = \mathrm{cov}(X)$, we are considering

   $$\mu_k^0 = \bar{\mu}_k + \frac{1}{10} N(0, S) \qquad k = 1, \dots, K.$$

   $\boldsymbol{\beta^0}$: $\beta_g^0 = I_d, \quad g = 1, \dots, G.$

   $\boldsymbol{\Lambda^0}$: $\Lambda_k^0 = (1, \dots, 1), \quad k = 1, \dots, G$ in G-PROP, and $k = 1, \dots, K$ in G-CPC.

   $\boldsymbol{\gamma^0}$: $\gamma_k^0 = 1, \quad k = 1, \dots, K.$
   (This simple initial solution verifies determinant and shape constraint independently of the c_{sh} and c_{vol} values)

2. **Iterations:** The E and M steps are repeated until convergence:

- **E step**: Given the current values of the parameters $\boldsymbol{\pi^m}, \boldsymbol{\mu^m}, \boldsymbol{u^m}, \boldsymbol{V^m}, \boldsymbol{C^m}$, we compute the posterior probabilities

  $$z_{i,k} = \frac{\pi_k \, \phi\Big(y_i \Big| \mu_k^m, \Sigma_k^m\Big)}{\sum_{l=1}^K \pi_l \, \phi\Big(y_i | \mu_l^m, \Sigma_l^m\Big)}$$

  for $k = 1, \dots, K \quad i = 1, \dots, N$, where the matrix Σ_k^m is defined by

  $$\to \text{G-PROP: } \Sigma_k^m = \gamma_k^m \beta_{u_k^m} \Lambda_{u_k^m}^m \beta_{u_k^m}^T .$$

  $$\to \text{G-CPC: } \quad \Sigma_k^m = \gamma_k^m \beta_{u_k^m} \Lambda_k^m \beta_{u_k^m}^T .$$

- **M step**: In this step, we have to maximize the complete log-likelihood (12) given the expected values $\{z_{i,k}\}_{i,k}$.

$$\begin{aligned}\boldsymbol{n}:&\; n_k = \textstyle\sum_{i=1}^{N} z_{i,k} \quad k = 1, \ldots, K.\\ \boldsymbol{\pi}^{\boldsymbol{m+1}}:&\; \pi_k^{m+1} = \frac{n_k}{N} \quad k = 1, \ldots, K.\\ \boldsymbol{\mu}^{\boldsymbol{m+1}}:&\; \mu_k^{m+1} = \frac{\sum_{i=1}^{N} z_{i,k} y_i}{n_k} \quad k = 1, \ldots, K.\\ \boldsymbol{S}:&\; \text{For } k = 1, \ldots, K:\end{aligned}$$

$$S_k = \frac{1}{n_k} \sum_{i=1}^{N} z_{i,k} (y_i - \mu_k^{m+1})(y_i - \mu_k^{m+1})^T.$$

Class.: We solve the covariance matrix classification problem for the computed values:

$$(\boldsymbol{u}^{\boldsymbol{m+1}}, \boldsymbol{\gamma}^{\boldsymbol{m+1}}, \boldsymbol{\Lambda}^{\boldsymbol{m+1}}, \boldsymbol{\beta}^{\boldsymbol{m+1}}) = \textbf{Classification G-CPC/PROP} \left(S_1, \ldots, S_K, n_1, \ldots, n_K, G, c_{sh}, c_{vol}, nstart_1\right).$$

3. **Evaluate the target function**: Steps 1 and 2 are repeated $nstart_2$ times. At the end of each different initialization, we evaluate the target function (11), and we keep the parameters estimated in the iteration with the best value of the target function.

References

Azzalini, A., Menardi, G.: Clustering via nonparametric density estimation: the R package pdfCluster. J. Stat. Softw. **57**(11), 1–26 (2014). https://doi.org/10.18637/jss.v057.i11

Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics **49**(3), 803–821 (1993). https://doi.org/10.2307/2532201

Benaglia, T., Chauveau, D., Hunter, D., et al.: mixtools: an R package for analyzing finite mixture models. J. Stat. Softw. **32**(6), 1–29 (2009). https://doi.org/10.18637/jss.v032.i06

Bensmail, H., Celeux, G.: Regularized Gaussian discriminant analysis through eigenvalue decomposition. J. Am. Stat. Assoc. **91**(436), 1743–1748 (1996). https://doi.org/10.1080/01621459.1996.10476746

Bezdek, J.C., Hathaway, R.J., Howard, R.E., et al.: Local convergence analysis of a grouped variable version of coordinate descent. J. Opt. Theory Appl. **54**(3), 471–477 (1987). https://doi.org/10.1007/bf00940196

Biernacki, C., Govaert, G.: Choosing models in model-based clustering and discriminant analysis. J. Stat. Comput. Simul. **64**(1), 49–71 (1999). https://doi.org/10.1080/00949659908811966

Browne, R.P., McNicholas, P.D.: Estimating common principal components in high dimensions. Adv. Data Anal. Classif. **8**(2), 217–226 (2014). https://doi.org/10.1007/s11634-013-0139-1

Browne, R.P., McNicholas, P.D.: Orthogonal Stiefel manifold optimization for eigen-decomposed covariance parameter estimation in mixture models. Stat. Comput. **24**(2), 203–210 (2014). https://doi.org/10.1007/s11222-012-9364-2

Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Comp. Stat. Data Anal. **14**(3), 315–332 (1992). https://doi.org/10.1016/0167-9473(92)90042-E

Celeux, G., Govaert, G.: Gaussian parsimonious clustering models. Pattern Recognit. **28**(5), 781–793 (1995). https://doi.org/10.1016/0031-3203(94)00125-6

Dean, N., Murphy, T.B., Downey, G.: Using unlabelled data to update classification rules with applications in food authenticity studies. J. R. Stat. Soc. Ser. C Appl. Stat. **55**(1), 1–14 (2006). https://doi.org/10.1111/j.1467-9876.2005.00526.x

del Barrio, E., Cuesta-Albertos, J.A., Matrán, C., et al.: Robust clustering tools based on optimal transportation. Stat. Comput. **29**(1), 139–160 (2019). https://doi.org/10.1007/s11222-018-9800-z

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B Stat Methodol. **39**(1), 1–22 (1977). https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Flury, B.: Common principal components in k groups. J. Am. Stat. Assoc. **79**(388), 892–898 (1984). https://doi.org/10.1080/01621459.1984.10477108

Flury, B.: Common Principal Components and Related Multivariate Models. Wiley, New York (1988)

Flury, B.N., Gautschi, W.: An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. SIAM J. Sci. Comput. **7**(1), 169–184 (1986). https://doi.org/10.1137/0907013

Flury, B.W., Schmid, M.J., Narayanan, A.: Error rates in quadratic discrimination with constraints on the covariance matrices. J. Classif. **11**, 101–120 (1994). https://doi.org/10.1007/bf01201025

Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. J. Am. Stat. Assoc. **97**(458), 611–631 (2002). https://doi.org/10.1198/016214502760047131

Friedman, H.P., Rubin, J.: On some invariant criteria for grouping data. J. Am. Stat. Assoc. **62**(320), 1159–1178 (1967). https://doi.org/10.1080/01621459.1967.10500923

Fritz, H., García-Escudero, L.A., Mayo-Iscar, A.: tclust: an R package for a trimming approach to cluster analysis. J. Stat. Softw. **47**(12), 1–26 (2012). https://doi.org/10.18637/jss.v047.i12

Fritz, H., García-Escudero, L.A., Mayo-Iscar, A.: A fast algorithm for robust constrained clustering. Comp. Stat. Data Anal. **61**, 124–136 (2013). https://doi.org/10.1016/j.csda.2012.11.018

García-Escudero, L.A., Gordaliza, A., Matrán, C., et al.: A general trimming approach to robust cluster analysis. Ann. Stat. **36**(3), 1324–1345 (2008). https://doi.org/10.1214/07-AOS515

García-Escudero, L., Gordaliza, A., Matrán, C., et al.: Avoiding spurious local maximizers in mixture modeling. Stat. Comput. **25**, 619–633 (2015). https://doi.org/10.1007/s11222-014-9455-3

García-Escudero, L., Gordaliza, A., Greselin, F., et al.: Eigenvalues and constraints in mixture modeling: geometric and computational issues. Adv. Data Anal. Classif. **12**, 203–233 (2017). https://doi.org/10.1007/s11634-017-0293-y

García-Escudero, L., Mayo, A., Riani, M.: Model-based clustering with determinant-and-shape constraint. Stat. Comput. **30**, 1363–1380 (2020). https://doi.org/10.1007/s11222-020-09950-w

García-Escudero, L.A., Mayo-Iscar, A., Riani, M.: Constrained parsimonious model-based clustering. Stat. Comput. **32**(1), 2 (2022). https://doi.org/10.1007/s11222-021-10061-3

Lebret, R., Iovleff, S., Langrognet, F., et al.: Rmixmod: the R package of the model-based unsupervised, supervised, and semi-supervised classification mixmod library. J. Stat. Softw. **67**(6), 1–29 (2015). https://doi.org/10.18637/jss.v067.i06

Riani, M., Perrotta, D., Torti, F.: FSDA: a MATLAB toolbox for robust analysis and interactive data exploration. Chemom. Intell. Lab. Syst. **116**, 17–32 (2012). https://doi.org/10.1016/j.chemolab.2012.03.017

Scrucca, L., Fop, M., Murphy, T., et al.: mclust 5: clustering, classification and density estimation using gaussian finite mixture models. R J. **8**, 205–233 (2016). https://doi.org/10.32614/RJ-2016-021

Thorpe, R.: A review of the numerical methods for recognising and analysing racial differentiation. In: Felsenstein, J. (ed.) Numerical Taxonomy, pp. 404–423. Springer, Berlin (1983). https://doi.org/10.1007/978-3-642-69024-2_43

Venables, W.N., Ripley, B.D.: Modern Applied Statistics With S, 4th edn. Springer, New York (2002)

Weinstein, J.N., Collisson, E.A., Mills, G.B., et al.: The cancer genome atlas pan-cancer analysis project. Nat. Genet. **45**, 1113–1120 (2013). https://doi.org/10.1038/ng.2764

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.