Statistics and Computing (2022) 32:2
https://doi.org/10.1007/s11222-021-10061-3

Constrained parsimonious model-based clustering

Luis A. García-Escudero[1] · **Agustín Mayo-Iscar**[1] · **Marco Riani**[2]

Received: 4 April 2021 / Accepted: 24 October 2021 / Published online: 20 November 2021

Abstract
A new methodology for constrained parsimonious model-based clustering is introduced, where some tuning parameter allows to control the strength of these constraints. The methodology includes the 14 parsimonious models that are often applied in model-based clustering when assuming normal components as limit cases. This is done in a natural way by filling the gap among models and providing a smooth transition among them. The methodology provides mathematically well-defined problems and is also useful to prevent us from obtaining spurious solutions. Novel information criteria are proposed to help the user in choosing parameters. The interest of the proposed methodology is illustrated through simulation studies and a real-data application on COVID data.

Keywords Model-based clustering · Mixture modeling · Constraints

1 Introduction

Model-based clustering is a well-established and powerful approach to cluster analysis. Fitting k multivariate Gaussian distributed components to data is the most widely applied methodology and maximum likelihood is the principle often adopted for the fitting procedure.

This research is partially supported by Spanish Ministerio de Economía y Competitividad, Grant MTM2017-86061-C2-1-P, and by Consejería de Educación de la Junta de Castilla y León and FEDER, Grant VA005P17 and VA002G18 and by CRoNoS COST Action IC1408 and the project "Statistics for fraud detection, with applications to trade data and financial statements" of the University of Parma. This research benefits from the High Performance Computing facility of the University of Parma. Authors also thank Dr. P. Clavario and Dr. E. Capurro for providing us the COVID data and Anthony Atkinson for his careful reading of the manuscript and helpful comments.

✉ Luis A. García-Escudero
lagarcia@eio.uva.es

Agustín Mayo-Iscar
agustinm@eio.uva.es

Marco Riani
mriani@unipr.it

[1] Department of Statistics and Operational Research and IMUVA, University of Valladolid, Valladolid, Spain

[2] Department of Economics and Management and Interdepartmental Centre of Robust Statistics, University of Parma, Parma, Italy

In this work, we use the notation $\phi(\cdot; \mu, \Sigma)$ for the probability density functions of the p-variate normal distribution with mean μ and covariance matrix Σ. Given a sample of p-dimensional observations $\{x_1, \cdots, x_n\}$, the *classification likelihood* approach searches for a partition $\{H_1, \ldots, H_k\}$ of the $\{1, \cdots, n\}$ indices, mean vectors $\mu_1, \cdots, \mu_k$ in $\mathbb{R}^p$, symmetric positive semidefinite $p \times p$ scatter matrices $\Sigma_1, \cdots, \Sigma_k$ and positive weights $\pi_1, \cdots, \pi_k$ with $\sum_{j=1}^{k} \pi_j = 1$, which maximizes

$$\sum_{j=1}^{k} \sum_{i \in H_j} \log\left(\pi_j \phi(x_i; \mu_j, \Sigma_j)\right). \tag{1}$$

Alternatively, the *mixture likelihood* approach seeks the maximization of

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} \pi_j \phi(x_i; \mu_j, \Sigma_j)\right). \tag{2}$$

An important problem when maximizing (1) and (2) is that these two target likelihood functions are unbounded ones (Kiefer and Wolfowitz 1956; Day 1969). Another important issue is the typically large number of local maxima that can be found. In the mixture likelihood case, the existence of a sequence of local maxima converging to the true mixture parameters is guaranteed as the sample size n increases. However, it is not obvious how to choose those local maxima in

practical applications. In fact, many local maxima related to these very high values of the likelihoods are known to be clearly non-interesting and often referred to as "spurious solutions" (see, e.g., chapter 3.10 in McLachlan and Peel (2000)). In these cases, components basically defined from a few, almost collinear, observations are obtained. Algorithms applied for maximizing the target likelihood (EM algorithms when maximizing (1) and CEM algorithm when maximizing (2)) can be affected by unboundedness, being trapped into sub-optimal maxima or detect non-interesting local maxima. This is even more problematic when applying well-known information criteria (such as BIC and ICL). These criteria are based on penalized versions of the target likelihood values and spurious solutions or the unboundedness issue can result in artificially large values for the likelihood. Note also that it is necessary, when choosing k, to fit models with a higher than needed number of components. All the previously mentioned problems are even more likely there to appear.

These problems with local maxima can be in principle solved by carefully exploring and analyzing *all* possible local maxima (McLachlan and Peel 2000). Although some interesting procedures have been introduced in that direction (see, e.g., Gallegos and Ritter (2018)), this approach is not straightforward and is certainly time consuming. Another widely applied remedy consists in trying to initialize the algorithms adequately in order that iterations return good local maxima. It is well-known that EM and CEM algorithms are highly dependent on their initialization, but it is also true that adequate initialization strategies (for instance, appropriate hierarchical model-based clustering initializations) often result in sensible local maxima. However, theoretical guarantees about correctness of initializations are difficult to establish and it may happen that the final fitted model inherits significant drawbacks from the initializing procedure. Additionally, if two different initializations provide different final results, it is difficult to justify not choosing the one with the larger value of the likelihood without any further analysis. In fact, some initialization procedures are clearly aimed at searching directly for the largest values (see, e.g. Biernacki et al. 2003).

It is also common to enforce constraints on the Σ_j scatter matrices when maximizing (1) or (2). Among them, the use of "parsimonious" models (Celeux and Govaert 1995; Banfield and Raftery 1993) is one of the most popular and widely applied approaches in practice. These parsimonious models follow from a decomposition of the Σ_j scatter matrices as

$$\Sigma_j = \lambda_j \Omega_j \Gamma_j \Omega_j', \tag{3}$$

with $\lambda_j = |\Sigma_j|^{1/p}$ (volume parameters),

$$\Gamma_j = \mathsf{diag}(\gamma_{j1}, \ldots, \gamma_{jl}, \ldots, \gamma_{jp}) \text{ with}$$

$$\det(\Gamma_j) = \prod_{l=1}^{p} \gamma_{jl} = 1$$

(shape matrices), and Ω_j (rotation matrices) with $\Omega_j \Omega_j' = \mathsf{I}_p$. Different constraints on the λ_j, Ω_j and Γ_j elements are considered across components to get 14 parsimonious models (which are coded with a combination of three letters). These models reduce notably the number of free parameters to be estimated, so improving efficiency and model interpretability. Moreover, many of them turn the constrained maximization of the likelihoods into well-defined problems and help to avoid spurious solutions. Unfortunately, the problems remain for models with unconstrained λ_j volume parameters, which are coded with the first letter as a `V` (`V**` models). Aside from relying on good initializations, it is common to consider the early stopping of iterations when approaching scatter matrices with very small eigenvalues or when detecting components accounting for a reduced number of observations. A not fully iterated solution (or no solution at all) is then returned in these cases. The idea is known to be problematic when dealing with (well-separated) components made up of a few observations.

Starting from a seminal paper by Hathaway (1985), an alternative approach is to constrain the Σ_j scatter matrices by specifying some tuning constants that control the strength of the constraints. A fairly comprehensive review of this approach can be found in García-Escudero et al. (2018). For instance, a recent proposal following this idea is the "deter-and-shape" one in García-Escudero et al. (2020). The maximal ratio among the λ_j terms is there constrained to be smaller than a fixed constant and, additionally, the maximal ratio $\gamma_{jl}/\gamma_{jl'}$ in each Γ_j shape matrix is also constrained to be smaller than another fixed constant. In this work, we will refer to these second type of constraint as "shape-within" as they control the relative sizes of the shape elements "within" each shape matrix individually. When this second constant is set equal to 1, since all γ_{jl} are then equal 1, we are imposing spherical components.

In this work, we introduce new "shape-between" constraints where the maximal ratio $\gamma_{jl}/\gamma_{j'l}$ is controlled for every fixed l for $l = 1, \ldots, p$. Figure 1 shows a summary of the two types of constraints considered on the shape elements. Notice that we have $\gamma_{jl} = \gamma_{j'l}$ for every j and j' in the most constrained case, but the fitted components are not necessarily spherical. Therefore, these new constraints are better suited to control differences among shape matrices without assuming sphericity. The new constraints can be easily combined with others typically imposed on the Ω_j rotation matrices.

The main contributions of this work are the following:

(a) The proposed constraints yield well-defined problems and it is not necessary to include the specification of any particular initialization strategy. An underlying population (theoretical) problem can thus be defined. Section 4 shows existence results for both the sample and the pop-

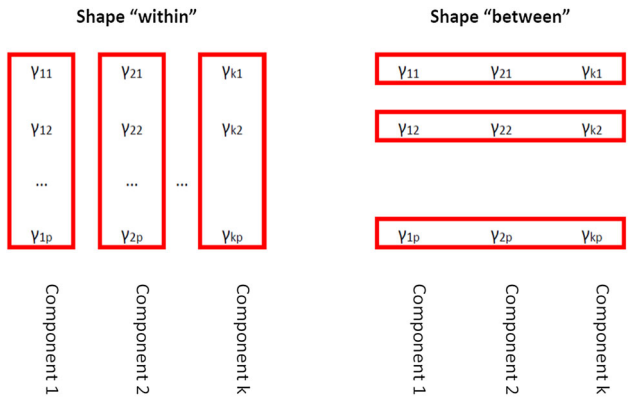

Fig. 1 Summary of the types of constraints considered on the shape elements

ulation problems and we prove the consistency for the sample solutions to the population one.

(b) The new constraints allow us to achieve, as limit cases, the 14 parsimonious models which are commonly applied in model-based clustering. These popular parsimonious models cannot be obtained as limit cases when only considering "deter-and-shape(within)" constraints or other constraints such as the ones based on eigenvalues ratios (García-Escudero et al. 2015). However, contrary to what happens with `V**` models, the associated likelihood maximization problems are always well defined. It is perhaps too extreme to choose only among strongly constrained models (maximal ratios exactly equal to 1) and the fully unconstrained ones (maximal ratios taking arbitrarily large values). Sometimes it is clear that data do not suggest considering the most constrained models, but leaving them fully unrestricted may cause estimation instabilities and the detection of spurious solutions. A smooth transition between these extreme cases can be obtained with the proposed methodology. An interesting connection between the two types of constraints (between-within) in the shape matrices elements is given in Sect. 2, together with some practical consequences.

(c) Although 14 different algorithms are often employed to estimate the classical parsimonious models, a unifying algorithm is proposed in Sect. 3 which includes all the 14 classical parsimonious models as limit cases.

(d) Some general guidelines about how to choose the tuning parameters are provided. In fact, the smooth transition among models turns out to be useful to introduce novel information criteria, inspired by those in Cerioli et al. (2018). These criteria penalize high likelihood values resulting from unnecessary model complexity associated with the constraints. Model complexity here does not necessarily correspond to the total number of parameters, but it simply means that more flexibility in the constraints allows us to fit more varied models. This proposal can be seen as a first step in order to obtain a reduced list of "sensible" cluster solutions, as done in Cerioli et al. (2018).

Some simulations and a real data example are provided in Sects. 6 and 7 to illustrate the interest of the proposed methodology. We do not claim that the well-established and widely applied proposals considered for comparison are useless; they have amply demonstrated their usefulness. However, we illustrate that the proposed methodology can also be very useful and that there is room for further investigations of this new proposal. Concluding remarks and open research directions are given in Sect. 8.

2 Proposed methodology

We impose three different types of constraints on the Σ_j matrices which depend on three constants c_{det}, c_{shw} and c_{shb} all of them being greater than or equal to 1.

The first type of constraint serves to control the maximal ratio among determinants and, consequently, the maximum allowed difference between component volumes:

$$\text{"deter"}: \quad \frac{\max_{j=1,\ldots,k} |\Sigma_j|}{\min_{j=1,\ldots,k} |\Sigma_j|} = \frac{\max_{j=1,\ldots,k} \lambda_j^p}{\min_{j=1,\ldots,k} \lambda_j^p} \leq c_{\mathsf{det}}. \tag{4}$$

The second type of constraint controls departures from sphericity "within" each component:

$$\text{shape-"within"}: \quad \frac{\max_{l=1,\ldots,p} \gamma_{jl}}{\min_{l=1,\ldots,p} \gamma_{jl}} \leq c_{\mathsf{shw}} \text{ for } j = 1, \ldots, k. \tag{5}$$

This provides a set of k constraints that in the most constrained case, $c_{\mathsf{shw}} = 1$, imposes $\Gamma_1 = \cdots = \Gamma_p = \mathsf{I}_p$, where I_p is the identity matrix of size p, i.e., sphericity of components.

Constraints (4) and (5) were the basis for the "deter-and-shape" constraints in García-Escudero et al. (2020). These two constraints resulted in mathematically well-defined constrained maximizations of the likelihoods in (1) and (2). However, although highly operative in many cases, they do not include, as limit cases, all the already mentioned 14 parsimonious models. For instance, we may be interested in the same (or not very different) Γ_j or Σ_j matrices for all the mixture components and these cannot be obtained as limit cases from the "deter-and-shape" constraints.

In this work, we introduce a third type of constraint that serves to control the maximum allowed difference between shape elements "between" components:

$$\text{shape-"between"}: \quad \frac{\max_{j=1,\ldots,k} \gamma_{jl}}{\min_{j=1,\ldots,k} \gamma_{jl}} \leq c_{\mathsf{shb}} \text{ for } l = 1, \ldots, p. \tag{6}$$

This new type of constraint allows us to impose "similar" shape matrices for the components and, consequently, enforce $\Gamma_1 = \cdots = \Gamma_k$ in the most constrained $c_{\mathsf{shb}} = 1$ case .

Additionally, another type of constraint on the rotation Ω_j matrices can be combined with the previous ones. Three different constraints `rot` on the rotation matrices can be considered and coded with the letters `E`, `I` and `V`. If `rot=E`, then we are assuming the same rotation matrices $\Omega_1 = \cdots = \Omega_k$ for all the components. If `rot=I`, then we are assuming $\Omega_1 = \cdots = \Omega_k = \mathsf{I}_p$, i.e. axes parallel to the coordinate axes (conditional independence within cluster components). Finally, `rot=V` leaves the rotation matrices Ω_j fully unconstrained.

In the third case of fully unconstrained rotation matrices `rot=V`, we choose the diagonal elements of Γ_j (by choosing the appropriate rotation Ω_j matrices) such that these shape elements appear in non-increasing order:

$$\gamma_{j1} \geq \cdots \geq \gamma_{jl} \geq \cdots \geq \gamma_{jp}. \tag{7}$$

This ordering makes sense since adequate rotations (in the `rot=V` case) can be performed such that ordered elements within each shape matrix are achieved.

The following lemma shows an interesting connection between the two different types of constraints on the shape matrices.

Lemma 1 *If the "shape-within" constraints* (5) *are satisfied for a constant* $c_{\mathsf{shw}} \geq 1$*, then*

$$\frac{\gamma_{jl}}{\gamma_{j'l}} \leq c_{\mathsf{shw}}^{(p-1)/p}, \tag{8}$$

for any $j, j' \in \{1, \ldots, k\}$ *and* $l = 1, \ldots, p$.

The proof of this technical lemma is given in Appendix A. When taking into account the definition of the "shape-between" constraints in (6) as a maximal ratio, the previous lemma implies that the choice of c_{shw} in (5) modifies the effect of c_{shb} in (6) and that there is no point in considering c_{shb} not obeying

$$c_{\mathsf{shb}} \leq c_{\mathsf{shw}}^{(p-1)/p}. \tag{9}$$

For instance, this implies that $c_{\mathsf{shb}} \leq \sqrt{c_{\mathsf{shw}}}$ when we are in dimension $p = 2$ and that we are obviously assuming $c_{\mathsf{shb}} = 1$ whenever we set $c_{\mathsf{shw}} = 1$.

An important consequence is that, although we potentially have $2^3 \times 3 = 24$ different extreme models (appearing when c_{det}, c_{shw} and c_{shb} are chosen equal to 1 or ∞ and the three possible constraints `rot` on the rotations), not all these 24 models are feasible (because $c_{\mathsf{shw}} = 1$ necessarily implies $c_{\mathsf{shb}} = 1$) and only the 14 well-known parsimonious models make sense. Table 1 shows how these 14 limit models are derived from different combinations of constraints (this table only includes 14 rows). Table 1 helps to understand the smooth transition among all these 14 models when constants c_{det}, c_{shw} and c_{shb} are moved in a controlled fashion. This smooth transition is useful for introducing the novel information criteria in Sect. 5.

Table 1 Extreme models for the different limiting values of constants c_{det}, c_{shw} and c_{shb} and the three possible rotations in `rot`

c_{det}	c_{shw}	c_{shb}	`rot`	Model
1	1	1		EII
	∞	1	`I`	EEI
			`E`	EEE
			`V`	EEV
		∞	`I`	EVI
			`E`	EVE
			`V`	EVV
∞	1	1		VII
	∞	1	`I`	VEI
			`E`	VEE
			`V`	VEV
		∞	`I`	VVI
			`E`	VVE
			`V`	VVV

The "deter-and-shape" constraints also appear as a limit cases when c_{shb} tends to ∞ and `rot=V` is chosen. Notice that Lemma 1 implies that when c_{shw} is chosen close to 1 in the "deter-and-shape" approach, then we are also (implicitly) assuming that c_{shb} is close to 1 too. On the contrary, a large c_{shw} is still compatible with a c_{shb} as close to 1 as desired. In fact, choosing moderate values for c_{det} and c_{shb} (but not exactly equal to 1) and fixing a very large c_{shw} value, together with `rot=V`, turns out to be convenient and advisable, providing a very competitive procedure in many cases.

3 Algorithm

In this section, we introduce a feasible ECM algorithm (Meng and Rubin 1993) that can be applied to the proposed methodology. This algorithm covers all the 14 classical parsimonious models as limit cases in a unified fashion and therefore we do not need to consider 14 different algorithms. The "optimal truncation" operator introduced in Fritz et al. (2013), denoted as opt.trunc, plays a very important role, as it has in previous

constrained model-based clustering approaches. For the sake of completeness, this operator is reviewed in Appendix B.

The proposed algorithm follows analogous steps as EM and CEM algorithms, but iterative procedures may be needed at some points. In the t-th step, the constrained scatter matrices are going to be updated as

$$\Sigma_j^{(t)} = d_j R_j D_j R_j',$$

for some $d_j > 0$, diagonal matrices D_j with $|D_j| = 1$ and R_j being orthogonal matrices. All these d_j, R_j and D_j elements are determined through iterative procedures where, roughly speaking, these elements are sequentially improved in turn by optimally updating one of them (in the sense of increasing the target likelihood and fulfilling the constraints) conditionally on the other elements. Further iterations are sometimes required for D_j and R_j within that iterative procedure.

Iterations can be stopped when reaching a maximal number of iterations and we use the notation "`iter.max.***`" when referring to maximal number of allowed iterations. Additionally, it is useful to monitor "relative changes" in updated parameters and stop iterations whenever relative changes are found smaller than some pre-specified small tolerances. We use the simplified notation Δb for measuring the relative change in the h-th iteration of parameter b denoted by $b^{(h)}$ with respect to $b^{(h-1)}$ in the previous $(h-1)$-th step (we simplify the notation by deleting the dependence on index h). All these small tolerances are going to be notated as "`tol.***`." More details on these aspects are provided in Remark 1. Additionally, `nstart` controls the total number of random initializations.

We denote the parameters at the t-th step of the proposed algorithm by $\theta^{(t)} = (\pi_1^{(t)}, \cdots, \pi_k^{(t)}, \mu_1^{(t)}, \cdots, \mu_k^{(t)}, \Sigma_1^{(t)}, \cdots, \Sigma_k^{(t)})$ and $W_j(x; \theta^{(t)}) = \pi_j^{(t)} \phi(x; \mu_j^{(t)}, \Sigma_j^{(t)})$.

1. *Initialization:* The procedure is initialized `nstart` times by randomly selecting different initial $\theta^{(0)} = (\pi_1^{(0)}, \ldots, \pi_k^{(0)}, \mu_1^{(0)}, \ldots, \mu_k^{(0)}, \Sigma_1^{(0)}, \ldots, \Sigma_k^{(0)})$ sets of parameters. A simple strategy for this initialization is to randomly select $k \times (p+1)$ observations and use them, after splitting them into k groups, to compute k initial $\mu_j^{(0)}$ centers and k initial scatter matrices $\Sigma_j^{(0)}$. It may happen that the initial $\Sigma_j^{(0)}$ matrices do not satisfy the required constraints but the constraints will be imposed in the following iterative step.
2. *Iterative step:*
   $t \leftarrow t+1$

   2.1. *Computing observation weights:* From $\theta^{(t-1)}$, observation weights are computed as

$$\tau_j(x_i; \theta^{(t)}) = \begin{cases} 1 \text{ if } W_j(x_i; \theta^{(t-1)}) \\ \quad = \max\{W_1(x_i; \theta^{(t-1)}), \ldots, W_k(x_i; \theta^{(t-1)})\} \\ 0 \text{ if not} \end{cases},$$

   when maximizing (1) and the associated H_j sets are $H_j^{(t)} = \{i : \tau_j(x_i; \theta^{(t)}) = 1\}$. On the other hand, when maximizing (2), observation weights are computed as

$$\tau_j(x_i; \theta^{(t)}) = \frac{W_j(x_i; \theta^{(t-1)})}{\sum_{j=1}^k W_j(x_i; \theta^{(t-1)})}.$$

   2.2. *Updating component weights:* From these $\tau_j(x_i; \theta^{(t)})$ weights, we define

$$n_j = \sum_{i=1}^n \tau_j(x_i; \theta^{(t)}),$$

   and the component weights are updated as

$$\pi_j^{(t)} = n_j/n.$$

   2.3. *Updating location parameters:* Location parameters are updated as

$$\mu_j^{(t)} = \frac{1}{n_j} \sum_{i=1}^n \tau_j(x_i; \theta^{(t)}) x_i.$$

   2.4. *Updating scatter matrices:* Updating the scatter matrices $\Sigma_j^{(t)}$ is not so straightforward. As previously commented, the updated scatter matrices are obtained as $\Sigma_j^{(t)} = d_j R_j D_j R_j'$, where these d_j, D_j and R_j terms have to be obtained through iterations. Our starting point is the k weighted sample covariance matrices defined as

$$S_j = \frac{1}{n_j} \sum_{i=1}^n \tau_j(x_i; \theta^{(t)})(x_i - \mu_j^{(t)})(x_i - \mu_j^{(t)})'.$$

   2.4.1 *Initialization:*
   $u \leftarrow 0$
   Initially set $d_j = |S_j|^{1/p}$ and R_j as follows:
   `rot=V` Take R_j as the matrix whose columns are the eigenvectors of the S_j matrices associated with their eigenvalues in decreasing order.
   `rot=I` Take $R_1 = \cdots = R_k = \mathrm{I}_p$.
   `rot=E` Take $R_1 = \cdots = R_k = R$ where R is the matrix whose columns are the eigenvectors of the "pooled" scatter matrix

$S = \sum_{j=1}^{k} \frac{n_j}{n} \frac{1}{d_j} S_j$ associated to its eigenvalues.

2.4.2 *Iterative part:*
$u \leftarrow u + 1$

2.4.2.1 *Improving the shape D_j matrices (needs iterations through $D_j^{(s)}$):*

(i) $s \leftarrow 0$ and initialize $D_j^{(0)}$ as follows:

`rot=V` $D_j^{(0)} = \texttt{diag}(R_j' S_j R_j)/d_j$

`rot=I` $D_j^{(0)} = \mathsf{diag}(\sum_{j=1}^{k} \frac{n_j}{n} \frac{1}{d_j} S_j)$

`rot=E` $D_j^{(0)} = \mathsf{diag}(\sum_{j=1}^{k} \frac{n_j}{n} \frac{1}{d_j} R S_j R')$

(ii) Take $s \leftarrow s + 1$ and apply the "within" constraints as:

$$\{e_{j1}, \ldots, e_{jp}\} \leftarrow \mathsf{opt.trunc}_{c_{\mathsf{shw}}}\big(\{1\}; \{d_{j1}^{(s-1)}, \ldots, d_{jp}^{(s-1)}\}\big),$$

for $j = 1, \ldots, k$.

(iii) Normalize the elements in $\{e_{j1}, \ldots, e_{jp}\}$ in order to get unit determinant shape matrices by taking $e_{jl} \leftarrow e_{jl} / \sqrt[p]{\prod_{l=1}^{p} e_{jl}}$ for $l = 1, \ldots, p$ and $j = 1, \ldots, k$.

(iv) `rot=V` case: Consider a permutation σ_j, which serves to sort the previous elements in decreasing order as $e_{j\sigma_j(1)} \geq \cdots \geq e_{j\sigma_j(p)}$ and take $e_{jl} \leftarrow e_{j\sigma_j(l)}$ for $l = 1, \ldots, p$.

v) Apply the "between" constraints:

$$\{e_{1l}, \ldots, e_{kl}\} \leftarrow \mathsf{opt.trunc}_{c_{\mathsf{shb}}}\big(\{n_j\}_{j=1}^{k}; \{e_{1l}, \ldots, e_{kl}\}\big)$$

for $l = 1, \ldots, p$.

(vi) `rot=V` case: Undo the order transformations, $e_{jl} \leftarrow e_{j\sigma_j^{-1}(l)}$.

vii) Update $D_j^{(s)} = \mathsf{diag}(d_{j1}^{(s)}, \ldots, d_{jp}^{(s)}) \leftarrow \mathsf{diag}(e_{j1}, \ldots, e_{jp})$

viii) Go back to ii) if $s <$ `iter.max.D` and $\Delta D^{(\cdot)} >$ `tol.D` or otherwise conclude iterations and finally update

$$D_j \leftarrow D_j^{(s)}.$$

2.4.2.2 *Improving the volume d_j parameters:*
Compute $\nu_1, \ldots, \nu_k$ with

$$\nu_j = \mathsf{trace}(D_j^{-1} R_j' S_j R_j)/p$$

and update the d_j parameters

$$\{d_1, \ldots, d_k\} \leftarrow \mathsf{opt.trunc}_{c_{\mathsf{det}}^{1/p}}\big(\{n_j\}_{j=1}^{k}; \{\nu_1, \ldots, \nu_k\}\big).$$

2.4.2.3 *Improving rotations R_j matrices:*

`rot=V` No change is needed in the R_j matrices

`rot=I` Nothing to be done because $R_j = I$

`rot=E` *(needs iterations through $R^{(r)}$)*: Let $W_j = \frac{n_j}{n} S_j$ and ω_j is the largest eigenvalue of W_j.

(i) Set $r \leftarrow 0$ and $R^{(0)} \leftarrow R$ for $R_1 = \cdots = R_k = R$

(ii) $r \leftarrow r + 1$ and

$$F = \sum_{j=1}^{k} \left(\frac{1}{d_j} D_j^{-1} W_j - \frac{\omega_j}{d_j} D_j^{-1} R^{(r-1)} \right)$$

and $F = U \Lambda V$ its singular value decomposition

(iii) $R^{(r)} \leftarrow VU$

(iv) Go back to ii) if $r <$ `iter.max.R` and $\Delta R^{(\cdot)} >$ `tol.R` or otherwise conclude iterations and finally update

$$R_1 = \cdots = R_k \leftarrow R^{(r)}.$$

2.4.2.4 Go back to Step 2.4.2 if $u <$ `iter.max.dDR` and $\max\{\Delta d, \Delta D, \Delta R\} >$ `tol.dDR`.

2.4.3 Update $\Sigma_j^{(t)} = d_j R_j D_j R_j'$

3. Go back to Step 2 if $t <$ `iter.max.theta` and $\Delta\theta >$ `tol.theta`.
4. *Evaluate the target function* after applying this iterative process, the associate likelihood, depending on the CEM or EM approach, is computed. The parameters yielding the highest value of this target function are returned as the algorithm's output.

Remark 1 As seen in the algorithm, several constants associated with the maximum number of iterations `iter.max.theta`, `iter.max.dDR`, `iter.max.D` and `iter.max.R` have to be specified. With respect to tolerances, we are using `tol.theta`, `tol.dDR`, `tol.D` and `tol.R`. When the monitoring parameters $b^{(h)}$ involving several terms, i.e., $b^{(h)} = \{b_1^{(h)}, \ldots, b_k^{(h)}\}$, the relative changes in the h-th iteration are measured as

$$\Delta b = \max_{j=1,\ldots,k} \left\{ ||\mathsf{vec}(b_j^{(h)}) - \mathsf{vec}(b_j^{(h-1)})||/||\mathsf{vec}(b_j^{(h-1)})|| \right\}.$$

Notice that we use $\mathsf{vec}(\cdot)$ to convert matrices into vectors when needed. The only exception is when monitoring iterated $R^{(h)}$ rotation matrices where we use

$$\Delta R = \frac{\left|p - \text{trace}\left[((R^{(h)})' R^{(h-1)})' (R^{(h)} (R^{(h-1)})')\right]\right|}{p}.$$

Remark 2 Trivial computational short-cuts can be introduced in limit cases, where any of these constants c_{det}, c_{shw} and c_{shb} are chosen equal to 1 or to ∞. In those cases, many iterative steps can be avoided. It is also worthwhile to notice that the most computationally demanding version of the algorithm happens when `rot=E`.

The rationale behind all the steps in this algorithm follows from adaptation of algorithms in the literature. We always try to improve a subset of parameters conditionally on the remaining ones, and, of course fulfilling the required constraints. For instance, the algorithm of Browne and McNicholas (2014) is used in Step 2.4.2.2 and the step in 2.4.2.2 follows from García-Escudero et al. (2020). The"optimal truncation" operator is also used in step 2.4.2.1 to impose the novel constraint in (6).

4 Theoretical results

If we assume that $\{x_1, \ldots, x_n\}$ is a sample from a theoretical probability distribution P, a population version of the constrained parsimonious methodology can be defined and existence and consistency results can be proved whenever finite restriction constants c_{det}, c_{shw} and c_{shb} are considered.

Given $\theta = (\pi_1, \cdots, \pi_k, \mu_1, \cdots, \mu_k, \Sigma_1, \cdots, \Sigma_k)$, we introduce the functions $W_j(x;\theta) = \pi_j \varphi(x;\mu_j,\Sigma_j)$ and $W(x;\theta) = \max\{W_1(x;\theta), \ldots, W_k(x;\theta)\}$, and the set

$$\Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]} = \{\theta : \Sigma_1, \ldots, \Sigma_k \text{ satisfy (4), (5) and (6) for } c_{\text{det}}, c_{\text{shw}} \text{ and } c_{\text{shb}}\}.$$

Theorem 1 provides existence (for both theoretical and sample problem) and consistency result under finite second-order moment conditions.

Theorem 1 *If P is not concentrated at k points and $E_P\|\cdot\|^2 < \infty$*

(a) *then there exists some $\theta \in \Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]}$ such that the maximum of*

$$E_P\left[\log\left[\sum_{j=1}^{k} W_j(\cdot;\theta)\right]\right] \tag{10}$$

is achieved when θ is constrained to be in $\Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]}$.

(b) *then there exists $\theta \in \Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]}$ such that the maximum of*

$$E_P\left[\sum_{j=1}^{k} z_j(\cdot;\theta)\log W_j(\cdot;\theta)\right], \tag{11}$$

with $z_j(x;\theta) = I\{x : W(x;\theta) = W_j(x;\theta)\}$, is achieved when θ is constrained to be in $\Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]}$.

Let us consider an i.i.d. sample $\{x_1, \ldots, x_n\}$ from the underlying distribution P and $P_n = \sum_{i=1}^{n} \delta_{\{x_i\}}$ the associated empirical distribution. The maximizations (1) and (2) under constraints (4), (5) and (6) when $P = P_n$ reduce exactly to the methodology just presented in Sect. 2. Consequently, Theorem 1 also guarantees the existence of the solution of the empirical problem.

Moreover, a consistency result can be proven for the sequence of empirical maximizers toward the maximizer of the theoretical problem if it is unique (up to a relabelling). Let $\theta_n = (\pi_1^n, \cdots, \pi_k^n, \mu_1^n, \cdots, \mu_k^n, \Sigma_1^n, \cdots, \Sigma_k^n) \subset \Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]}$ denote the sequence of empirical maximizers for the sequence of empirical sample distributions $\{P_n\}_{n=1}^{\infty}$ from P. With this notation, Theorem 2 presents the consistency result.

Theorem 2 *Let us assume that P is not concentrated at k points, that $E_P\|\cdot\|^2 < \infty$ and that $\theta_0 \in \Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]}$ is the unique constrained maximizer of (10), resp. (11), up to a relabelling of the parameters corresponding to each of the k components, for P. If $\{\theta_n\}_{n=1}^{\infty}$ is a sequence of empirical maximizers of (1), resp. (2), when $\theta_n \in \Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]}$ then $\theta_n \to \theta_0$ almost surely.*

The proofs of Theorems 1 and 2 derive from similar results in García-Escudero et al. (2020) given that, trivially, we have $\Theta_{[c_{\text{det}},c_{\text{shw}},c_{\text{shb}}]} \subset \Theta_{c_1,c_2}$, for $c_1 = c_{\text{det}}$ and $c_2 = c_{\text{shw}}$, with the same notation for Θ_{c_1,c_2} as in García-Escudero et al. (2020). The results in that previous work were, in turn, based on more general theoretical results in García-Escudero et al. (2015) that had been proved for eigenvalues ratio constraints.

Remark 3 It can be also proven that finite c_{det} and c_{shb} values are just enough for existence and consistence results if P satisfies $E_P\|\cdot\|^2 < \infty$ and P is not concentrated at k hyperplanes.

5 Novel information criteria

In this section, we introduce novel information criteria intended to automatically choose the number of mixture components k and $\text{pars} = [c_{\text{det}}, c_{\text{shw}}, c_{\text{swb}}, \text{rot}]$. The proposal is to choose

$$[\widehat{k}, \widehat{\mathsf{pars}}] = \arg\min_{k,\mathsf{pars}} \mathsf{BIC}[k, \mathsf{pars}],$$

for

$$\mathsf{BIC}[k, \mathsf{pars}] = -2L_k^{\mathsf{pars}} + v_k^{\mathsf{pars}} \log n, \tag{12}$$

where L_k^{pars} is the maximum value achieved in the constrained maximization of (2) under constraints defined by pars, and where v_k^{pars} is a penalty term defined as:

$$\begin{aligned} v_k^{\mathsf{pars}} = & \underbrace{kp}_{\text{means}} + \underbrace{k-1}_{\text{weigths}} + \underbrace{(k-1)\left(1 - \frac{1}{c_{\mathsf{det}}^{1/p}}\right) + 1}_{\text{determinant pars.}} \\ & + \underbrace{(p-1)\left(1 - \frac{1}{c_{\mathsf{shw}}}\right)\left[(k-1)\left(1 - \frac{1}{c_{\mathsf{shb}}}\right) + 1\right]}_{\text{shape pars.}} \\ & + \underbrace{k(\mathsf{rot})\frac{p(p-1)}{2}}_{\text{rotation pars.}}, \end{aligned}$$

with

$$k(\mathsf{rot}) = \begin{cases} 0 \text{ if } \texttt{rot=I} \\ 1 \text{ if } \texttt{rot=E} \\ k \text{ if } \texttt{rot=V} \end{cases}.$$

Notice that larger values of c_{det}, c_{shw} and c_{swb} yield less restricted Σ_j scatter matrices, given that more complex models are allowed to be fitted. It is important to note that "model complexity" here does not necessarily correspond to an increased number of parameters, and so "smaller complexity" does not necessarily mean that there are fewer parameters to be interpreted. We also consider a source of complexity for the Σ_j matrices which depends on the allowed rotations through rot.

The proposal follows a similar philosophy so that introduced in Cerioli et al. (2018). The same arguments in terms of "relative volumes" (as those in Theorem 3.1 in that paper) have been taken into account to derive the previous expression for v_k^{pars}. It is easy to see that v_k^{pars} exactly coincides with the number of free parameters for the classical 14 parametrizations which appear as limit cases (restriction constants equal to 1 or tending to ∞) reviewed in Table 1. Moreover, the proposal also coincides with the BIC proposal for the "deter-and-shape" constraints previously introduced in García-Escudero et al. (2020) when the constraint (6) is removed by taking $c_{\mathsf{shb}} \to \infty$. In that case, the contribution to v_k^{pars} due to parameters associated to "shape elements" is just

$$k(p-1)\left(1 - \frac{1}{c_{\mathsf{shw}}}\right)$$

and $k(\mathsf{rot}) = k$ (no constraints on the rotation matrix).

In this work, we have just focused on the BIC proposal, which is an extension of the MIX-MIX approach in Cerioli et al. (2018). A classification likelihood approach can be also applied by replacing the target function (2) with (1) to define extensions of the MIX-CLA and CLA-CLA approaches (in the spirit of the ICL criterion in Biernacki et al. (2000)).

The minimization of the criterion (12) over all the possible combinations of k and pars is not an easy task. In order to circumvent this problem, we just consider powers of 2 for the restriction constants and propose the following procedure:

1. Fix K as an upper bound for the maximum number of components and C such that 2^C is large enough that the constraints enforced are not very strict.
2. We first obtain

$$\begin{aligned} & [k^*, c_{\mathsf{det}}^*, c_{\mathsf{shw}}^*, c_{\mathsf{shb}}^*, \mathsf{rot}^*] \\ & = \underset{(k,\mathsf{pars}) \in \{1,\ldots,K\} \times \{1, 2^{C-1}\}^3 \times \{\mathsf{I},\mathsf{E},\mathsf{V}\}}{\arg\max} \mathsf{BIC}[k, \mathsf{pars}]. \end{aligned} \tag{13}$$

   This implies applying the proposed methodology for $K \times 14$ slightly constrained models (and guaranteeing that numerical issues due to singularities are avoided), in correspondence with all the feasible models in Table 1. These $K \times 14$ models need also to be fitted initially as happens with other BIC proposals for parsimonious models. All the other intermediate models (to evaluate) are included within those model fitted with restriction constants equal to 2^{C-1}.
   This maximization will directly provide our final choice for the number of clusters k^* and for the chosen rotation rot^*. Moreover, it also returns our final choices for c_{det}, c_{shw} and c_{shb} whenever any of these c_{det}^*, c_{shw}^* and c_{shb}^* take the value 1.
3. Constants c_{det}, c_{shw} and c_{shb} need to be refined because just upper bounds are initially allowed in Step 1. To perform these refinements, let us obtain

$$\begin{aligned} & [c_{\mathsf{det}}^{**}, c_{\mathsf{shw}}^{**}, c_{\mathsf{shb}}^{**}] \\ & = \underset{(c_{\mathsf{det}}, c_{\mathsf{shw}}, c_{\mathsf{shb}}) \in \mathcal{C}}{\arg\max} \mathsf{BIC}[k = k^*, c_{\mathsf{det}}, c_{\mathsf{shw}}, c_{\mathsf{shb}}, \mathsf{rot} \\ & = \mathsf{rot}^*], \end{aligned} \tag{14}$$

   where

$$\begin{aligned} \mathcal{C} = & \left\{2^0, \ldots, \min\{c_{\mathsf{det}}^*, 2^{C-1}\}\right\} \\ & \times \left\{2^0, \ldots, \min\{c_{\mathsf{shw}}^*, 2^{C-1}\}\right\} \\ & \times \left\{2^0, \ldots, \min\{c_{\mathsf{shb}}^*, (c_{\mathsf{shw}}^*)^{(p-1)/p}, 2^{C-1}\}\right\}. \end{aligned}$$

   We are taking advantage of the initial information about parameters resulting from Step 1 and applying Lemma

1 to reduce notably the number of configurations to be tested.

4. After this process, we finally consider $[k^*, c^{**}_{\mathsf{det}}, c^{**}_{\mathsf{shw}}, c^{**}_{\mathsf{shb}}, \mathsf{rot}^*]$ as a suggestion for $[\widehat{k}, \widehat{\mathsf{pars}}]$.

6 Simulation study

In this section, we show the advantage of our constrained proposals using simulated datasets. We first consider examples where the number of clusters k is assumed to be known (Sect. 6.1). We then show the effectiveness of the novel information criteria for choosing models (Sect. 6.2). As mentioned in the introduction section, we do not claim that the well-established and widely applied proposals considered for comparison in this section are useless, since they clearly have long proven their validity in many scenarios and real data applications. We show examples that highlight the usefulness of the proposed methodology in achieving extra stability.

6.1 Comparison for a fixed number of components

We compare first the performance of the proposed methodology with respect to well-known implementations of the 14 parsimonious model-based clustering methods when assuming a known number of components and the parametrization needed (the `VVV` parametrization in all cases).

The first example is based on $k = 3$ normally distributed components, where the first two coordinates $(X_1, X_2)'$ of these components are generated from bivariate normals with mean parameters equal to $(0, 0)'$, $(2, 6)'$ and $(6, 0)'$, and the covariances matrices are

$$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix},$$

respectively. A third independent component X_3 is generated from a univariate normal with 0 mean and variance 100, i.e., $X_3 \sim N(0, 100)$. This simulation scheme is denoted as "lower p" case, but we also add an independent fourth coordinate $X_4 \sim N(0, 100)$ to generate the "higher p" case. In the simulation study, we take random samples with $n_1 = 50$, $n_2 = 20$ and $n_3 = 20$ from each component in the "lower n" case, and doubled sizes $n_1 = 100$, $n_2 = 40$ and $n_3 = 40$ in the "higher n" case.

To explore the effect of the restriction constants, always applying the `rot=V` case, a three letters notation is used when summarizing the simulation results. The first letter corresponds to the restriction constant chosen in c_{det}, the second letter to the constant in c_{shw} and the third letter to the constant c_{shb}. In these three letters, we use the letter "C" when the constant defining the constraint is exactly chosen equal to the maximal ratio for this constant computed from the true model generating the data. On the other hand, we use letter "U" when this constant is chosen so high that the procedure is (almost unrestricted) by fixing it to be equal to 10^{10} (just to avoid very extreme numerical issues). Letter "D" is used when we double the value for the constant fixed in C, letter "E" when we multiply the constant in C by 2^2, letter "F" when we multiply by 2^4 and letter "G" when we multiply by 2^8.

We always consider the case $k = 3$ and `rot=V` and apply the algorithm in Sect. 3 with `nstart=` 1000 and `iter.max=` 100. We have included in the simulation study two particular cases, namely, the CCU (that corresponds to the "deter-and-shape" proposal with a large $c_{\mathsf{shb}} = 10^{10}$), and the CUC (corresponding to a large $c_{\mathsf{shw}} = 10^{10}$).

The results obtained are compared with those which come from applying the `mixture` (Browne et al. 2018) and the `Mclust` (Scrucca et al. 2016) packages in R, when using the `VVV` parametrization in both cases and when searching for $k = 3$ components. We consider two available options for initializing based on the k-means method ("mix_km") and on 1000 random starts ("mix_rs") when applying the `mixture` package.

Figure 2 shows the value of the ARI-Adjusted Rand Indexes (Hubert and Arabie 1985) for the obtained partitions, with respect to the true classification, on the same 100 simulated data sets for each of the four possible scenarios depending on the two possible dimensions and two possible sample sizes.

In order to see the effect on the estimation of the parameters, the Euclidean distances when estimating all the true mean vectors $\mu_1 = (0, 0, 0)'$, $\mu_2 = (2, 6, 0)'$ and $\mu_3 = (6, 0, 0)'$ (lower p) and $\mu_1 = (0, 0, 0, 0)'$, $\mu_2 = (2, 6, 0, 0)'$ and $\mu_3 = (6, 0, 0, 0)'$ (higher p) are shown in Fig. 3. Relabelling has been applied to match estimators with the estimated location parameters.

Figure 4 finally shows the maximum values achieved by the mixture likelihood obtained when maximizing the defining target likelihood function in (2).

As expected, the "higher n" cases exhibit clearly better performances. We can also see in Figs. 2 and 3 that the constrained approaches seem to provide higher ARI values and lower estimation errors than their competitors, and that those including letters C and D exhibit the most accurate results among them. We can also see that the least constrained approaches (including letters E, F, G and U) do not provide good results (because stability seems to be lost when increasing the restriction constants), but values of these constants greater than the true ones in C are in general not excessively detrimental. On the other hand, the unconstrained case UUU gives the worst performance. We can also see similar unsatisfactory behavior as in the UUU case when applying the `VVV` models with the `mixture` and the `Mclust` packages. These approaches also considered theoretically a fully unre-

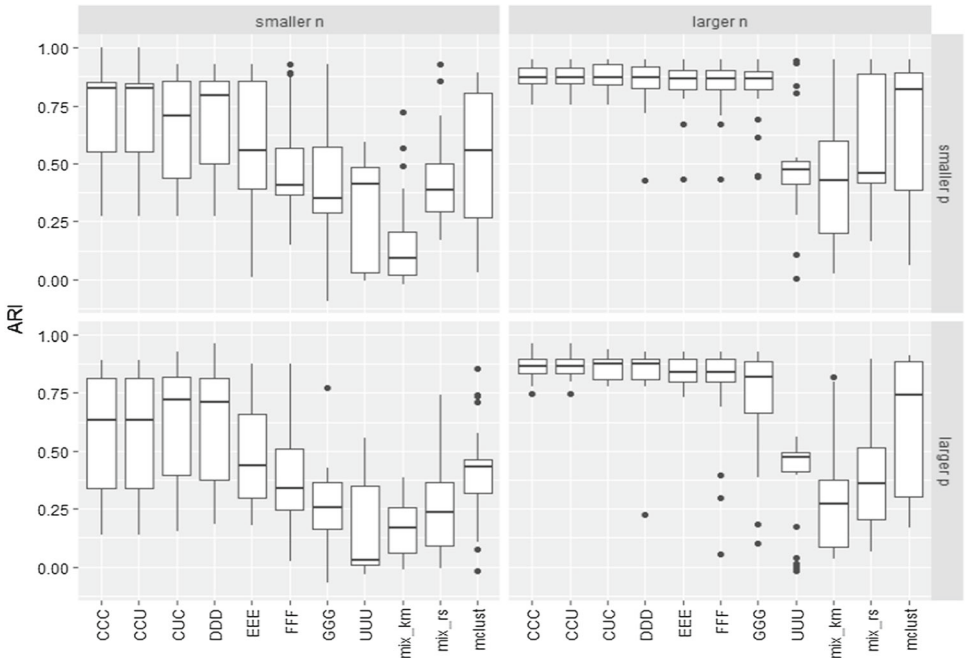

Fig. 2 ARI values in the comparative study for the $k = 3$ components example

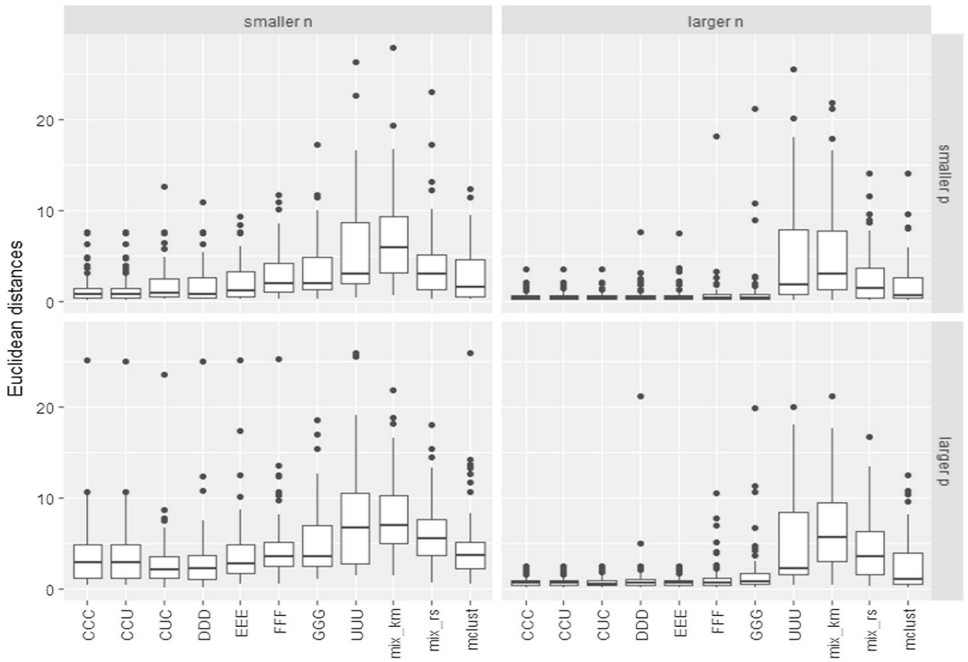

Fig. 3 Sum of the Euclidean distances when estimating the true mean vectors in the comparative study for the $k = 3$ components example

stricted approach as in the UUU case. However, despite this lack of constraints in the scatter matrices, we observe that the performance of `mixture` and `Mclust` depend heavily on the initializing procedure. In this regard, we can see that the initialization based on k-means is not satisfactory due to the particular data generation scheme, which is clearly not appropriate for k-means.

Even though "mix_rs" is also based on 1000 random initializations, as in our constrained proposals, we can see in Fig. 4 that it does not reach values in the target likelihood so high as those obtained in the UUU case (that could perhaps be even greater if these constants were chosen at values greater than 10^{10}). Therefore, the type of initializations considered in Step 1 of our algorithm seems to better explore the parametric space than the initializations in "mix_rs". The same happens with the initializations provided by k-means or by the initializing procedure based on hierarchical model-based clustering applied by `Mclust`. The initializations can be useful to avoid spurious solutions, but it is also important to note that they are not striving to maximize the target likelihood

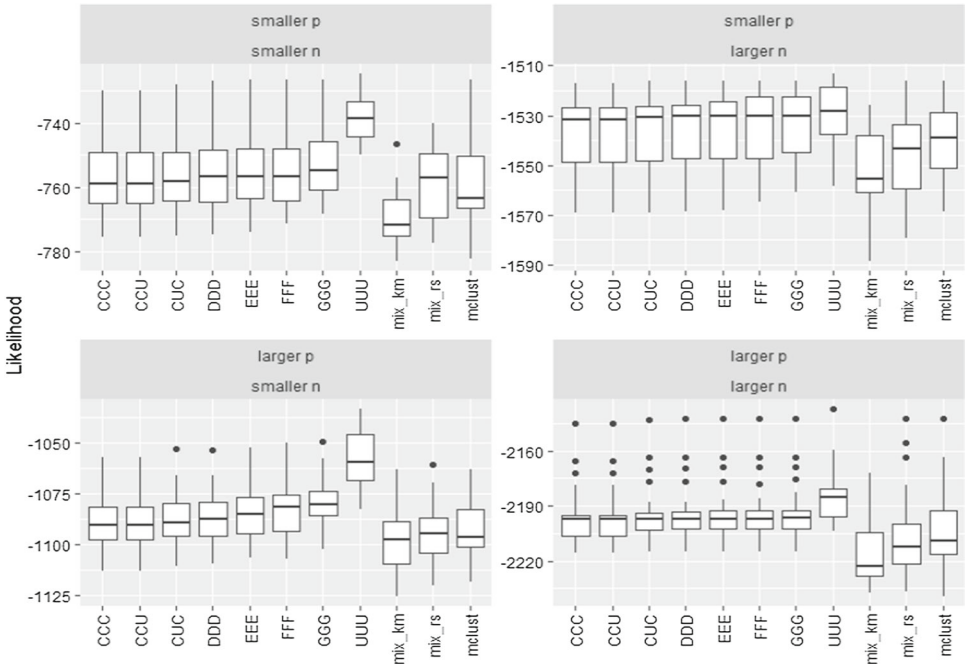

Fig. 4 Maximum values achieved when maximizing the target likelihood function for the $k = 3$ components example

function and they clearly influence the performance of the methodology. Figure 4 also shows how the target likelihoods steadily increase when increasing the values of the restriction constants (C, D, E, F, G and U) and this could serve to understand the degree of stability provided by constraints and help us to achieve smooth transitions between models.

We briefly present another example with a higher $k = 6$ number of components. The example starts from a bivariate mixture of six spherical normally distributed components with the same scatter and component sizes equal to $n_1 = 23$, $n_2 = 36$, $n_3 = 93$, $n_4 = 38$, $n_5 = 123$ and $n_6 = 12$ with the following mean vectors: $(-4.5, 3.6)'$, $(0.40, 3.6)'$, $(-4.4, -1)'$, $(9.2, -1)'$, $(0.4, -1)'$, and $(9.2, 3.6)'$. We add a third dimension by using an independent variable $X_3 \sim N(0, 100)$, which makes these clusters become elongated.. A simulation study, completely analogous to the one previously described is considered. Figure 5 provides the Euclidean distances when estimating the corresponding six means of the components individually.

We can see that the constrained approach seems to provide better results also in this $k = 6$ example.

6.2 Comparison when choosing number of components and models

We also compare the performance of the novel information criteria introduced in Sect. 5 with respect to the BIC procedures resulting from the application of the `mixture` package (Browne et al. 2018). With this aim in mind, we simulate 100 samples of size $n = 200$ in dimension $p = 10$ for each of the 14 classical parsimonious models. To be more precise, each sample is generated from a $k = 3$ components mixture where μ_1, μ_2 and μ_3 and Σ_1, Σ_2 and Σ_3 are randomly generated parameters in such a way that the covariance matrices satisfy the specific model constraints and also a prefixed overlap rate equal to 0.05. That overlap is achieved by applying the extension of the `MixSim` method of Maitra and Melnykov (2010) given in Riani et al. (2015). Given two clusters j and l obtained from normal densities $\phi(\cdot; \mu_j, \Sigma_j)$ and $\phi(\cdot; \mu_l, \Sigma_l)$, with probabilities of occurrence π_j and π_l, the overlap between groups j and l is defined as the sum of the two misclassification probabilities $w_{jl} = w_{j|l} + w_{l|j}$ where $w_{j|l} = P[\pi_l \phi(X; \mu_l, \Sigma_l) < \pi_j \phi(X; \mu_j, \Sigma_j)]$. The average overlap is the sum of the off-diagonal elements of the matrix of the misclassification probabilities $w_{j|l}$ divided by $k(k-1)/2$. Note that when we say that the covariance matrices satisfy the model constraints, we mean that we ensure that the Σ_1, Σ_2 and Σ_3 matrices do exactly satisfy the constrained models with the values of c_{det}, c_{shw} and c_{swb} as in Table 1 but replacing the values of "∞" in that table by values equal to 100 in the case of c_{det} or c_{shw} and by a value equal to 10 in the case of c_{swb}. The mixture components weights are always $\pi_1 = \pi_2 = \pi_3 = 1/3$.

Figure 6 shows the ARI indexes between the true partition and the partition obtained from the fitted mixture suggested by the BIC-type information criterion. In this figure, we use the notation "new" for the results associated with the new proposed methodology and "mix_km" for those with the `mixture` package when initialized with k-means and "mix_rs" when initialized by using random starts (the same number of random starts `nstart` as in "new" are considered).

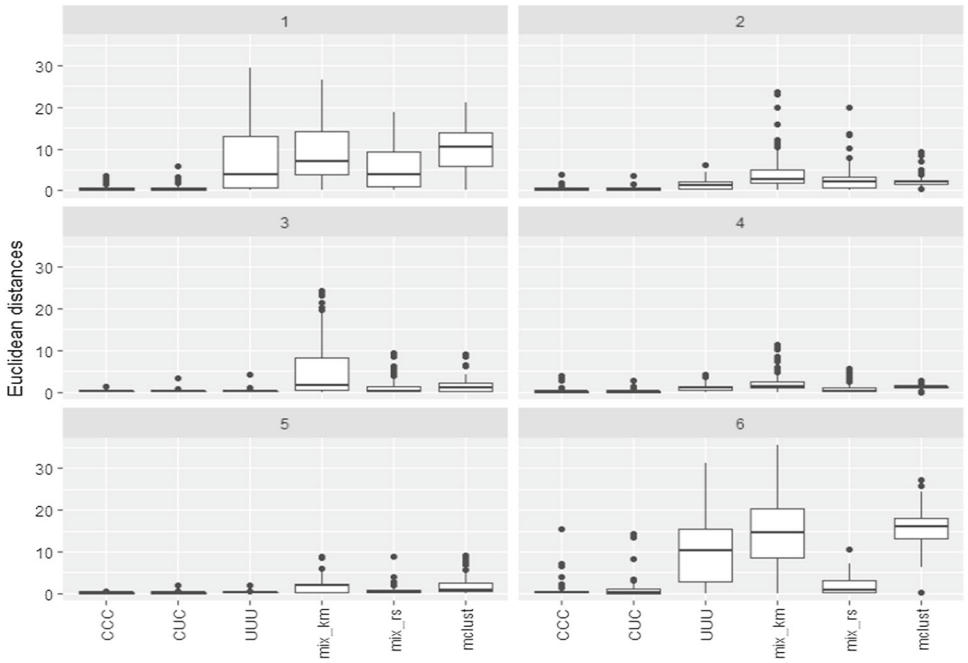

Fig. 5 Euclidean distances when estimating the true mean vectors in the comparative study for the $k = 6$ components example. Each panel corresponds to one of the components

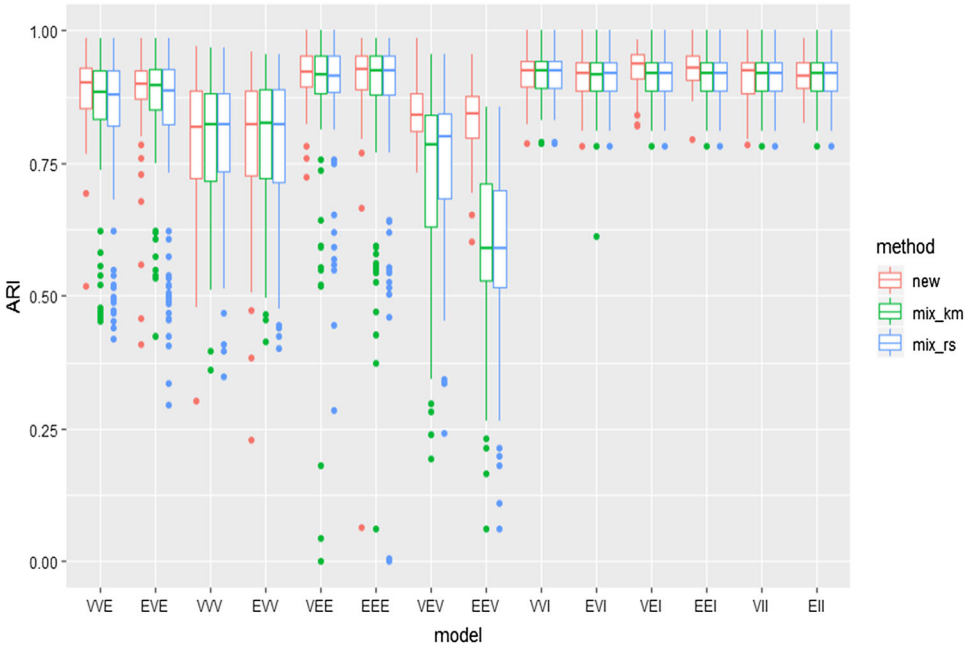

Fig. 6 ARI values for the simulated data sets when applying the novel information criteria and the BIC procedure in the `mixture` package

Figure 7 shows the proportion of times that the true number of components, $k = 3$, is determined by the BIC-type information criteria.

We can see in these two figures that the BIC methodology in the `mixture` package is reasonably able to recover the right number of components and the true data generating mechanism. However, the comparison is clearer in Fig. 7 when looking at the proportion of times that the true number of components is detected. We see that better results are obtained when applying the new constrained approaches. Of course, those differences are not so noticeable for the most constrained (`VVI`, `EVI`, …, `EII`) models, where no great advantages can be achieved by restricting even more.

The improvement is more clearly seen in Fig. 7 than in Fig. 6, perhaps because spurious components, made up of few observations, do not significantly modify the ARI, even though they change the number of components detected. This wrong determination of the number of components may of course be problematic, when interpreting results. Note also that constrained approaches seem to avoid partitions exhibiting very low ARI values (outliers outside whiskers in Fig. 6).

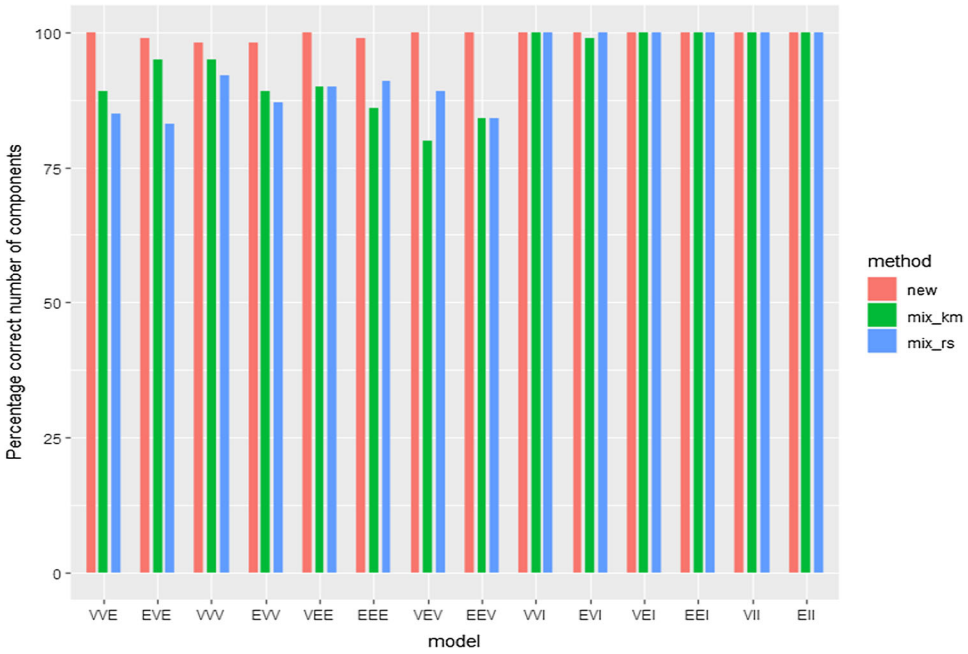

Fig. 7 Proportion of times that the true number of components $k = 3$ is determined by the information criteria

7 Real data example: COVID data

The example is inspired by the analysis of a real data set on the SARS-CoV-2 symptoms kindly provided to us by the ASL3 Genovese Hospital. Measurements on six variables x_1 = "heart rate (the number of beats the heart per minute)", x_2 = "Oxygen Uptake Efficiency Slope (index of functional reserve derived from the logarithmic relation between oxygen uptake and minute ventilation during incremental exercise)," x_3 = "watts (reached by the patient during the stress test on a cycle ergometer (stationary bike) at the aerobic threshold, that is, when the patient 'begins to struggle')," x_4 = "watts peak (watts reached at maximum effort (during exercise test on exercise bike)," x_5 = "value of the maximum repetition (maximum force of muscle contraction of the quadriceps femoris of the dominant limb expressed in kg)" and x_6 = "previous variable corrected on the subject (in relation to the patient's weight)" on 79 COVID patients and 77 non-COVID ones. Figure 8a shows the (supposedly) true classification, as provided by the doctors. Data have been collected by "Post-COVID Outpatient Rehabilitation Center ASL3 Liguria Region Health System" and approved by the Ethics Committee of Liguria region (Italy).

We will apply the (unsupervised) constrained model-based clustering approach to see if something close to the doctor's classification partition is achieved. We will use the modified BIC approach described in Sect. 5 to determine the underlying number of clusters and the set of constraints to be imposed.

The proposed BIC approach described in Sect. 5 is applied with $K = 5$ and $C = 8$ (i.e., $2^{C-1} = 128$). In the maximization of (13), after fitting 5×14 models, we obtain $k^* = 2$, $c^*_{\mathsf{det}} = 1$, $c^*_{\mathsf{shw}} = 128$, $c^*_{\mathsf{shb}} = 1$ and rot^*=E. Afterwards, we perform the maximization in (14) where $\mathcal{C} = 1 \times \{2^0, 2^1, \ldots, 2^{7-1}\} \times 1$. This means that we need to obtain

$$c^{**}_{\mathsf{shw}} = \underset{c_{\mathsf{shw}} \in \{1,2,\ldots,128\}}{\arg\max} \mathsf{BIC}[k = 2, c_{\mathsf{det}} = 1, c_{\mathsf{shw}}, c_{\mathsf{shb}} = 1, \mathsf{rot} = \mathsf{E}],$$

(as we directly have $c^{**}_{\mathsf{det}} = 1$ and $c^{**}_{\mathsf{shb}} = 1$). Our BIC proposal suggests $[\widehat{k}, \widehat{\mathsf{pars}}] = [2, 1, 128, 1, \mathsf{E}]$. This is a quite constrained solution where only the "within" components shape elements are left notably unrestricted (the value of c_{shw} is such that only a slightly constrained ratio is considered). The associated partition is shown in Fig. 8b and exhibits an ARI index with respect to the "true" doctor's classification equal to 0.5891.

On the other hand, the BIC approach implemented through function `gpcm()` in the `mixture` package in R (Browne et al. 2018) suggests $k = 2$ but the `EEE` parameterization results in an ARI equal to 0.0117 with respect to the doctor's suggested partition. This result is obtained when considering a k-means type initialization but the results do not seem to improve when considering random initializations. The results, for this particular data set, do not improve when we apply the BIC criterion provided by the `Mclust` package (Scrucca et al. 2016) that only suggests one $k = 1$ component for this data set. The `VEE` parameterization is suggested when considering `Mclust`'s BIC criterion but restricted to models with $k = 2$, which yields a ARI=0.0414 that is notably

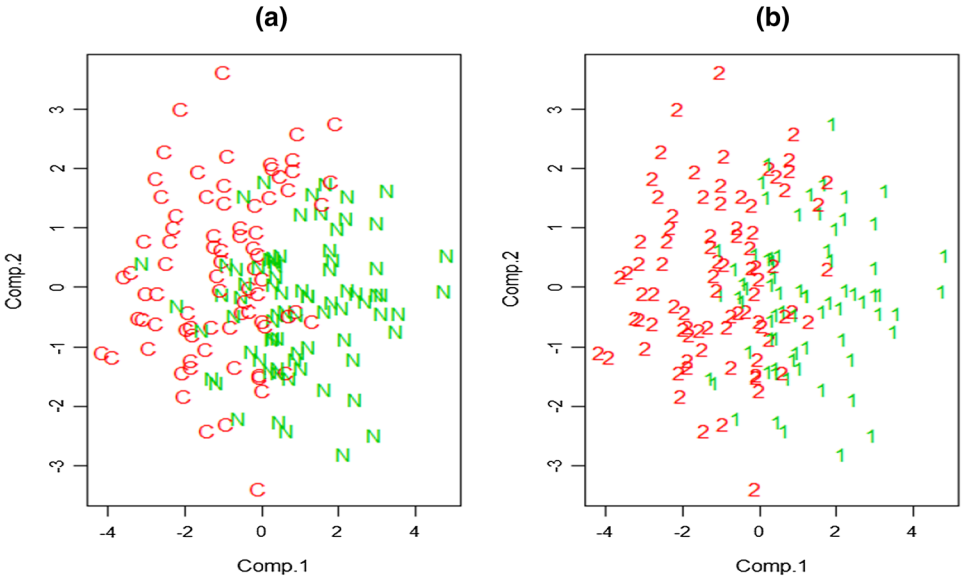

Fig. 8 **a** Doctor-based "true" classification for the COVID data set with COVID patients denoted with `C` symbols and non-COVID by `N` symbols with observations represented in the first two principal components. **b** Clustering results of the constrained parsimonious model-based clustering proposal with parameters chosen from the new BIC procedure

smaller that the 0.5891 achieved when applying the proposed methodology with the new BIC proposal.

8 Conclusions and further directions

We have introduced a new methodology for constrained parsimonious model-based clustering that depends on three restriction constants c_{det}, c_{shw} and c_{shb} and on fixing a particular type `rot` of rotations. The methodology provides a smooth transition among the well-known 14 parsimonious models that are commonly applied in model-based clustering when assuming normality for the components. The proposed constraints result in mathematically well-defined problems and provide extra control on the covariance matrices of the fitted components. Novel information criteria have been introduced to help the user in providing sensible choices for all the tuning decisions.

There are many open research lines related to this new approach. For instance, dealing with computational aspects could still be needed to speed up the procedures. Although MATLAB code for its practical application is now available, we are developing a more dedicated and easy to apply implementation within the `FSDA` MATLAB toolbox (Riani et al. 2012). This implementation will hopefully include more elaborate graphical and numerical tools in helping to determine and explore the solutions obtained when moving all the involved parameters in the spirit of Cerioli et al. (2018). With that aim, stability and ARI distances among partitions could be taken into account in order to derive a reduced (and ranked) list of sensible partitions and also graphical summaries as the "car-bike plots." The methodology can be also adapted to include "trimming" to introduce new robust model-based clustering approaches.

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

Appendix A: Proof of Lemma 1

Let us consider $\xi_{j,l} = \log \gamma_{jl}$ such that

$$\xi_{j,1} \geq \cdots \geq \xi_{j,l} \geq \cdots \geq \xi_{j,p},$$

as in (7), under the constraint

$$\sum_{l=1}^{p} \xi_{j,l} = 0, \tag{15}$$

(given that $\prod_{l=1}^{p} \gamma_{jl} = 1$). Constraints (5), after log-transformation, reduce to $\xi_{j,l} - \xi_{j,l'} \leq \log c_{\mathsf{shw}}$ for any $l, l' \in \{1, \ldots, p\}$ for a fixed j.

It is not difficult to see that the maximal possible difference is achieved with repeated $\xi_{j,l}$ values and setting that maximal

difference equal to $\log c_{\mathsf{shw}}$. In that case, the most extreme difference is for j and j' when

$$\xi_{j,1} = \cdots = \xi_{j,l} = \xi_{j,p} + \log c_{\mathsf{shw}} \text{ and } \xi_{j,l+1} = \cdots = \xi_{j,p}$$

and

$$\xi_{j',1} = \cdots = \xi_{j',l-1} = \xi_{j',p} + \log c_{\mathsf{shw}} \text{ and}$$
$$\xi_{j',l} = \cdots = \xi_{j',p}.$$

By taking into account the zero sum condition (15), we have $0 = p\xi_{j,p} + l \log c_{\mathsf{shw}}$ and $0 = p\xi_{j',p} + (l-1)\log c_{\mathsf{shw}}$, for these two configurations. Therefore,

$$\begin{aligned} p(\xi_{j,l} - \xi_{j',l}) &= (p\xi_{j,l} - (p\xi_{j,p} + l \log c_{\mathsf{shw}})) \\ &\quad -(p\xi_{j',l} - (p\xi_{j',p} + (l-1)\log c_{\mathsf{shw}})) \\ &= (p-1)\log c_{\mathsf{shw}}, \end{aligned}$$

where we have used that $\xi_{j,l} - \xi_{j,p} = \log c_{\mathsf{shw}}$ and that $\xi_{j',l} - \xi_{j',p} = 0$. Consequently, $\xi_{j,l} - \xi_{j',l} = \frac{p-1}{p} \log c_{\mathsf{shw}}$ for that most extreme possible difference and so (8) is just proven after exponentiation.

Appendix B: "Optimal truncation" operator

For sake of completeness, we review the "optimal truncation" procedure (Fritz et al. 2013) that has been extensively used in the algorithm in Sect. 3.

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

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

Given $\{n_j\}_{j=1}^J \in \mathbb{N}^J$ and $\{d_{j1}, \ldots, d_{jL}\}_{j=1}^J \in [0,\infty)^{J\times L}$, we define that operator as

$$\mathsf{opt.trunc}_c\big(\{n_j\}_{j=1}^J; \{d_{j1}, \ldots, d_{jL}\}_{j=1}^J\big),$$

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

$$m_{\text{opt}} = \arg\min_m \sum_{j=1}^J n_j \sum_{l=1}^L \left(\log\left(d_{jl}^m\right) + \frac{d_{jl}}{d_{jl}^m} \right). \tag{16}$$

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 (Fritz et al. 2013) of (16) through a procedure which can be fully vectorized.

References

Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics **49**, 803–821 (1993)

Biernacki, C., Celeux, G., Govaert, G.: Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans. Pattern. Anal. Mach. Intell. **22**, 719–725 (2000)

Biernacki, C., Celeux, G., Govaert, G.: Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. Comput. Stat. Data Anal. **41**, 561–575 (2003)

Browne, R., McNicholas, P.: Estimating common principal components in high dimensions. Adv. Data. Anal. Classif. **8**, 217–226 (2014)

Browne, R.P., ElSherbiny, A., McNicholas, P.D.: mixture: mixture models for clustering and classification. R Package Version **1**, 5 (2018)

Celeux, G., Govaert, G.: Gaussian parsimonious clustering models. Pattern Recognit. **28**, 781–793 (1995)

Cerioli, A., García-Escudero, L.A., Mayo-Iscar, A., Riani, M.: Finding the number of normal groups in model-based clustering via constrained likelihoods. J. Comput. Graph. Stat. **27**, 404–416 (2018)

Day, N.: Estimating the components of a mixture of normal distributions. Biometrika **56**, 463–474 (1969)

Fritz, H., García-Escudero, L.A., Mayo-Iscar, A.: A fast algorithm for robust constrained clustering. Comput. Stat. Data Anal. **61**, 124–136 (2013)

Gallegos, M.T., Ritter, G.: Probabilistic clustering via pareto solutions and significance tests. Adv. Data Anal. Classif. **12**, 179–202 (2018)

García-Escudero, L.A., Gordaliza, A., Matrán, C., Mayo-Iscar, A.: Avoiding spurious local maximizers in mixture modeling. Stat. Comput. **25**, 619–633 (2015)

García-Escudero, L.A., Gordaliza, A., Greselin, F., Ingrassia, S., Mayo-Iscar, A.: Eigenvalues and constraints in mixture modeling: geometric and computational issues. Adv. Data Anal. Classif. **12**, 203–233 (2018)

García-Escudero, L.A., Mayo-Iscar, A., Riani, M.: Model-based clustering with determinant-and-shape constraint. Stat. Comput. **25**, 1–18 (2020)

Hathaway, R.: A constrained formulation of maximum likelihood estimation for normal mixture distributions. Ann. Stat. **13**, 795–800 (1985)

Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)

Kiefer, J., Wolfowitz, J.: Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. Ann. Math. Stat. **27**, 887–906 (1956)

Maitra, R., Melnykov, V.: Simulating data to study performance of finite mixture modeling and clustering algorithms. J. Comput. Graph. Stat. **19**, 354–376 (2010)

McLachlan, G., Peel, D.: Finite Mixture Models. Wiley Series in Probability and Statistics, New York (2000)

Meng, X., Rubin, D.: Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika **80**, 267–278 (1993)

Riani, M., Perrotta, D., Torti, F.: FSDA: a Matlab toolbox for robust analysis and interactive data exploration. Chemometr. Intell. Lab. Syst. **116**, 17–32 (2012)

Riani, M., Cerioli, A., Perrotta, D., Torti, F.: Simulating mixtures of multivariate data with fixed cluster overlap in FSDA library. Adv. Data Anal. Classif. **9**, 2015 (2015)

Scrucca, L., Fop, M., Murphy, T.B., Raftery, A.E.: mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. R J. **8**(1), 289–317 (2016)