Adv Data Anal Classif (2015) 9:371–394
DOI 10.1007/s11634-015-0209-7

REGULAR ARTICLE

Maximum likelihood estimation of Gaussian mixture models without matrix operations

Hien D. Nguyen[1] · **Geoffrey J. McLachlan**[1]

Received: 28 September 2014 / Revised: 3 May 2015 / Accepted: 22 May 2015 /
Published online: 5 June 2015

Abstract The Gaussian mixture model (GMM) is a popular tool for multivariate analysis, in particular, cluster analysis. The expectation–maximization (EM) algorithm is generally used to perform maximum likelihood (ML) estimation for GMMs due to the M-step existing in closed form and its desirable numerical properties, such as monotonicity. However, the EM algorithm has been criticized as being slow to converge and thus computationally expensive in some situations. In this article, we introduce the linear regression characterization (LRC) of the GMM. We show that the parameters of an LRC of the GMM can be mapped back to the natural parameters, and that a minorization–maximization (MM) algorithm can be constructed, which retains the desirable numerical properties of the EM algorithm, without the use of matrix operations. We prove that the ML estimators of the LRC parameters are consistent and asymptotically normal, like their natural counterparts. Furthermore, we show that the LRC allows for simple handling of singularities in the ML estimation of GMMs. Using numerical simulations in the R programming environment, we then demonstrate that the MM algorithm can be faster than the EM algorithm in various large data situations, where sample sizes range in the tens to hundreds of thousands and for estimating models with up to 16 mixture components on multivariate data with up to 16 variables.

Keywords Gaussian mixture model · Minorization–maximization algorithm · matrix operation-free · Linear Regression

Mathematics Subject Classification 65C60 · 62E10

✉ Hien D. Nguyen
h.nguyen7@uq.edu.au

[1] Department of Mathematics, School of Mathematics and Physics, University of Queensland, St. Lucia 4072, Australia

1 Introduction

The Gaussian mixture model (GMM) is a ubiquitous tool in the domain of model-based cluster analysis; for instance, see McLachlan and Basford (1988) and Chapter 3 of McLachlan and Peel (2000) for a statistical perspective, and Chapter 9 of Bishop (2006) and Chapter 10 of Duda et al. (2001) for a machine learning point of view. Furthermore, discussions of applications of GMMs and comparisons of GMMs to other cluster analysis methods can be found in Chapter 8 of Clarke et al. (2009), Section 14.3 of Hastie et al. (2009), and Section 9.3 of Ripley (1996), as well as Hartigan (1985), Jain et al. (1999), and Jain (2010). The GMM framework can be defined as follows.

Let $Z \in \{1, \ldots, g\}$ be a categorical random variable such that

$$\mathbb{P}(Z = i) = \pi_i > 0$$

for $i = 1, \ldots, g-1$ and $\mathbb{P}(Z = g) = 1 - \sum_{i=1}^{g-1} \pi_i = \pi_g$, and let $\boldsymbol{X} \in \mathbb{R}^d$ be such that $\boldsymbol{X}|Z = i$ has density $\phi_d\left(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)$, where

$$\phi_d(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{d}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right] \tag{1}$$

is a d-dimensional multivariate Gaussian density function with mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and positive-definite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$. Here, the superscript T represents matrix/vector transposition.

If we suppose that Z is unobserved (i.e. Z is a latent variable), then the density of $\boldsymbol{X}$ can be written as

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{i=1}^{g} \pi_i \phi_d\left(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right), \tag{2}$$

where $\boldsymbol{\theta} = (\boldsymbol{\pi}^T, \boldsymbol{\mu}_1^T, \ldots, \boldsymbol{\mu}_g^T, \text{vech}^T(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^T(\boldsymbol{\Sigma}_g))^T$ is the model parameter vector and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{g-1})^T$. Densities of form (2) are known as GMMs, and we refer to each $\phi_d(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ as component densities.

Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be a random sample from a population characterized by density $f(\boldsymbol{x}; \boldsymbol{\theta}^0)$, where the parameter vector $\boldsymbol{\theta}^0$ is unknown. In such cases, the estimation of $\boldsymbol{\theta}^0$ is required for further inference regarding the data. Given an observed sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, such estimation can be conducted via maximum likelihood (ML) estimation to yield the ML estimator $\hat{\boldsymbol{\theta}}_n$, where $\hat{\boldsymbol{\theta}}_n$ is an appropriate local maximizer of the likelihood function $\mathcal{L}_n(\boldsymbol{\theta}) = \prod_{j=1}^{n} f(\boldsymbol{x}_j; \boldsymbol{\theta})$.

Due to the summation form of (2), the computation of $\hat{\boldsymbol{\theta}}_n$ cannot be conducted in closed form. However, since its introduction by Dempster et al. (1977), the expectation–maximization (EM) algorithm has provided a stable and monotonic iterative method for computing the ML estimator; see Section 3.2 of McLachlan and Peel (2000) for details. Although effective, the EM algorithm for GMM is not without some criticisms; for example, it is known that the convergence of EM algorithms can be very

slow, and may be computationally expensive in some applications; see Chapters 3 and 4 of McLachlan and Krishnan (2008) for details regarding theses issues.

To remedy the aforementioned issues, there have been broad developments in constructing modifications of the GMM EM algorithm, as well as suggestions of alternative methodologies for estimation. For example, Andrews and McNicholas (2013), Botev and Kroese (2004), Ingrassia (1991), and Pernkopf and Bouchaffra (2005) considered the use of metaheuristic algorithms; and Andrews and McNicholas (2013), Celeux and Govaert (1992), Ganesalingam and McLachlan (1980), and McLachlan (1982) considered alternatives to the likelihood criterion [see Chapter 12 of McLachlan and Peel (2000) for a literature review of other developments in this direction]. Each of the aforementioned methods have been shown to improve upon the performance of the EM algorithm via simulation studies, although there are no theoretical results to show that any of the methods uniformly out performs the EM algorithm. Furthermore, each of the methods either require randomization, which relinquishes the monotonicity of the EM algorithm, or replacement of the likelihood criterion, which abandons the statistical properties of the ML estimators.

In this article, we devise a new algorithm for the estimation of GMMs that retains the monotonicity properties of the EM algorithm whilst not utilizing matrix operations. Our approach extends from the following characterization of the multivariate Gaussian density function.

Consider the following decomposition of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, from (1), in which

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_{1:d-1} \\ \mu_d \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{1:d-1,1:d-1} & \boldsymbol{\Sigma}_{d,1:d-1}^T \\ \boldsymbol{\Sigma}_{d,1:d-1} & \Sigma_{d,d} \end{bmatrix}, \tag{3}$$

where $\boldsymbol{\mu}_{1:k}$ contains the first k elements of $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}_{1:k,1:k}$ is the submatrix made up of the first k rows and columns of $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_{k,1:l}$ contains the first l elements from row k of $\boldsymbol{\Sigma}$, and $\Sigma_{k,l}$ is the element from the kth row and lth column of $\boldsymbol{\Sigma}$, for $k, l = 1, \ldots, d$. Furthermore, let $\tilde{\boldsymbol{X}}_k = (1, \boldsymbol{X}_{1:k-1})^T$, where $\boldsymbol{X}_{1:k}$ contains the first k elements of $\boldsymbol{X}$, and let $\boldsymbol{\beta}_k = (\beta_{k,0}, \ldots, \beta_{k,k-1})^T \in \mathbb{R}^k$ and $\sigma_k^2 > 0$ be parameters. Using this notation, Ingrassia et al. (2012) showed that for every $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, there exists a parametrization $\boldsymbol{\beta}_d, \sigma_d^2, \boldsymbol{\mu}_{1:d-1}$, and $\boldsymbol{\Sigma}_{1:d-1,1:d-1}$ such that the density function

$$\begin{aligned} f_{CW}(\boldsymbol{x}; \boldsymbol{\beta}_d, \sigma_d^2, \boldsymbol{\mu}_{1:d-1}, \boldsymbol{\Sigma}_{1:d-1,1:d-1}) &= \phi_1(x_d; \boldsymbol{\beta}_d^T \tilde{\boldsymbol{x}}_d, \sigma_d^2) \\ &\times \phi_{d-1}(\boldsymbol{x}_{1:d-1}; \boldsymbol{\mu}_{1:d-1}, \boldsymbol{\Sigma}_{1:d-1,1:d-1}) \end{aligned} \tag{4}$$

is equal to $\phi_d(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, for all values of $\boldsymbol{x}$. This alternative parametrization allows for the d-variate Gaussian distribution to be considered in two parts: a linear regression (LR) and density estimation component. In Ingrassia et al. (2012, 2014), this parametrization was used within the cluster-weighted modeling framework for clustering data arising from LR processes.

We extend upon the regression decomposition of Ingrassia et al. (2012) to characterize the multivariate Gaussian distribution entirely in terms of LR components. In doing so, we are able to apply a minorization–maximization (MM) algorithm presented in Becker et al. (1997), for the estimation of LR models without matrix operations via

an iterative scheme; see Hunter and Lange (2004) for further details regarding MM algorithms. Furthermore, by leveraging its MM construction, the algorithm that we present is proven to be monotonic in its iterations as well as convergent to a stationary point of the log-likelihood function. We are able to show via simulations that our algorithm compares favorably with the EM algorithm in various practical scenarios, when implemented in the *R* programming environment R Core Team (2013).

Aside from the numerical properties that we derive, we also address the statistical properties of our LR characterization (LRC). We are able to establish both the consistency and asymptotic normality of the LRC ML estimators, as well as devise a simple procedure for the satisfactory handling of singularities, which can arise in the ML estimation of GMMs.

The remainder of the article is organized as follows. Firstly, we describe the LRC of the multivariate normal distribution in Sect. 2. In Sect. 3, we devise the MM algorithm for ML estimation as well as establish its numerical properties. In Sect. 4, the statistical properties of the ML estimators and the model are derived, and in Sect. 5, the performance of the MM algorithm is demonstrated via numerical simulations. Lastly, conclusions are drawn in Sect. 6.

2 Linear regression characterization

Using the same notation as in (3) and (4), the LRC of the multivariate Gaussian density is

$$\lambda(\boldsymbol{x}; \boldsymbol{\gamma}, \boldsymbol{\sigma}^2) = \prod_{k=1}^{d} \phi_1(x_k; \boldsymbol{\beta}_k^T \tilde{\boldsymbol{x}}_k, \sigma_k^2), \tag{5}$$

where $\boldsymbol{\gamma} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_d^T)^T$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_d^2)$. Here, we define $\tilde{\boldsymbol{x}}_1 = 1$ and $\boldsymbol{\beta}_1 = \beta_{1,0}$. We now show that (5) is a d-dimensional multivariate Gaussian density and that there is a one-to-one correspondence between the LRC parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}^2$ and the natural parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of (2). To attain such a result, we require the following lemma.

Lemma 1 *Using the same notation as in* (3) *and* (4), *if* $\boldsymbol{X}$ *has density function* (2), *then for each* k, $\boldsymbol{X}_{1:k}$ *and* $X_k|\boldsymbol{X}_{1:k-1} = \boldsymbol{x}_{1:k-1}$ *have density functions*

$$\phi_k(\boldsymbol{x}_{1:k}; \boldsymbol{\mu}_{1:k}, \boldsymbol{\Sigma}_{1:k,1:k}) \quad \textit{and} \quad \phi_1(x_k; \mu_{k|1:k-1}(\boldsymbol{x}_{1:k-1}), \Sigma_{k|1:k-1}),$$

respectively, where

$$\mu_{k|1:k-1}(\boldsymbol{x}_{1:k-1}) = \mu_k + \boldsymbol{\Sigma}_{k,1:k-1} \boldsymbol{\Sigma}_{1:k-1,1:k-1}^{-1} (\boldsymbol{x}_{1:k-1} - \boldsymbol{\mu}_{1:k-1}),$$

and

$$\Sigma_{k|1:k-1} = \Sigma_{k,k} - \boldsymbol{\Sigma}_{k,1:k-1} \boldsymbol{\Sigma}_{1:k-1,1:k-1}^{-1} \boldsymbol{\Sigma}_{k,1:k-1}^T.$$

Lemma 1 can be seen as a special case Theorems 2.4.1 and 2.5.1 from Anderson (2003). We can apply Lemma 1 to derive the following result.

Theorem 1 *Every density function of form* (2) *can be expressed in form* (5) *via a bijective mapping between the parameters* $\boldsymbol{\mu}$ *and* $\boldsymbol{\Sigma}$, *and* $\boldsymbol{\gamma}$ *and* $\boldsymbol{\sigma}^2$.

The proof of Theorem 1 and other major results can be found in the Appendix. As an example application of Theorem 1, consider that the 3-dimensional Gaussian density $\phi_3(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is equal to

$$\lambda(\boldsymbol{x}; \boldsymbol{\gamma}, \boldsymbol{\sigma}^2) = \phi_1(x_1; \beta_{1,0}, \sigma_1^2)\phi_1(x_2; \boldsymbol{\beta}_2^T \tilde{\boldsymbol{x}}_2, \sigma_2^2)\phi_1(x_3; \boldsymbol{\beta}_3^T \tilde{\boldsymbol{x}}_3, \sigma_3^2)$$

for all $\boldsymbol{x} \in \mathbb{R}^3$, where $\beta_{1,0} = \mu_1$, $\sigma_1^2 = \Sigma_{1,1}$, $\beta_{2,0} = \mu_2 - \Sigma_{2,1}\Sigma_{1,1}^{-1}\mu_1$, $\beta_{2,1} = \Sigma_{2,1}\Sigma_{1,1}^{-1}$, $\sigma_2^2 = \Sigma_{2,2} - \Sigma_{2,1}^2\Sigma_{1,1}^{-1}$,

$$\begin{aligned} \beta_{3,0} &= \mu_3 - \boldsymbol{\Sigma}_{3,1:2}\boldsymbol{\Sigma}_{1:2,1:2}^{-1}\boldsymbol{\mu}_{1:2}, \\ (\beta_{3,1}, \beta_{3,2})^T &= \boldsymbol{\Sigma}_{3,1:2}\boldsymbol{\Sigma}_{1:2,1:2}^{-1}, \end{aligned}$$

and

$$\sigma_3^2 = \Sigma_{3,3} - \boldsymbol{\Sigma}_{3,1:2}\boldsymbol{\Sigma}_{1:2,1:2}^{-1}\boldsymbol{\Sigma}_{3,1:2}^T.$$

2.1 Ordering of variables

As noted by a reviewer, the bijective mapping between $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}^2$ only holds for the unpermuted ordering of the elements of $\boldsymbol{X} = (X_1, \ldots, X_d)^T$. That is, if $\boldsymbol{\Pi} \neq \boldsymbol{I}$ is a permutation matrix, as defined in Section 8.2 of Seber (2008) [i.e. $\boldsymbol{\Pi}\boldsymbol{X}$ permutes the ordering of the elements of $\boldsymbol{X}$; e.g. there exists a $\boldsymbol{\Pi}$ such that $\boldsymbol{\Pi}(X_1, X_2, X_3)^T = (X_3, X_1, X_2)^T$], then there exist functions $\boldsymbol{\gamma}_{\boldsymbol{\Pi}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{\sigma}_{\boldsymbol{\Pi}}^2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that

$$\begin{aligned} \lambda(\boldsymbol{x}; \boldsymbol{\gamma}, \boldsymbol{\sigma}^2) &= \lambda(\boldsymbol{\Pi}\boldsymbol{x}; \boldsymbol{\gamma}_{\boldsymbol{\Pi}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \boldsymbol{\sigma}_{\boldsymbol{\Pi}}^2(\boldsymbol{\mu}, \boldsymbol{\Sigma})) \\ &= \phi_d(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{aligned}$$

for all $\boldsymbol{x} \in \mathbb{R}^d$, where it is possible that $\boldsymbol{\gamma} \neq \boldsymbol{\gamma}_{\boldsymbol{\Pi}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $\boldsymbol{\sigma}^2 \neq \boldsymbol{\sigma}_{\boldsymbol{\Pi}}^2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Here, $\boldsymbol{I}$ is an identity matrix of appropriate dimension. This implies that if we permute the order of the elements in the data vector, then there exists an alternative LRC of any Gaussian density that we wish to represent, which may not be the same as the original LRC.

We note that although the explicit forms of $\boldsymbol{\gamma}_{\boldsymbol{\Pi}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{\sigma}_{\boldsymbol{\Pi}}^2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are not provided, they can be constructed in the following way. Firstly, consider that for any permutation matrix $\boldsymbol{\Pi}$,

$$\phi_d(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \phi_d(\boldsymbol{\Pi}\boldsymbol{x}; \boldsymbol{\Pi}\boldsymbol{\mu}, \boldsymbol{\Pi}\boldsymbol{\Sigma}\boldsymbol{\Pi}^T)$$

for all $\boldsymbol{x} \in \mathbb{R}^d$. Secondly, Eqs. (22)–(25) can be used to map the elements of $\boldsymbol{\Pi}\boldsymbol{\mu}$ and $\boldsymbol{\Pi}\boldsymbol{\Sigma}\boldsymbol{\Pi}^T$ to the elements of $\boldsymbol{\gamma}_{\boldsymbol{\Pi}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{\sigma}_{\boldsymbol{\Pi}}^2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Thus, the LRC parameters of any permutation of the elements of $\boldsymbol{X}$ can be obtain via knowledge of the form of the permutation, and the parameters of the underlying Gaussian density function of $\boldsymbol{X}$.

2.2 Gaussian mixture model

We now consider the LRC of the GMM density function

$$f_R(\boldsymbol{x};\boldsymbol{\psi}) = \sum_{i=1}^{g} \pi_i \lambda(\boldsymbol{x};\boldsymbol{\gamma}_i,\boldsymbol{\sigma}_i^2), \tag{6}$$

where $\boldsymbol{\psi} = (\boldsymbol{\pi}^T, \boldsymbol{\gamma}_1^T, \ldots, \boldsymbol{\gamma}_g^T, \boldsymbol{\sigma}_1^2, \ldots, \boldsymbol{\sigma}_g^2)^T$ is the vector of model parameters. Here, $\boldsymbol{\gamma}_i = (\boldsymbol{\beta}_{i,1}^T, \ldots, \boldsymbol{\beta}_{i,d}^T)^T$, $\boldsymbol{\sigma}_i^2 = (\sigma_{i,1}^2, \ldots, \sigma_{i,d}^2)$, and $\boldsymbol{\beta}_{i,k} = (\beta_{i,k,0}, \ldots, \beta_{i,k,k-1})^T$, for each i and k. Furthermore, $\boldsymbol{\pi}$ is restricted in the same way as in (2). Using Theorem 1, the following result can be shown.

Corollary 1 *For each natural parameter vector $\boldsymbol{\theta}$, there exists a mapping to an LRC parameter vector $\boldsymbol{\psi}$, and vice versa, such that $f(\boldsymbol{x};\boldsymbol{\theta}) = f_R(\boldsymbol{x};\boldsymbol{\psi})$ at every $\boldsymbol{x} \in \mathbb{R}^d$.*

Corollary 1 can be seen as an extension of Proposition 1 from Ingrassia et al. (2012). Unfortunately, unlike Theorem 1, the mapping between $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ is not bijective, due to the non-identifiability of GMMs; this issue is well documented in Section 3.1 of Titterington et al. (1985). Nevertheless, Corollary 1 allows us to consider the density estimation of data generated from a GMM using an LRC of the GMM instead. We shall show that this representation permits the construction of a matrix-free algorithm for ML estimation.

3 Maximum likelihood estimation

Upon observing data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, the likelihood and the log-likelihood that the data arise from a LRC of the GMM are $\mathcal{L}_{R,n}(\boldsymbol{\psi}) = \prod_{j=1}^{n} f_R(\boldsymbol{x}_j;\boldsymbol{\psi})$ and

$$\begin{aligned}\log \mathcal{L}_{R,n}(\boldsymbol{\psi}) &= \sum_{j=1}^{n} \log f_R(\boldsymbol{x};\boldsymbol{\psi}) \\ &= \sum_{j=1}^{n} \log \sum_{i=1}^{g} \pi_i \lambda(\boldsymbol{x};\boldsymbol{\gamma}_i,\boldsymbol{\sigma}_i^2),\end{aligned} \tag{7}$$

respectively.

As with the natural parametrization of the GMM, we generally assume that the data were generated from a process with density function $f_R(\boldsymbol{x};\boldsymbol{\psi}^0)$, where $\boldsymbol{\psi}^0$ is unknown. In such cases, $\boldsymbol{\psi}^0$ can be estimated via the ML estimator $\hat{\boldsymbol{\psi}}_n$, where $\hat{\boldsymbol{\psi}}_n$ is an appropriate local maximizer of (7).

Like $\hat{\boldsymbol{\theta}}_n$, $\hat{\boldsymbol{\psi}}_n$ cannot be computed in closed form. Thus, we must devise an iterative computation scheme. We now present an MM algorithm for such a purpose.

3.1 Minorization–maximization algorithms

Suppose that we wish to maximize some objective function $\eta(\boldsymbol{t})$, where $\boldsymbol{t} \in S$ for some set $S \subset \mathbb{R}^r$. If we cannot obtain the maximizer of $\eta(\boldsymbol{t})$ directly, then we can seek

a minorizer of $\eta(\boldsymbol{t})$ over S, instead. A minorizer of $\eta(\boldsymbol{t})$ is a function $h(\boldsymbol{s};\boldsymbol{t})$ such that $\eta(\boldsymbol{t}) = h(\boldsymbol{t};\boldsymbol{t})$ and $\eta(\boldsymbol{s}) \geq h(\boldsymbol{s};\boldsymbol{t})$, whenever $\boldsymbol{s} \neq \boldsymbol{t}$, and $\boldsymbol{s} \in S$. Here, $h(\boldsymbol{s};\boldsymbol{t})$ is said to minorize $\eta(\boldsymbol{t})$.

Upon finding an appropriate minorizer and denoting $\boldsymbol{t}^{(m)}$ as the mth iteration of the algorithm, an MM algorithm can be defined using the update scheme

$$\boldsymbol{t}^{(m+1)} = \arg\max_{\boldsymbol{s}\in S} h(\boldsymbol{s};\boldsymbol{t}^{(m)}). \tag{8}$$

By the properties of the minorizer and by definition of (8), iterative applications of the MM update scheme yields the inequalities,

$$\eta(\boldsymbol{t}^{(m+1)}) \geq h(\boldsymbol{t}^{(m+1)};\boldsymbol{t}^{(m)}) \geq h(\boldsymbol{t}^{(m)};\boldsymbol{t}^{(m)}) = \eta(\boldsymbol{t}^{(m)}).$$

This shows that the sequence of objective function evaluations $\eta(\boldsymbol{t}^{(m)})$ is monotonically increasing in each step. Furthermore, under some regularity conditions, it can also be shown that the sequence of iterates $\boldsymbol{t}^{(m)}$ converges to some stationary point $\boldsymbol{t}^*$ of $\eta(\boldsymbol{t})$.

In this article, we consider minorizers for the functions

$$\eta_1(\boldsymbol{t}) = \log\left(\sum_{i=1}^{r} t_i\right),$$

where $S = \{\boldsymbol{t} : t_i \geq 0, i = 1,\ldots,r\}$, and

$$\eta_2(\boldsymbol{t}) = -(a - \boldsymbol{t}^T\boldsymbol{b})^2,$$

where $a \in \mathbb{R}$, $\boldsymbol{b} \in \mathbb{R}^r$, and $S = \mathbb{R}^r$. These objective functions [i.e. $\eta_1(\boldsymbol{t})$ and $\eta_2(\boldsymbol{t})$] can be minorized via the functions

$$h_1(\boldsymbol{s};\boldsymbol{t}) = \sum_{i=1}^{r} \frac{t_i}{\sum_{i'=1}^{r} t_{i'}} \left[\log\left(\frac{\sum_{i'=1}^{r} t_{i'}}{t_i}\right) + \log(s_i)\right], \tag{9}$$

and

$$h_2(\boldsymbol{s};\boldsymbol{t}) = -\sum_{i=1}^{r} \alpha_i \left[a - \frac{b_i}{\alpha_i}(s_i - t_i) - \boldsymbol{t}^T\boldsymbol{b}\right]^2, \tag{10}$$

respectively, where $\alpha_i = (|b_i|^p + \delta)/\sum_{i'=1}^{r}(|b_{i'}|^p + \delta)$, $p > 0$, and $\delta > 0$ is a small coefficient. The two minorizers were devised in Zhou and Lange (2010) and Becker et al. (1997), respectively; the latter was applied to perform LR without matrix operations. Here, we choose $p = 2$ in α_i for use throughout the article, as per a suggestion from Becker et al. (1997).

Let $\boldsymbol{\psi}^{(m)}$ be the mth MM iterate. By setting $s_i = \pi_i\lambda(\boldsymbol{x}_j;\boldsymbol{\gamma}_i,\boldsymbol{\sigma}_i^2)$ and $t_i = \pi_i^{(m)}\lambda(\boldsymbol{x}_j;\boldsymbol{\gamma}_i^{(m)},\boldsymbol{\sigma}_i^{(m)2})$ in (9), for each i and j, we get the minorizer for (7),

$$
\begin{aligned}
Q(\boldsymbol{\psi}; \boldsymbol{\psi}^{(m)}) &= C(\boldsymbol{\psi}^{(m)}) + \sum_{i=1}^{g}\sum_{j=1}^{n} \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)}) \log[\pi_i \lambda(\boldsymbol{x}; \boldsymbol{\gamma}_i, \sigma_i^2)] \\
&= C(\boldsymbol{\psi}^{(m)}) + Q_1(\boldsymbol{\psi}; \boldsymbol{\psi}^{(m)}) \\
&\quad + \sum_{i=1}^{g}\sum_{k=1}^{d} \frac{1}{2\sigma_{i,k}^2} \sum_{j=1}^{n} \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)}) Q_{2,i,j,k}(\boldsymbol{\beta}_{i,k}; \boldsymbol{\beta}_{i,k}^{(m)}), \qquad (11)
\end{aligned}
$$

where

$$
\begin{aligned}
Q_1(\boldsymbol{\psi}; \boldsymbol{\psi}^{(m)}) &= \sum_{i=1}^{g}\sum_{j=1}^{n} \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)}) \log \pi_i - \frac{1}{2} \sum_{i=1}^{g}\sum_{k=1}^{d} \log \sigma_{i,k}^2 \sum_{j=1}^{n} \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)}), \\
Q_{2,i,j,k}(\boldsymbol{\beta}_{i,k}; \boldsymbol{\beta}_{i,k}^{(m)}) &= -(x_{j,k} - \boldsymbol{\beta}_{i,k}^T \tilde{\boldsymbol{x}}_{j,k})^2, \qquad (12)
\end{aligned}
$$

and

$$
\tau_i(\boldsymbol{x}; \boldsymbol{\psi}) = \frac{\pi_i \lambda(\boldsymbol{x}; \boldsymbol{\gamma}_i, \sigma_i^2)}{\sum_{i'=1}^{g} \pi_{i'} \lambda(\boldsymbol{x}; \boldsymbol{\gamma}_{i'}, \sigma_{i'}^2)}. \qquad (13)
$$

Here, $\boldsymbol{x}_j = (x_{j,1}, \ldots, x_{j,d})^T$ and $\tilde{\boldsymbol{x}}_{j,k} = (1, x_{j,1}, \ldots, x_{j,k-1})^T$ for each j and k, and $C(\boldsymbol{\psi}^{(m)})$ is a constant that does not depend on $\boldsymbol{\psi}$.

Now, by setting $a = x_{j,k}$, $\boldsymbol{b} = \tilde{\boldsymbol{x}}_{j,k}$, $\boldsymbol{s} = \boldsymbol{\beta}_{i,k}$, and $\boldsymbol{t} = \boldsymbol{\beta}_{i,k}^{(m)}$ in (10), we obtain the minorizer for (12),

$$
Q'_{2,i,j,k}(\boldsymbol{\beta}_{i,k}; \boldsymbol{\beta}_{i,k}^{(m)}) = -\sum_{l=0}^{k-1} \alpha_{j,l} \left[x_{j,k} - \frac{x_{j,l}}{\alpha_{j,l}} (\beta_{i,k,l} - \beta_{i,k,l}^{(m)}) - \boldsymbol{\beta}_{i,k}^{(m)T} \tilde{\boldsymbol{x}}_{j,k} \right]^2, \qquad (14)
$$

where $\alpha_{j,l} = (|x_{j,l}|^p + \delta) / \sum_{l'=0}^{k-1} (|x_{j,l}|^p + \delta)$ and $x_{j,0} = 1$ by definition, for j and $l = 0, \ldots, k-1$.

Using (14), we can further minorize (11), and thus (7), by

$$
\begin{aligned}
Q'(\boldsymbol{\psi}; \boldsymbol{\psi}^{(m)}) &= C(\boldsymbol{\psi}^{(m)}) + Q_1(\boldsymbol{\psi}; \boldsymbol{\psi}^{(m)}) \\
&\quad + \sum_{i=1}^{g}\sum_{k=1}^{d} \frac{1}{2\sigma_{i,k}^2} \sum_{j=1}^{n} \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)}) Q'_{2,i,j,k}(\boldsymbol{\beta}_{i,k}; \boldsymbol{\beta}_{i,k}^{(m)}). \qquad (15)
\end{aligned}
$$

We now consider the partition of $\boldsymbol{\psi}$ into $\boldsymbol{\psi}_1 = (\boldsymbol{\pi}^T, \boldsymbol{\gamma}_1^T, \ldots, \boldsymbol{\gamma}_g^T)^T$ and $\boldsymbol{\psi}_2 = (\sigma_1^2, \ldots, \sigma_g^2)^T$, where $\boldsymbol{\psi} = (\boldsymbol{\psi}_1^T, \boldsymbol{\psi}_2^T)^T$. By fixing $\boldsymbol{\psi}_2$ at $\boldsymbol{\psi}_2^{(m)}$, $Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})$ is additively separable in the subsets of $\boldsymbol{\psi}_1$; furthermore, for each i, the elements of $\boldsymbol{\gamma}_i$ are additively separable as well. Similarly, by fixing $\boldsymbol{\psi}_1$ at $\boldsymbol{\psi}_1^{(m)}$, $Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2; \boldsymbol{\psi}^{(m)})$ is additively separable in the elements of the subsets of $\boldsymbol{\psi}_2$. This result suggests the following block successive MM update scheme.

Let $\boldsymbol{\psi}^{(0)}$ be an initial parameter vector. At the $(m+1)$th iteration, the algorithm proceeds in two steps: the min (minorization)-step and the max (maximization)-step. In the min-step, we either construct (15) if m is odd, or (11) if m is even; in either cases, the min-step requires the computation of $\tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)})$, for each j.

In the max-step, if m is odd, then we set $\boldsymbol{\psi}_2^{(m+1)} = \boldsymbol{\psi}_2^{(m)}$ and solve for the root of

$$\frac{\partial Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})}{\partial \boldsymbol{\psi}_1} = \mathbf{0},$$

to get the updates

$$\pi_i^{(m+1)} = \frac{\sum_{j=1}^n \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)})}{n} \tag{16}$$

and

$$\beta_{i,k,l}^{(m+1)} = \beta_{i,k,l}^{(m)} + \frac{\sum_{j=1}^n x_{j,l}\tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)})[x_{j,k} - \boldsymbol{\beta}_{i,k}^{(m)T}\tilde{\boldsymbol{x}}_{j,k}]}{\sum_{j=1}^n [x_{j,l}^2 \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)})/\alpha_{j,l}]} \tag{17}$$

for each i, k, and $l = 0, \ldots, k-1$. Here, $\mathbf{0}$ is a zero matrix/vector of appropriate dimensionality.

Similarly, the max-step for even m proceeds by setting $\boldsymbol{\psi}_1^{(m+1)} = \boldsymbol{\psi}_1^{(m)}$ and solving for the root of

$$\frac{\partial Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2; \boldsymbol{\psi}^{(m)})}{\partial \boldsymbol{\psi}_2} = \mathbf{0}$$

to obtain the updates

$$\sigma_{i,k}^{(m+1)2} = \frac{\sum_{j=1}^n \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)})[x_{j,k} - \boldsymbol{\beta}_{i,k}^{(m)T}\tilde{\boldsymbol{x}}_{j,k}]^2}{\sum_{j=1}^n \tau_i(\boldsymbol{x}_j; \boldsymbol{\psi}^{(m)})} \tag{18}$$

for each i and k. We now show that together, updates (16)–(18) generate a sequence of monotonically increasing log-likelihood values.

Theorem 2 *If m is odd and $\boldsymbol{\psi}_2^{(m+1)} = \boldsymbol{\psi}_2^{(m)}$, then updates (16) and (17) result in the inequalities*,

$$\begin{aligned} \log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m+1)}) &\geq Q'(\boldsymbol{\psi}_1^{(m+1)}, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)}) \\ &\geq Q'(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)}) \geq \log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m)}). \end{aligned} \tag{19}$$

If m is even and $\boldsymbol{\psi}_1^{(m+1)} = \boldsymbol{\psi}_1^{(m)}$, then update (18) results in the inequalities,

$$\begin{aligned} \log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m+1)}) &\geq Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2^{(m+1)}; \boldsymbol{\psi}^{(m)}) \\ &\geq Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)}) \geq \log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m)}). \end{aligned} \tag{20}$$

In general, the MM algorithm is iterated until some convergence criterion is met. Usually, this is either the absolute convergence criterion

$$\log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m+1)}) - \log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m)}) \leq \epsilon,$$

or the relative convergence criterion

$$\frac{\log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m+1)}) - \log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m)})}{|\log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m)})|} \leq \epsilon, \tag{21}$$

for some small $\epsilon > 0$. In either case, upon convergence, the final iterate of the algorithm is declared the ML estimator $\hat{\boldsymbol{\psi}}_n$. Let $\boldsymbol{\psi}^*$ be a limit point of the MM algorithm, such that $\hat{\boldsymbol{\psi}}_n \to \boldsymbol{\psi}^*$ as $\epsilon \to 0$ for some starting parameter $\boldsymbol{\psi}^{(0)}$.

The algorithm that we have devised is an instance of the block successive lower-bound maximization algorithms of Razaviyayn et al. (2013). As such, we can apply Theorem 2 of Razaviyayn et al. (2013) to obtain the following result regarding its limit points.

Theorem 3 *If $\boldsymbol{\psi}^*$ is a limit point of the algorithm conducted via the steps $\boldsymbol{\psi}_2^{(m+1)} = \boldsymbol{\psi}_2^{(m)}$, (16), and (17), when m is odd; and $\boldsymbol{\psi}_1^{(m+1)} = \boldsymbol{\psi}_1^{(m)}$ and (18), when m is even, for some initial vector $\boldsymbol{\psi}^{(0)}$; then $\boldsymbol{\psi}^*$ is a stationary point of (7).*

Theorem 3 shows that given suitable initial parameter vector $\boldsymbol{\psi}^{(0)}$, the MM algorithm generates a sequence $\boldsymbol{\psi}^{(m)}$ that converges to a stationary point of (7); this is a good result considering its multimodality.

3.2 Covariance constraints

We note that Theorem 3 requires that the limit points be finite values. This cannot always be guaranteed since $\log \mathcal{L}_{R,n}(\boldsymbol{\psi}^{(m)}) \to \infty$ if any of the sequences $\sigma_{i,k}^{(m)2} \to 0$. This is equivalent to the problem of component covariance matrices $\boldsymbol{\Sigma}_i$ becoming singular in the natural parametrization [i.e. in $\mathcal{L}_n(\boldsymbol{\theta})$]. In the natural parametrization, the usual approach is to restrict the component covariance matrices to be positive definite via conditioning on the eigenvalues of the matrices. Such approaches were pioneered in Hathaway (1985); examples of recent developments include Greselin and Ingrassia (2008), Ingrassia (2004), and Ingrassia and Rocci (2007, 2011). We proceed to provide a simple alternative to the aforementioned approaches, based upon the LRC parametrization.

In the LRC, ensuring finite limit points amounts to guaranteeing that for each i and k, $\sigma_{i,k}^{(m)2} \to \xi_{i,k}$ for some $\xi_{i,k} > 0$. This can be implemented by adding a small $\xi > 0$ to the right-hand side of update (18) at each iteration. Through doing this, we ensure that each $\sigma_{i,k}^{*2}$ is positive, as well as retaining the monotonicity of the likelihood, for each update. The following result is then applicable.

Theorem 4 *In (5), if $\sigma_k^2 > 0$ for each k, then the corresponding covariance matrix $\boldsymbol{\Sigma}$, of the natural parametrization, is positive-definite.*

Theorem 4 implies that the covariance matrices in the natural parametrization will always be positive-definite, if we apply the described process. As suggested by a reviewer, it is practically important to choose a ξ such that it does not impede upon the estimation of mixture components with small variances. If we assume that the marginal variances of each marginal component do not exceed a proportion Ξ^{-1} ($\Xi > 1$) of the corresponding marginal variances of the overall distribution [i.e. $\text{var}(X_1), \ldots, \text{var}(X_d)$], then we can choose

$$\xi = \min\left\{\frac{\hat{\text{var}}\,(X_1)}{\Xi}, \ldots, \frac{\hat{\text{var}}\,(X_d)}{\Xi}\right\},$$

where $\hat{\text{var}}(X_k)$ is an estimate for the marginal variance of the kth dimension. We use $\Xi = 10^{10}$ for all numerical applications presented in this article.

4 Statistical properties

The consistency and asymptotic normality of ML estimators for GMM under the natural parametrization have been proven in many instances; see for example, Redner and Walker (1984), Hathaway (1985), and Atienza et al. (2007). We now seek the consistency of the ML estimators of the LRC of the GMM. Such a result can be obtained via Theorem 4.1.2 of Amemiya (1985).

Theorem 5 *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be independent and identically distributed random samples from a distribution with density $f_R(\boldsymbol{x}; \boldsymbol{\psi}^0)$, and let Ψ_n be the set of roots of the equation $\partial(\log \mathcal{L}_{R,n}(\boldsymbol{\psi}))/\partial\boldsymbol{\psi} = \boldsymbol{0}$, where $\Psi_n = \{\boldsymbol{0}\}$ if there are no roots. If $\boldsymbol{\psi}^0$ is a strict local maximizer of $\mathbb{E}[\log f_R(\boldsymbol{X}; \boldsymbol{\psi})]$, then for any $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}\left[\inf_{\boldsymbol{\psi}\in\Psi_n} (\boldsymbol{\psi} - \boldsymbol{\psi}^0)^T(\boldsymbol{\psi} - \boldsymbol{\psi}^0) > \epsilon\right] = 0.$$

Theorem 5 is an adequate result, considering that the log-likelihoods of GMMs are often multimodal and unbounded, and that the MM algorithm is able to locate local maximizers when started from suitable values. We note that the result similarly holds when a lower bound is enforced for each of the variance limit points σ_{ik}^{*2}, as in Sect. 3.2.

We now seek to establish the asymptotic normality of the ML estimators. Upon making some assumptions (see the proof in the Appendix), we are able to utilize Theorem 4.2.4 of Amemiya (1985) to get the following result.

Theorem 6 *Under Assumption* B4, *the ML estimator $\hat{\boldsymbol{\psi}}_n$ (as in Theorem 5) satisfies*

$$\sqrt{n}(\hat{\boldsymbol{\psi}}_n - \boldsymbol{\psi}^0) \xrightarrow{D} N\left(\boldsymbol{0}, -\mathbb{E}\left[\frac{\partial^2 \log f_R(\boldsymbol{x}; \boldsymbol{\psi})}{\partial\boldsymbol{\psi}\,\partial\boldsymbol{\psi}^T}\bigg|_{\boldsymbol{\psi}=\boldsymbol{\psi}^0}\right]^{-1}\right).$$

Theorems 5 and 6 allow for inferences to be drawn from the ML estimators and their functions. For example, if $\boldsymbol{x}$ is an observation that arises from a distribution

with density $f_R(\boldsymbol{x}; \boldsymbol{\psi}^0)$, then we can compute the conditional probability of its latent component variable $Z = i$ given $\boldsymbol{X} = \boldsymbol{x}$, by $\tau_i(\boldsymbol{x}; \boldsymbol{\psi}^0)$ via an application of Bayes rule. Furthermore, if $\hat{\boldsymbol{\psi}}_n \xrightarrow{P} \boldsymbol{\psi}^0$, then by continuous mapping, we have $\tau_i(\boldsymbol{x}; \hat{\boldsymbol{\psi}}_n) \xrightarrow{P} \tau_i(\boldsymbol{x}; \boldsymbol{\psi}^0)$, for each i. Thus, the estimated allocation rule that assigns $\boldsymbol{x}$ into component $\hat{z} \in \{1, \ldots, g\}$,

$$\hat{z} = \arg \max_{i \in \{1,\ldots,g\}} \tau_i(\boldsymbol{x}; \hat{\boldsymbol{\psi}}_n),$$

is asymptotically correct (i.e. it asymptotically assigns $\boldsymbol{x}$ to the component which maximizes the a posteriori probability).

5 Numerical simulations

To assess the performance of the MM algorithm proposed in Sect. 3.1, we conduct a set of three numerical simulation studies: S1, S2, and S3. In S1, we investigate the performance of the MM algorithm against the standard EM algorithm for estimating GMMs in a setting where the simulated data arise from clusters of equal sample sizes. In S2, the setup from S1 is repeated albeit with simulated data that arises from clusters with differing sample sizes. Finally, in S3, we assess whether or not the ordering of the variables (as discussed in Sect. 2.1) are influential in the performance of the MM algorithm. We present the setups and results of the numerical simulations below.

5.1 Numerical simulation S1

In S1, we simulated 2^{N-G} observations from each of 2^G Gaussian distributions of dimensionality 2^D, where $D = 1, \ldots, 4$, $G = 1, \ldots, 4$, and $N = 15, 16, 17$. These sample sizes were chosen since performance improvements are most relevant in large data sets. A sample simulated via this design approximately corresponds to a sample of 2^N observations from a 2^G component GMM, where $\pi_1 = \cdots = \pi_{2^G} = 2^{-G}$.

In each scenario, each of the 2^G distribution means is randomly sampled from a Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $20 \times \boldsymbol{I}$. The covariance matrices of the Gaussian distributions are each sampled from a Wishart distribution with scale matrix $\boldsymbol{I}$ and $2^D + 2$ degrees of freedom. An example of the $D = 2$, $G = 3$, and $N = 15$ case is shown in Fig. 1.

For each D, G, and N, 100 trials are simulated. In each trial, the MM algorithm is used to compute the ML estimates $\hat{\boldsymbol{\psi}}_n$ for the LRC parameters. Here, the algorithm is terminated using criterion (21) with $\epsilon = 10^{-5}$, and the computational time is recorded. The traditional EM algorithm (see Section 3.2 of McLachlan and Peel 2000) is then used to compute the ML estimates $\hat{\boldsymbol{\theta}}_n$ for the natural parameters, using the same starting values as for the MM algorithm. The EM algorithm is terminated using the criterion

$$\log \mathcal{L}_{R,n}(\hat{\boldsymbol{\psi}}_n) - \log \mathcal{L}_n(\boldsymbol{\theta}_n^{(m)}) < \epsilon,$$

using $\epsilon = 10^{-5}$, and the computational time is recorded. The k-means algorithm was used to initialize parameters, as per Section 2.12 of McLachlan and Peel (2000); see MacQueen (1967) regarding the k-means algorithm.

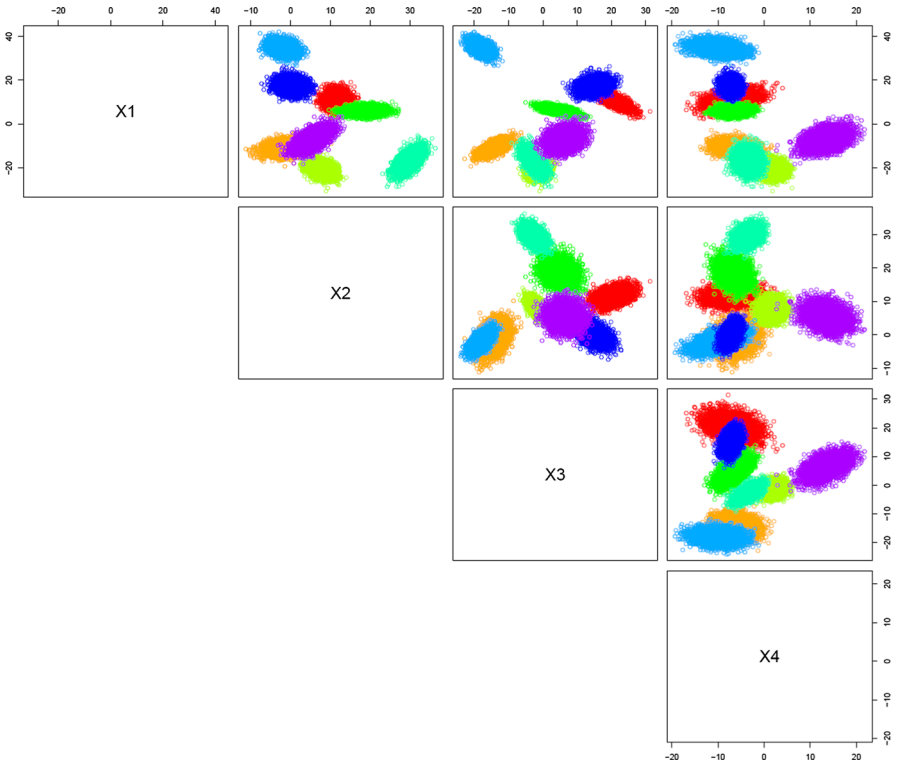

Fig. 1 Pairwise marginal plots of data generated from the $D = 2$, $G = 3$, and $N = 15$ case of the simulations. The *eight colors* indicate the different origins of each of the generated data points

The algorithms were applied via implementations in the *R* programming environment (version 3.0.2) on an Intel Core i7-2600 CPU running at 3.40 GHz with 16 GB internal RAM, and the timing was conducted using the *proc.time* function from said environment. The computational times, in seconds, for both algorithms were then averaged over the trials, for each scenario, and the results are reported in Table 1. In Fig. 2, we also plot the average ratio of EM to MM algorithm computational times, for each scenario.

Upon inspection, Table 1 suggests that both the MM and the EM algorithms behave as expected, with regards to the increases in computation times with respect to increases in D, G, and N. Furthermore, we notice that in each scenario, the MM algorithm is faster than the EM algorithm on average. In the best case, the MM algorithm is approximately 35 times faster than the EM (i.e. case $D = 1$, $G = 2$, and $N = 15$), and in the worst case, the MM and EM are approximately at parity (i.e. case $D = 4$, $G = 1$, and $N = 17$).

In Fig. 2, the performance of the MM algorithm over the EM decreases due to increases in D, G, and N, with D decreasing this gain more severely than the other two variables. This pattern may be explained by some of the additional computation overhead of the MM algorithm. For instance, notice that the MM algorithm requires the

Table 1 Results of numerical simulation S1 for assessing the performance of the MM and EM algorithms

N	D	G	MM	EM	Ratio	N	D	G	MM	EM	Ratio	N	D	G	MM	EM	Ratio
15	1	1	0.05	1.46	31.99	16	1	1	0.11	2.97	28.57	17	1	1	0.20	5.71	29.56
15	1	2	0.10	2.95	34.93	16	1	2	0.25	6.48	29.53	17	1	2	0.38	10.22	28.00
15	1	3	0.64	14.01	25.09	16	1	3	1.11	19.70	20.01	17	1	3	2.85	45.24	18.90
15	1	4	1.75	29.13	16.62	16	1	4	3.69	56.46	15.00	17	1	4	6.80	121.18	17.51
15	2	1	0.12	1.51	13.00	16	2	1	0.28	2.51	9.38	17	2	1	0.54	5.47	10.38
15	2	2	0.23	3.07	14.20	16	2	2	0.56	4.98	9.38	17	2	2	0.91	9.73	10.69
15	2	3	1.05	7.24	8.91	16	2	3	2.00	12.59	7.43	17	2	3	3.67	25.38	8.10
15	2	4	2.60	13.18	5.14	16	2	4	4.85	25.99	5.36	17	2	4	9.31	48.43	5.23
15	3	1	0.39	1.47	3.84	16	3	1	0.68	2.69	3.97	17	3	1	1.31	5.38	4.13
15	3	2	0.64	2.72	4.35	16	3	2	1.10	5.34	4.86	17	3	2	2.12	10.69	5.06
15	3	3	1.73	5.39	3.50	16	3	3	3.19	10.96	3.81	17	3	3	6.32	21.98	3.93
15	3	4	3.96	10.64	2.78	16	3	4	7.83	21.26	2.81	17	3	4	17.43	43.60	2.58
15	4	1	1.22	1.52	1.25	16	4	1	2.27	3.05	1.35	17	4	1	5.83	6.18	1.06
15	4	2	1.67	3.00	1.80	16	4	2	3.54	6.06	1.72	17	4	2	8.25	12.20	1.48
15	4	3	3.83	5.98	1.66	16	4	3	8.48	12.08	1.50	17	4	3	19.11	24.30	1.34
15	4	4	8.36	11.92	1.43	16	4	4	20.01	24.17	1.21	17	4	4	42.20	48.55	1.15

The columns N, D, and G indicate the simulation scenario, and MM and EM column displays the average computational time, in seconds, of the respective algorithms (over the 100 trials). The Ratio column displays the average ratio between the EM and the MM algorithm computation times

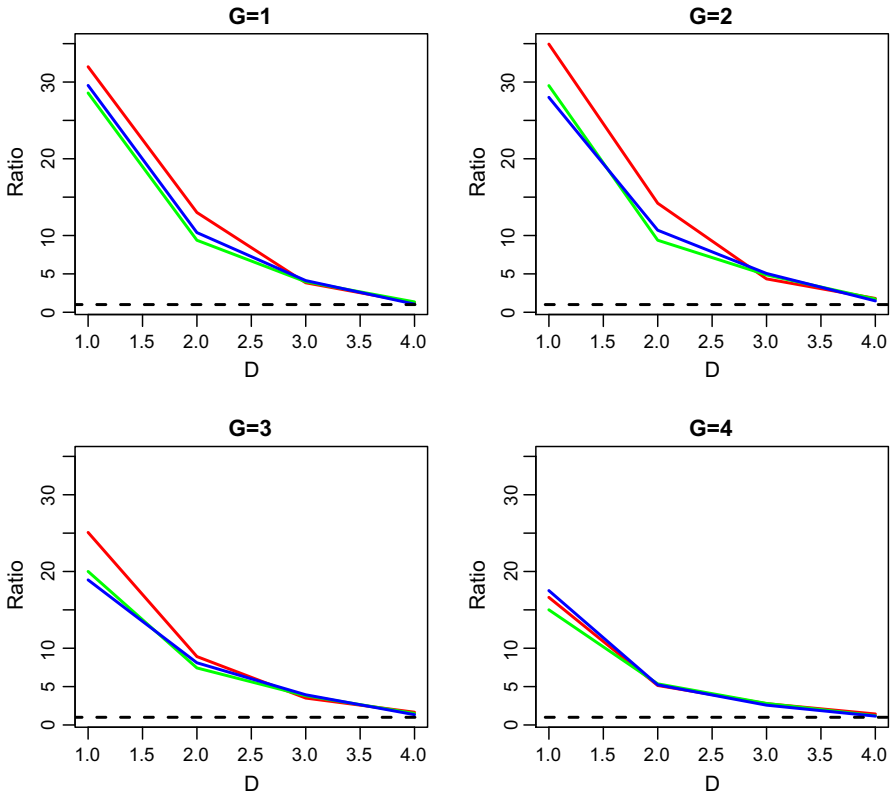

Fig. 2 Plots of the average ratio of EM to MM algorithm computational times in S1. The panels are separated into the G values of each scenario, and the N values are indicated by the *line colors*. Here, *red*, *green*, and *blue* indicate the values $N = 15, 16, 17$, respectively. The *dotted line* indicates a ratio of 1 (color figure online)

computation and storage of α_{jl} for each j, l, and k. The number of these computations increase quadratically in D, but only linearly in G and N. Due to this effect, we cannot recommend the MM algorithm in all situations. However, it is noticeable that in the low D cases, the MM algorithm appears to be distinctly faster than the EM.

5.2 Numerical simulation S2

In S2, we repeat the simulation design of S1 except instead of simulated 2^G equally sized samples, we simulated 2^{G-1} samples of size $(1/2) \times 2^{N-G}$ and 2^{G-1} samples of size $(3/2) \times 2^{N-G}$. A sample simulated via this design approximately corresponds to a sample of 2^N observations from a 2^G component GMM, where $\pi_1 = \cdots = \pi_{2^{G-1}} = 1/2^{G+1}$ and $\pi_{2^{G-1}+1} =, \ldots, \pi_{2^G} = 3/2^{G+1}$. Using the same comparison method and termination criterion as applied in S1, we obtain the results presented in Table 2. These results are further visualized in Fig. 3.

From Table 2, we notice that both the MM and EM algorithm average computational times are greater than the times in each of the respective scenarios, in S1. This indicates

Table 2 Results of numerical simulation S2 for assessing the performance of the MM and EM algorithms

N	D	G	MM	EM	Ratio	N	D	G	MM	EM	Ratio	N	D	G	MM	EM	Ratio
15	1	1	0.08	1.71	22.02	16	1	1	0.18	3.38	19.39	17	1	1	0.31	6.26	20.12
15	1	2	0.18	4.65	25.60	16	1	2	0.37	7.14	19.88	17	1	2	0.81	13.74	18.94
15	1	3	0.70	15.66	22.42	16	1	3	1.86	42.40	20.66	17	1	3	3.47	69.80	19.23
15	1	4	1.83	33.61	18.32	16	1	4	5.98	74.41	16.98	17	1	4	8.07	145.67	17.25
15	2	1	0.20	1.73	9.28	16	2	1	0.44	2.93	6.88	17	2	1	0.85	6.18	7.33
15	2	2	0.39	3.53	9.73	16	2	2	0.80	5.71	7.30	17	2	2	1.48	11.94	7.99
15	2	3	1.21	7.94	6.96	16	2	3	2.47	15.50	6.43	17	2	3	4.46	30.75	6.99
15	2	4	2.84	14.94	5.24	16	2	4	5.48	32.96	5.91	17	2	4	10.68	62.21	5.88
15	3	1	0.60	1.66	2.81	16	3	1	0.99	3.13	3.17	17	3	1	1.92	6.24	3.26
15	3	2	0.96	3.27	3.44	16	3	2	1.73	6.22	3.61	17	3	2	3.32	12.44	3.78
15	3	3	2.18	6.47	3.07	16	3	3	3.97	13.28	3.42	17	3	3	7.21	25.07	3.60
15	3	4	4.39	12.43	2.94	16	3	4	8.40	25.29	3.13	17	3	4	16.34	50.65	3.23
15	4	1	1.58	1.74	1.10	16	4	1	3.02	3.47	1.15	17	4	1	6.15	7.00	1.15
15	4	2	2.46	3.44	1.40	16	4	2	4.71	6.87	1.46	17	4	2	10.23	13.85	1.36
15	4	3	4.33	6.83	1.58	16	4	3	8.39	13.69	1.64	17	4	3	19.44	27.61	1.43
15	4	4	8.18	13.63	1.70	16	4	4	15.81	27.33	1.74	17	4	4	38.03	55.38	1.46

The columns N, D, and G indicate the simulation scenario, and MM and EM column displays the average computational time, in seconds, of the respective algorithms (over the 100 trials). The ratio column displays the average ratio between the EM and the MM algorithm computation times

that the problem of having estimating GMMs with differing cluster sizes using either algorithms may require more iterations to converge, on average. Like S1, it appears that the MM algorithm is again faster than the EM in all tested scenarios. However, we do note that the difference between the two algorithms is less in S2. For example, in the best case, the MM algorithm is only 25.6, times faster than the EM (i.e. case $D = 1$, $G = 2$, and $N = 15$).

Upon inspection of Fig. 3, we again see that the performance of the MM algorithm over the EM decreases due to increases in D, G, and N, again with D decreasing this gain more severely than the other two variables. Thus, we recognize that although there is a decrease in computational gains as the tested variables increase, there is a good case for the MM algorithm to be used instead of the EM when D is relatively small.

5.3 Numerical simulation S3

Following the design of S1, we simulated 100 samples from the $D = 2$, $G = 2$, and $N = 15, 16, 17$ cases. For each sample of each of the three cases, we perform ML estimation using the MM algorithm on all 24 possible permutations of the four variables. The computation time and the log-likelihood value of each permutation is then recorded, and a mean, range and relative range (as a ratio to the absolute value of the mean) is then computed over the 24 permutations, for both the computation time

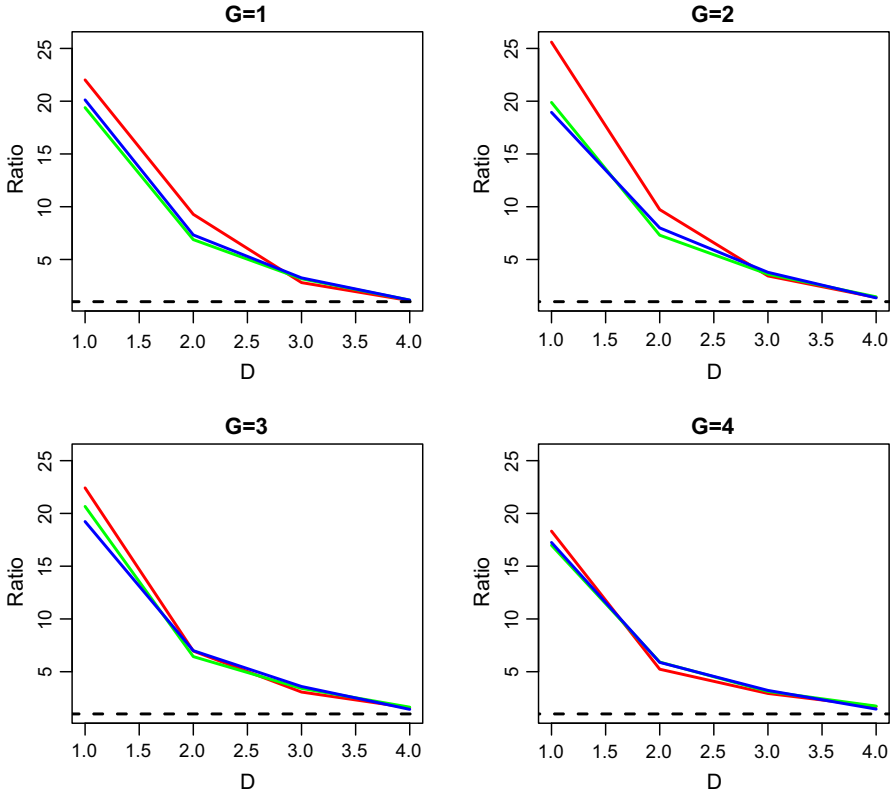

Fig. 3 Plots of the average ratio of EM to MM algorithm computational times in S2. The panels are separated into the G values of each scenario, and the N values are indicated by the *line colors*. Here, *red*, *green*, and *blue* indicate the values $N = 15, 16, 17$, respectively. The *dotted line* indicates a ratio of 1 (color figure online)

Table 3 Results of numerical simulation S3 for assessing the effects of variable ordering

N	Computation time			Log-likelihood		
	Mean	Range	RR	Mean	Range	RR
15	0.48	0.25	0.52	-2.71×10^5	5.80×10^3	2.11×10^{-2}
16	0.87	0.45	0.50	-5.42×10^5	1.56×10^4	2.88×10^{-2}
17	1.68	0.67	0.38	-1.09×10^6	2.84×10^4	2.63×10^{-2}

The column N indicate the scenarios, and the columns mean, range, and RR display the average means, ranges, and relative ranges of the computational time and the log-likelihood values

and the log-likelihood value. The average mean values, ranges and relative ranges over the 100 samples for all three scenarios are presented in Table 3.

Upon inspection of Table 3, we notice that the range of computation times across the three scenarios can be 40 to 50 % of the average computational times. Thus, selecting an advantageous permutation of the variables can lead to significant improvements in

algorithm performance. Unfortunately, the most advantageous permutation was not predictable in our simulations, and even in the $D = 2$ case with four variables, the number of permutations can be very large. Fortunately, the range of log-likelihood values is only two to three percent of that of the mean log-likelihood values in each scenario. As such, there appears to be little variation of the optimal outcome, once the algorithm has converged. This implies that regardless of performance, the algorithm is converging as expected according to Theorem 3.

We finally note that the results from all three numerical simulation studies are dependent on the specific performances of the subroutines, algorithms, and hardware. Thus, the results may vary when conducting performance tests under different settings. As such, we believe that this simulation study serves to demonstrate the potential computational performance in *R*, rather than the performance in all settings.

6 Conclusions

In this article, we introduced the LRC of the GMM, and show that there is a mapping between the LRC parameters and the natural parameters of a GMM. Using the LRC, we devised an MM algorithm for ML estimation, which does not depend on matrix operations. We then proved that the MM algorithm monotonically increases the log-likelihood in ML estimation, and that the sequence of estimators obtained from the algorithm is convergent to a stationary point of the log-likelihood function, under regularity conditions. Through simulations, we were able to demonstrate that the computational speed of the MM algorithm for the LRC parameter estimates was faster than the traditional EM algorithm for estimating GMMs in some large data situations, when both algorithms are implemented in the *R* programming environment. We also show that although the ordering of the variables may have a significant effect on the computational times of the MM algorithm, there appears to be little effect on the ability of the algorithm to converge to an appropriate limit point.

We also proved that the ML estimators of the LRC parameters, like those of the natural parameters, are also consistent and asymptotically normal. This allows for asymptotically valid statistical inference, such as using the LRC of the GMM for clustering data. Furthermore, we showed that the LRC allows for a simple method for handling singularities in the ML estimation of GMM parameters.

To the best of our knowledge, we are the first to apply the LRC for constructing a matrix operation-free algorithm for estimating GMMs. In the future, we hope to extend our matrix operation-free approach to the ML estimation of mixtures of t-distributions, as well as skew variants of the GMM.

Appendix

Proof of Theorem 1

We shall show the result by construction. Firstly, set

$$\beta_{0,1} = \mu_1 \quad \text{and} \quad \sigma_1^2 = \Sigma_{1,1}, \tag{22}$$

followed by

$$\beta_{k,0} = \mu_k - \boldsymbol{\Sigma}_{k,1:k-1}\boldsymbol{\Sigma}^{-1}_{1:k-1,1:k-1}\boldsymbol{\mu}_{1:k-1}, \tag{23}$$

$$(\beta_{k,1}, \ldots, \beta_{k,k-1}) = \boldsymbol{\Sigma}_{k,1:k-1}\boldsymbol{\Sigma}^{-1}_{1:k-1,1:k-1}, \tag{24}$$

and

$$\sigma_k^2 = \Sigma_{k,k} - \boldsymbol{\Sigma}_{k,1:k-1}\boldsymbol{\Sigma}^{-1}_{1:k-1,1:k-1}\boldsymbol{\Sigma}^{T}_{k,1:k-1}, \tag{25}$$

for each $k = 2, \ldots, d$, in order, to get

$$\begin{aligned}\boldsymbol{\beta}_k^T \tilde{\boldsymbol{x}}_k &= \beta_{k,0} + (\beta_{k,1}, \ldots, \beta_{k,k-1})\boldsymbol{x}_1 \\ &= \mu_k + \boldsymbol{\Sigma}_{k,1:k-1}\boldsymbol{\Sigma}^{-1}_{1:k-1,1:k-1}(\boldsymbol{x}_{1:k-1} - \boldsymbol{\mu}_{1:k-1}) \\ &= \mu_{k|1:k-1}(\boldsymbol{x}_{1:k-1}),\end{aligned}$$

and $\sigma_k^2 = \Sigma_{k|1:k-1}$.

Now, by Lemma 1, and by definition of conditional densities,

$$\phi_1(x_1; \mu_1, \Sigma_{1,1}) \prod_{k=2}^{d} \phi_1(x_k; \mu_{k|1:k-1}(\boldsymbol{x}_{1:k-1}), \Sigma_{k|1:k-1}) = \phi_d(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

for all $\boldsymbol{x} \in \mathbb{R}^d$, which implies $\lambda(\boldsymbol{x}; \boldsymbol{\gamma}, \boldsymbol{\sigma}^2) = \phi_d(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ by application of the mappings (22)–(25). Note that $\boldsymbol{\mu}$ and vech($\boldsymbol{\Sigma}$), and $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}^2$ have equal numbers of elements, and (22)–(25) are unique for each k. Thus, there is an injective mapping between the LRC and the natural parameters. The inverse mapping can also be constructed by setting

$$\mu_1 = \beta_{0,1} \quad \text{and} \quad \Sigma_{1,1} = \sigma_1^2, \tag{26}$$

followed by

$$\boldsymbol{\Sigma}_{k,1:k-1} = (\beta_{k,1}, \ldots, \beta_{k,k-1})\boldsymbol{\Sigma}^{-1}_{1:k-1,1:k-1}, \tag{27}$$

$$\Sigma_{k,k} = \sigma_k^2 + \boldsymbol{\Sigma}_{k,1:k-1}\boldsymbol{\Sigma}^{-1}_{1:k-1,1:k-1}\boldsymbol{\Sigma}^{T}_{k,1:k-1}, \tag{28}$$

and

$$\mu_k = \beta_{k,0} + \boldsymbol{\Sigma}_{k,1:k-1}\boldsymbol{\Sigma}^{-1}_{1:k-1,1:k-1}\boldsymbol{\mu}_{1:k-1}, \tag{29}$$

for each $k = 2, \ldots, d$, in order. The mappings (26)–(29) are also unique for each k, and thus constitutes a surjective mapping.

Proof of Theorem 2

The first and last inequalities of (19) and (20) are due to the definition of minorization [i.e. (11) and (14) are of forms (9) and (10), respectively]. The middle inequality of

(19) is due to the concavity of $Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})$. This can be shown by firstly noting that

$$\sum_{i=1}^{g-1}\sum_{j=1}^{n}\tau_i(\boldsymbol{x}_j;\boldsymbol{\psi}^{(m)})\log(\pi_i)+\sum_{j=1}^{n}\tau_g(\boldsymbol{x}_j;\boldsymbol{\psi}^{(m)})\log\left(1-\sum_{i=1}^{g-1}\exp[\log(\pi_i)]\right)$$

is concave in $\log(\pi_i)$ since $1-\sum_{i=1}^{g-1}\exp[\log(\pi_i)]$ is concave and log is an increasing concave function. Secondly, note that

$$\frac{\partial^2 Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})}{\partial \beta_{i,k,l}^2} = -\frac{1}{2\sigma_{i,k}^{(m)2}}\sum_{j=1}^{n}\frac{x_{j,l}^2\tau_i(\boldsymbol{x}_j;\boldsymbol{\psi}^{(m)})}{\alpha_{j,l}}$$

is negative, and thus $Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})$ is concave with respect to each $\beta_{i,k,l}$ for each i, k and $l = 0, \ldots, k-1$. Thus, $Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})$ is the additive composition of concave functions and is therefore concave with respect to a bijection of $\boldsymbol{\psi}_1$. Furthermore, the system of equations

$$\frac{\partial Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})}{\partial \log(\pi_i)} = 0,$$

for $i = 1, \ldots, g-1$, has a unique root that is equivalent to update (16), which always satisfies the positivity restrictions on each π_i.

The middle inequality of (20) is due to the concavity of $Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2; \boldsymbol{\psi}^{(m)})$. This can be shown by noting that

$$-\frac{1}{2}\log\sigma_{i,k}^2\sum_{j=1}^{n}\tau(\boldsymbol{x}_j;\boldsymbol{\psi}^{(m)})-\frac{1}{2\exp[\log\sigma_{i,k}^2]}\sum_{j=1}^{n}\tau_i(\boldsymbol{x}_j;\boldsymbol{\psi}^{(m)})Q_{2,i,j,k}(\boldsymbol{\beta}_{i,k};\boldsymbol{\beta}_{i,k}^{(m)})$$

is concave in $\log\sigma_{i,k}^2$ for each i and k, since the inverse of $\exp(x)$ is convex. Thus, $Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2; \boldsymbol{\psi}^{(m)})$ is concave with respect to a bijection of $\boldsymbol{\psi}_2$. Furthermore, the system of equations

$$\frac{\partial Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2; \boldsymbol{\psi}^{(m)})}{\partial \log\sigma_{i,k}^2} = 0$$

has a unique root that is equivalent to update (18).

Proof of Theorem 3

This result follows from part (a) of Theorem 2 from Razaviyayn et al. (2013), which assumes that $Q'(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2^{(m)}; \boldsymbol{\psi}^{(m)})$ and $Q(\boldsymbol{\psi}_1^{(m)}, \boldsymbol{\psi}_2; \boldsymbol{\psi}^{(m)})$ both satisfy the definition

of a minorizer, and are quasi-concave and have unique critical points, with respect to the parameters $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$, respectively.

Firstly, the definition of a minorizer is satisfied via construction [i.e. (11) and (14) are of forms (9) and (10), respectively]. Secondly, from the proof of Theorem 2, both functions are concave with respect to some bijective mappings, and are therefore quasi-concave under said mappings [see Section 3.4 of Boyd and Vandenberghe (2004) regarding quasi-concavity]. Finally, since both functions are concave with respect to some bijective mapping, the critical points obtained must be unique.

Proof of Theorem 4

We show this result via induction. Firstly, using (26), we see that $\sigma_1^2 = \det(\Sigma_{1,1}) > 0$ is the first leading principal minor of $\boldsymbol{\Sigma}$, and is positive. Now, by definition of (23), σ_2^2 is the Schur complement of $\boldsymbol{\Sigma}_{1:k,1:k}$, for $k = 2$, where

$$\boldsymbol{\Sigma}_{1:k,1:k} = \begin{bmatrix} \boldsymbol{\Sigma}_{1:k-1,1:k-1} & \boldsymbol{\Sigma}_{k,1:k-1}^T \\ \boldsymbol{\Sigma}_{k,1:k-1} & \Sigma_{k,k} \end{bmatrix}. \tag{30}$$

Since σ_2^2 is positive and $\Sigma_{1,1}$ is positive definite, we have the result that

$$\det(\boldsymbol{\Sigma}_{1:2,1:2}) = \det(\Sigma_{1,1})\sigma_2^2 > 0$$

via the partitioning of the determinant. Thus, $\boldsymbol{\Sigma}_{1:2,1:2}$ is also positive definite because both the first and second leading principal minors are positive.

Now, for each $k = 3, \ldots, d$, we assume that $\boldsymbol{\Sigma}_{1:k-1,1:k-1}$ is positive-definite. Since $\sigma_k^2 > 0$ is the Schur complement of the partitioning (30), we have the result that

$$\det(\boldsymbol{\Sigma}_{1:k,1:k}) = \det(\boldsymbol{\Sigma}_{1:k-1,1:k-1})\sigma_k^2 > 0.$$

Thus, the kth leading principal minor is positive, for all k. The result follows by the property of positive-definite matrices; see Chapters 10 and 14 of Seber (2008) for all relevant matrix results.

Proof of Theorem 5

Theorem 5 can be established from Theorem 4.1.2 of Amemiya (1985), which requires the validation of the assumptions,

- A1 The parameter space Ψ is an open subset of some Euclidean space.
- A2 The log-likelihood $\log \mathcal{L}_{R,n}(\boldsymbol{\psi})$ is a measurable function for all $\boldsymbol{\psi} \in \Psi$, $\partial(\log \mathcal{L}_{R,n}(\boldsymbol{\psi}))/\partial\boldsymbol{\psi}$ exist and is continuous in an open neighborhood $N_1(\boldsymbol{\psi}^0)$ of $\boldsymbol{\psi}^0$.
- A3 There exists an open neighborhood $N_2(\boldsymbol{\psi}^0)$ of $\boldsymbol{\psi}^0$, where $n^{-1} \log \mathcal{L}_{R,n}(\boldsymbol{\psi})$ converges to $\mathbb{E}[\log f_R(\boldsymbol{X}; \boldsymbol{\psi})]$ in probability uniformly in $\boldsymbol{\psi}$ in any compact subset of $N_2(\boldsymbol{\psi}^0)$.

Assumptions A1, and A2 are fulfilled by noting that the parameter space $\Psi = (0,1)^{g-1} \times \mathbb{R}^{g(d^2+d)/2+gd}$ is an open subset of $\mathbb{R}^{(g-1)+g(d^2+d)/2+gd}$, and that $\log \mathcal{L}_{R,n}(\boldsymbol{\psi})$ is smooth with respect to the parameters $\boldsymbol{\psi}$. Using Theorem 2 of Jennrich (1969), we can show that A3 holds by verifying that

$$\mathbb{E} \sup_{\boldsymbol{\psi} \in \bar{N}} |\log f_R(\boldsymbol{X}; \boldsymbol{\psi})| < \infty, \tag{31}$$

where $\bar{N}$ is a compact subset of $N_2(\boldsymbol{\psi}^0)$. Since $f_R(\boldsymbol{X}; \boldsymbol{\psi})$ is smooth, this is equivalent to showing that $\mathbb{E}|f_R(\boldsymbol{X}; \boldsymbol{\psi})| < \infty$, for any fixed $\boldsymbol{\psi} \in \bar{N}$. This is achieved by noting that

$$\begin{aligned}
\mathbb{E}|\log f_R(\boldsymbol{X}; \boldsymbol{\psi})| &= \mathbb{E}|\log f_R(\boldsymbol{X}; \boldsymbol{\psi})| \\
&= \mathbb{E}\left|\log \sum_{i=1}^{g} \pi_i \lambda(\boldsymbol{x}; \boldsymbol{\gamma}_i, \boldsymbol{\sigma}_i^2)\right| \\
&\leq \sum_{i=1}^{g} \mathbb{E}|\log \lambda(\boldsymbol{x}; \boldsymbol{\gamma}_i, \boldsymbol{\sigma}_i^2)| \\
&= \sum_{i=1}^{g} \mathbb{E}\left|\sum_{k=1}^{d} \log \phi_1(x_k; \boldsymbol{\beta}_k^T \tilde{\boldsymbol{x}}_k, \sigma_k^2)\right| \\
&\leq \sum_{i=1}^{g} \sum_{k=1}^{d} \mathbb{E}|\log \phi_1(x_k; \boldsymbol{\beta}_{i,k}^T \tilde{\boldsymbol{x}}_k, \sigma_{i,k}^2)|. \qquad (32)
\end{aligned}$$

The inequality on line 3 of (32) is due to Lemma 1 of Atienza et al. (2007). Considering that $\log \phi_1(x_k; \boldsymbol{\beta}_{i,k}^T \tilde{\boldsymbol{x}}_k, \sigma_{i,k}^2)$ is a polynomial function of Gaussian random variables, we have $\mathbb{E}|\log \phi_1(x_k; \boldsymbol{\beta}_{i,k}^T \tilde{\boldsymbol{x}}_k, \sigma_{i,k}^2)| < \infty$ for each i and k. The result then follows.

Proof of Theorem 6

Theorem 6 can be established from Theorem 4.2.4 of Amemiya (1985), which requires the validation of the assumptions,

B1 The Hessian $\partial^2(\log \mathcal{L}_{R,n}(\boldsymbol{\psi}))/\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T$ exists and is continuous in an open neighborhood of $\boldsymbol{\psi}^0$.

B2 The equations

$$\int \frac{\partial \log f_R(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} \mathrm{d}\boldsymbol{x} = \mathbf{0},$$

and

$$\int \frac{\partial^2 \log f_R(\boldsymbol{\psi})}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathrm{d}\boldsymbol{x} = \mathbf{0},$$

hold, for any $\boldsymbol{\psi} \in \Psi$.

B3 The averaged Hessian satisfies

$$\frac{1}{n}\frac{\partial^2 \log \mathcal{L}_{R,n}(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}\partial \boldsymbol{\psi}^T} \xrightarrow{P} \mathbb{E}\left[\frac{\partial^2 \log f_R(\boldsymbol{X};\boldsymbol{\psi})}{\partial \boldsymbol{\psi}\partial \boldsymbol{\psi}^T}\right],$$

uniformly in $\boldsymbol{\psi}$, in all compact subsets of an open neighborhood of $\boldsymbol{\psi}^0$.

B4 The Fisher information

$$-\mathbb{E}\left[\frac{\partial^2 \log f_R(\boldsymbol{x};\boldsymbol{\psi})}{\partial \boldsymbol{\psi}\partial \boldsymbol{\psi}^T}\bigg|_{\boldsymbol{\psi}=\boldsymbol{\psi}^0}\right]^{-1},$$

is positive-definite.

Assumption B1 is validated via the smoothness of $\log \mathcal{L}_{R,n}(\boldsymbol{\psi})$, and it is mechanical to check the validity of B2. Assumption B3 can be shown via Theorem 2 of Jennrich (1969). Unlike the others, B4 must be taken as given.

References

Amemiya T (1985) Advanced econometrics. Harvard University Press, Cambridge

Anderson TW (2003) An introduction to multivariate statistical analysis. Wiley, New York

Andrews JL, McNicholas PD (2013) Using evolutionary algorithms for model-based clustering. Pattern Recognit Lett 34:987–992

Atienza N, Garcia-Heras J, Munoz-Pichardo JM, Villa R (2007) On the consistency of MLE in finite mixture models of exponential families. J Stat Plan Inference 137:496–505

Becker MP, Yang I, Lange K (1997) EM algorithms without missing data. Stat Methods Med Res 6:38–54

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York

Botev Z, Kroese DP (2004) Global likelihood optimization via the cross-entropy method with an application to mixture models. In: Proceedings of the 36th conference on winter simulation

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Celeux G, Govaert G (1992) A classification EM algorithm for clustering and two stochastic versions. Comput Stat Data Anal 14:315–332

Clarke B, Fokoue E, Zhang HH (2009) Principles and theory for data mining and machine learning. Springer, New York

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B 39:1–38

Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New York

Ganesalingam S, McLachlan GJ (1980) A comparison of the mixture and classification approaches to cluster analysis. Commun Stat Theory Methods 9:923–933

Greselin F, Ingrassia S (2008) A note on constrained EM algorithms for mixtures of elliptical distributions. Advances in data analysis, data handling and business intelligence In: Proceedings of the 32nd annual conference of the German classification society. vol 53

Hartigan JA (1985) Statistical theory in clustering. J Classif 2:63–76

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning. Springer, New York

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

Hunter DR, Lange K (2004) A tutorial on MM algorithms. Am Stat 58:30–37

Ingrassia S (1991) Mixture decomposition via the simulated annealing algorithm. Appl Stoch Models Data Anal 7:317–325

Ingrassia S (2004) A likelihood-based constrained algorithm for multivariate normal mixture models. Stat Methods Appl 13:151–166

Ingrassia S, Rocci R (2007) Constrained monotone EM algorithms for finite mixture of multivariate Gaussians. Comput Stat Data Anal 51:5339–5351
Ingrassia S, Rocci R (2011) Degeneracy of the EM algorithm for the MLE of multivariate Gaussian mixtures and dynamic constraints. Comput Stat Data Anal 55:1714–1725
Ingrassia S, Minotti SC, Vittadini G (2012) Local statistical modeling via a cluster-weighted approach with elliptical distributions. J Classif 29:363–401
Ingrassia S, Minotti SC, Punzo A (2014) Model-based clustering via linear cluster-weighted models. Comput Stat Data Anal 71:159–182
Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recognit Lett 31:651–666
Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31:264–323
Jennrich RI (1969) Asymptotic properties of non-linear least squares estimators. Ann Math Stat 40:633–643
MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkley symposium on mathematical statistics and probability, University of California press, 281–297
McLachlan GJ (1982) The classification and mixture maximum likelihood approaches to cluster analysis. In: Krishnaiah PR, Kanal L (eds) Handbook of statistics, vol 2. North-Holland, Amsterdam
McLachlan GJ, Basford KE (1988) Mixture models: inference and applications to clustering. Marcel Dekker, New York
McLachlan GJ, Peel D (2000) Finite mixture models. Wiley, New York
McLachlan GJ, Krishnan T (2008) The EM algorithm and extensions. Wiley, New York
Pernkopf F, Bouchaffra D (2005) Genetic-based EM algorithm for learning Gaussian mixture models. IEEE Trans Pattern Anal Mach Intell 27:1344–1348
R Core Team (2013) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna
Razaviyayn M, Hong M, Luo ZQ (2013) A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM J Optim 23:1126–1153
Redner RA, Walker HF (1984) Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev 26:195–239
Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
Seber GAF (2008) A matrix handbook for statisticians. Wiley, New York
Titterington DM, Smith AFM, Makov UE (1985) Statistical analysis of finite mixture distributions. Wiley, New York
Zhou H, Lange K (2010) Mm algorithms for some discrete multivariate distributions. J Comput Graph Stat 19:645–665