Maximum likelihood estimation in constrained parameter spaces for mixtures of factor analyzers

Francesca Greselin · Salvatore Ingrassia

Received: 3 January 2013 / Accepted: 10 October 2013 / Published online: 8 November 2013
© Springer Science+Business Media New York 2013

Abstract Mixtures of factor analyzers are becoming more and more popular in the area of model based clustering of multivariate data. According to the likelihood approach in data modeling, it is well known that the unconstrained likelihood function may present spurious maxima and singularities. To reduce such drawbacks, in this paper we introduce a procedure for parameter estimation of mixtures of factor analyzers, which maximizes the likelihood function under the mild requirement that the eigenvalues of the covariance matrices lie into some interval $[a,b]$. Moreover, we give a recipe on how to select appropriate bounds for the constrained EM algorithm, directly from the handled data. We then analyze and measure its performance, compared with the usual non-constrained approach, and also with other constrained models in the literature. Results show that the data-driven constraints improve the estimation and the subsequent classification, at the same time.

Keywords Constrained estimation · Factor analyzers modeling · Mixture models · Model-based clustering

1 Introduction and motivation

Finite mixture distributions have been receiving a growing interest in statistical modeling. Their central role is mainly due to their double nature: they combine the flexibility of non-parametric models with the strong and useful mathematical properties of parametric models. According to this approach, when we know that a sample of observations has been drawn from different populations, we assume a specific distributional form in each of the underlying populations. The purpose is to decompose the sample into its mixture components—which, for quantitative data, are usually modeled as a multivariate Gaussian distribution—and to estimate parameters. The assumption of underlying normality, besides the elegant analytic properties, allows also to employ the EM algorithm for the ML estimation of the parameters. On the other side, when considering a large number of observed variables, Gaussian mixture models can provide an over-parameterized solution as, in addition to the mixing weights, it is required to estimate the mean vector and the covariance matrix for each component. As a consequence, we observe an undue load of computationally intensive procedures for the estimation.

This is the reason why a number of strategies have been introduced in the literature to avoid over-parameterized solutions. Among the various proposals, some authors developed methodologies for variable selection (see, e.g., Liu et al. (2003) and Hoff (2005) in the Bayesian framework, Pan and Shen (2007) and Raftery and Dean (2006) in the frequentist one). They further motivate their approach from the observation that the presence of non-informative variables can be strongly misleading for some clustering methods. With the same purpose of parsimony, but following a completely different approach, Banfield and Raftery (1993) devised a methodology to identify common patterns among the component-covariance matrices; their proposal arose a great attention in the literature. Along a slightly different line of thinking, Ghahramani and Hilton (1997) and McLachlan and Peel (2000b), proposed to employ latent variables
to perform dimensional reduction in each component, starting from the consideration that, in many phenomena, many observed features could be explained by a few unobserved ones. This provides a statistical method which concurrently performs clustering and, within each cluster, local dimensionality reduction.

In this paper, we address mixtures of factor analyzers by assuming that the data have been generated by a linear factor model with latent variables modeled as Gaussian mixtures. Our purpose is to improve the performance of the EM algorithm, dealing with some of its issues and giving practical recipes to overcome them. It is well known that the EM algorithm generates a sequence of estimates, starting from an initial guess, so that the corresponding sequence of the log-likelihood values is not decreasing. However, the convergence towards the MLE is not guaranteed, because the log-likelihood is unbounded and presents local maxima. An initial guess, so that the corresponding sequence of the algorithm generates a sequence of estimates, starting from an initial guess, so that the corresponding sequence of the log-likelihood values is not decreasing. However, the convergence towards the MLE is not guaranteed, because the log-likelihood is unbounded and presents local maxima.

In the literature about mixtures of Gaussian factors (MFA), with the aim of avoiding spurious local maximizers and singularities, some authors propose to take a common (diagonal) error matrix (as for Mixtures of Common Factor Analyzers, denoted by MCFA, in Baek et al. 2010) or to impose an isotropic error matrix (Bishop and Tippin 1998; McNicholas and Murphy 2008). This strategy has proven to be effective in many cases, at the expenses of stronger distributional restrictions on the data. Quoting Baek et al. (2010): “As the MFA approach allows a more general representation of the component-covariance matrices and places no restrictions on the component means, it is in this sense preferable to the MCFA approach if its application is feasible given the values of p (original dimension of the dataset) and g (number of groups).” Our proposal here is to introduce and implement a procedure for parameter estimation of mixtures of factor analyzers, which maximizes the likelihood function under the mild requirement that the eigenvalues of the covariance matrices lie into some interval [a, b], along the lines of Ingrassia (2004). This assures that the likelihood does not have singularities and reduces the number of its spurious local maxima. Then, we analyze and measure its performance, compared to the usual non-constrained approach, and also with the previously mentioned models, say MCFA and PGMMs ( Parsimonious Gaussian Mixture Models in McNicholas and Murphy 2008). Moreover, we give a recipe on how to select appropriate bounds for the constrained EM algorithm, directly from the handled data.

We have organized the rest of the paper as follows. In Sect. 2 we summarize main ideas about Gaussian Mixture of Factor Analyzers model; in Sect. 3 we provide fairly extensive notes concerning the likelihood function and the AECM algorithm. Some well known considerations (Hathaway 1985) related to spurious maximizers and singularities in the EM algorithm are recalled in Sect. 4, and motivate our proposal to introduce constraints on factor analyzers. Further, we give a detailed methodology to implement such constraints into the EM algorithm. In Sect. 5 we show and discuss the improved performance of our procedure, on the ground of some numerical results obtained from simulated and real data. Section 6 contains concluding notes and provides ideas for future research.

### 2 The Gaussian Mixture of Factor Analyzers

Within the Gaussian Mixture (GM) model-based approach to density estimation and clustering, the density of the d-dimensional random variable \( \mathbf{X} \) of interest is modeled as a mixture of a number, say \( G \), of multivariate normal densities in some unknown proportions \( \pi_1, \ldots, \pi_G \). That is, each data point is taken to be a realization of the mixture probability density function,

\[
  f(\mathbf{x}; \theta) = \sum_{g=1}^{G} \pi_g \phi_d(\mathbf{x}; \mu_g, \Sigma_g)
\]

where \( \phi_d(\mathbf{x}; \mu, \Sigma) \) denotes the d-variate normal density function with mean vector \( \mu \) and covariance matrix \( \Sigma \). Here the vector \( \theta = \theta_{GM}(d, G) \) of unknown parameters consists of \( (G - 1) \) mixing proportions \( \pi_g \), the \( Gd \) elements of the component means \( \mu_g \), and the \( \frac{1}{2}Gd(d + 1) \) distinct elements of the component-covariance matrices \( \Sigma_g \). Therefore, the G-component normal mixture model (1) is a highly parameterized model. We crucially need some method for parsimonious parametrization of the matrices \( \Sigma_g \), requiring \( O(d^2) \) parameters. Among the various proposals for dimensionality reduction, we are interested here in considering Gaussian Mixtures of Factor Analyzers (MFA), which allow to explain data by explicitly modeling correlations between variables in multivariate observations. We postulate a finite mixture of linear sub-models for the distribution of the full observation vector \( \mathbf{X} \), given the (unobservable) factors \( \mathbf{U} \). That is, we provide a local dimensionality reduction method by assuming that the distribution of the observation \( \mathbf{X}_i \) can be given as

\[
  \mathbf{X}_i = \mu_g + A_g \mathbf{U}_{ig} + \mathbf{e}_{ig} \quad \text{with probability } \pi_g \quad (g = 1, \ldots, G) \text{ for } i = 1, \ldots, n,
\]

where \( A_g \) is a \( d \times q \) matrix of factor loadings, the factors \( \mathbf{U}_{1g}, \ldots, \mathbf{U}_{ng} \) are \( \mathcal{N}(0, I_q) \) distributed independently of the errors \( \mathbf{e}_{ig} \), which are independently \( \mathcal{N}(0, \Psi_g) \) distributed, and \( \Psi_g \) is a \( d \times d \) diagonal matrix \( (g = 1, \ldots, G) \). The diagonality of \( \Psi_g \) is one of the key assumptions of factor analysis: the observed variables are independent given the factors. Note that the factor variables \( \mathbf{U}_{ig} \) model correlations between the elements of \( \mathbf{X}_i \), while the \( \mathbf{e}_{ig} \) variables account for independent noise for \( \mathbf{X}_i \). We suppose that \( q < d \), which...
The relative reduction to give an idea of the gain in parsimony when dealing with (1), where the g-th component-covariance matrix $\Sigma_g$ has the form

$$\Sigma_g = A_g A_g' + \Psi_g \quad (g = 1, \ldots, G). \quad (3)$$

The parameter vector $\theta = \theta_{MFA}(d, q, G)$ now consists of the elements of the component means $\mu_g$, the $A_g$, and the $\Psi_g$, along with the mixing proportions $\pi_g$ ($g = 1, \ldots, G - 1$), on putting $\pi_G = 1 - \sum_{i=1}^{G-1} \pi_g$. Note that, in the case of $q > 1$, there is an infinity of choices for $A_g$, since model (2) is still satisfied if we replace $A_g$ by $A_g H'$, where $H$ is any orthogonal matrix of order $q$. As $q(q - 1)/2$ constraints are needed for $A_g$ to be uniquely defined, the number of free parameters for each component of the mixture, is

$$dq + d - \frac{1}{2}q(q - 1).$$

Denoting by $|\theta_{CovGM}(d, G)|$ and $|\theta_{CovMFA}(d, q, G)|$ the number of the estimated parameters for the covariance matrices in the GM and MFA models, respectively, we get

$$|\theta_{CovGM}(d, G)| - |\theta_{CovMFA}(d, q, G)| = G\frac{d(d + 1)}{2} - G\left[ dq + d - \frac{1}{2}q(q - 1) \right] > 0$$

i.e.:

$$G\frac{1}{2}(d - q)^2 - (d + q) > 0.$$ 

The relative reduction $RR(d, q, G) = RR(d, q)$, in terms of number of covariance parameters, is given by

$$RR(d, q) = \frac{|\theta_{CovGM}(d, G)| - |\theta_{CovMFA}(d, q, G)|}{|\theta_{CovGM}(d, G)|} = \frac{(d - q)^2 - (d + q)}{d(d + 1)}. $$

Table 1 provides some values of the relative reduction, to give an idea of the gain in parsimony when dealing with MFA, in comparison to GM models.

### Table 1: Relative reduction $RR(d, q)$

| $q/d$ | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 15  | 20  | 30  | 50  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1     | --  | 0.20| 0.33| 0.43| 0.50| 0.56| 0.60| 0.64| 0.75| 0.81| 0.87| 0.92|
| 2     | --  | --  | 0.07| 0.19| 0.29| 0.36| 0.42| 0.47| 0.63| 0.72| 0.81| 0.88|
| 3     | --  | --  | --  | --  | 0.11| 0.19| 0.27| 0.33| 0.53| 0.63| 0.75| 0.85|
| 4     | --  | --  | --  | --  | --  | 0.06| 0.13| 0.20| 0.43| 0.55| 0.69| 0.81|
| 5     | --  | --  | --  | --  | --  | --  | 0.02| 0.09| 0.33| 0.48| 0.63| 0.77|

#### 3 The likelihood function and the EM algorithm for MFA

In this section we summarize the main steps of the EM algorithm for mixtures of factor analyzers, see e.g. McLachlan and Peel (2000a) for details.

Let $S = (x_1, \ldots, x_n)$ be a sample of size $n$ from density (1). For given data $S$, parameters in (1) can be estimated according to the likelihood approach via the EM algorithm, where the likelihood function is given by:

$$L(\theta; S) = \prod_{i=1}^{n} \left\{ \sum_{g=1}^{G} \phi_d(x_i; \mu_g, \Sigma_g) \pi_g \right\} = \prod_{i=1}^{n} \left\{ \sum_{g=1}^{G} \phi_d(x_i; \mu_g, A_g, \Psi_g) \pi_g \right\},$$

and we set $\Sigma_g = A_g A_g' + \Psi_g \quad (g = 1, \ldots, G)$. Note that $L(\theta; S)$ diverges as $\mu_g \rightarrow x_i$ and $\det(\Psi_g) \rightarrow 0$ if $q < p$, because $A_g A_g'$ are singular.

In the EM algorithm framework, the generic observation $x_i$ is viewed as being incomplete; its complete counterpart is given by $(x_i, u_{ig}, z_{ig})$, $i = 1, \ldots, n$, where $z_{ig} = (z_{i1}, \ldots, z_{iq})'$, with $z_{ig} = 1$ if $x_i$ comes from the $g$-th population and $z_{ig} = 0$ otherwise. Then, the complete-data likelihood function can be written in the form:

$$L_c(\theta; S) = \prod_{i=1}^{n} \prod_{g=1}^{G} \phi_d(x_i; u_{ig}, A_g, \Psi_g) \phi_q(u_{ig}) \pi_g^{z_{ig}}.$$ 

In particular, due to the factor structure of the model, we have to consider the alternating expectation-conditional maximization (AECM) algorithm, see Meng and van Dyk (1997). Such a procedure is an extension of the EM algorithm that uses different specifications of missing data at each stage. The idea is to partition $\theta = (\theta_1, \theta_2)'$ in such a way that $L(\theta; S)$ is easy to maximize for $\theta_1$ given $\theta_2$ and vice versa. In more detail, $\theta_1 = \{ \pi_g, \mu_g, g = 1, \ldots, G \}$ where the missing data are the unobserved group labels $Z = (z_{i1}, \ldots, z_{iq})$, and $\theta_2 = \{ (A_g, \Psi_g), g = 1, \ldots, G \}$ where the missing data are the group labels $Z$ and the unobserved latent factors $U = (U_{11}, \ldots, U_{nG})$. Hence, the AECM algorithm consists of two cycles, and there is one E-step and one CM-step alternatively considering $\theta_1$ and $\theta_2$ in each pair of
cycles. Before describing the algorithm, we underline that in both cycles the unobserved group labels $Z = (z_1', \ldots, z_n')$ are considered missing data. Therefore, on the $k$-th iteration, we shall denote by $z_{ig}^{(k+1/2)}$ and $z_{ig}^{(k+1)}$ the conditional expectations at the first and second cycle, respectively.

First cycle Here we set $\theta_1 = \{\pi_g, \mu_g, g = 1, \ldots, G\}$ where the unobserved group labels $Z = (z_1', \ldots, z_n')$ are the missing data. The complete data likelihood is

$$L_{c1}(\theta_1) = \prod_{i=1}^n \prod_{g=1}^G [\phi_d(x_i; \mu_g, \Sigma_g) \pi_g]^{z_{ig}}.$$ \hspace{1cm} (4)

The E-step on the first cycle on the $(k+1)$-th iteration requires the calculation of $Q_1(\theta_1; \theta^{(k)}) = \mathbb{E}_{\theta^{(k)}}[L_{c1}(\theta_1) | S]$ which is the expected complete-data log-likelihood given the data $S$ and using the current estimate $\theta^{(k)}$ for $\theta$. In practice, it requires calculating $\mathbb{E}_{\theta^{(k)}}[Z_{ig} | S]$ and usual computations show that this step is achieved by replacing each $z_{ig}$ by its current conditional expectation given the observed data $x_i$, that is we replace $z_{ig}$ by $z_{ig}^{(k+1/2)}$, where

$$z_{ig}^{(k+1/2)} = \mathbb{E}_{\theta^{(k)}}[Z_{ig} | S] = \frac{\phi_d(x_i; \mu_g^{(k)}, A_g^{(k)}, \Psi_g^{(k)}) \pi_g^{(k)}}{\sum_{g=1}^G \phi_d(x_i; \mu_g^{(k)}, A_g^{(k)}, \Psi_g^{(k)}) \pi_g^{(k)}}.$$ \hspace{1cm} (5)

On the M-step, the maximization of the complete-data likelihood in (4) yields

$$z_{ig}^{(k+1)} = \frac{1}{n^{(k+1/2)}} \sum_{i=1}^n z_{ig}^{(k+1/2)} x_i,$$

$$\mu_g^{(k+1)} = \mathbb{E}_{\theta^{(k)}}[Z_{ig} | S] = \frac{\phi_d(x_i; \mu_g^{(k)}, A_g^{(k)}, \Psi_g^{(k)}) \pi_g^{(k)}}{\sum_{g=1}^G \phi_d(x_i; \mu_g^{(k)}, A_g^{(k)}, \Psi_g^{(k)}) \pi_g^{(k)}}.$$ \hspace{1cm} (5)

where $n^{(k+1/2)} = \sum_{i=1}^n z_{ig}^{(k+1/2)}$. According to notation in McCulloch and Peel (2000a), we set $\theta^{(k+1/2)} = (\theta^{(k)}, \theta^{(k)}')'$.

Second cycle Here $\theta_2 = \{\Sigma_g, g = 1, \ldots, G\} = \{A_g, \Psi_g\}, g = 1, \ldots, G$ where the missing data are the unobserved group labels $Z$ and the latent factors $U$. The complete data likelihood is

$$L_{c2}(\theta_2) = \prod_{i=1}^n \prod_{g=1}^G [\phi_d(x_i; \mu_g^{(k+1)}, \Sigma_g) \phi_q(u_{ig}; \nu_g) \pi_g^{(k+1)}]^{z_{ig}} \times \phi_q(u_{ig}; \nu_g) \pi_g^{(k+1)}.$$ \hspace{1cm} (6)

where

$$\phi_d(x_i; \mu_g^{(k+1)}, A_g, \Psi_g) = \frac{1}{(2\pi)^{d/2} \Psi_g} \exp \left\{ -\frac{1}{2} (x_i - \mu_g^{(k+1)} - A_g u_{ig})' \Psi_g^{-1} (x_i - \mu_g^{(k+1)} - A_g u_{ig}) \right\},$$

$$\phi_q(u_{ig}; \nu_g) = \frac{1}{(2\pi)^{q/2} \nu_g} \exp \left\{ -\frac{1}{2} u_{ig}' u_{ig} \right\}.$$ 

Now the complete data log-likelihood is given by

$$L_{c2}(\theta_2) = \mathbb{E}_{\theta^{(k+1/2)}}[L_{c2}(\theta_2) | S] = \mathbb{E}_{\theta^{(k+1/2)}}[\mathbb{E}_{\theta^{(k+1/2)}}[L_{c2}(\theta_2) | S] | S].$$ \hspace{1cm} (7)

The E-step on the second cycle on the $(k+1)$-th iteration requires the calculation of $\mathbb{E}_{\theta^{(k+1/2)}}[L_{c2}(\theta_2) | S]$ which denotes the conditional expectation of (5) given the observed data $S$ and using $\theta^{(k+1/2)}$ for $\theta$. Therefore, we must calculate the following conditional expectations

$$z_{ig}^{(k+1)} = \text{argmax}_{z_{ig}}\{L_{c2}(\theta_2) | S\} = \frac{\phi_d(x_i; \mu_g^{(k+1)}, A_g^{(k)}, \Psi_g^{(k)}) \pi_g^{(k+1)}}{\sum_{g=1}^G \phi_d(x_i; \mu_g^{(k+1)}, A_g^{(k)}, \Psi_g^{(k)}) \pi_g^{(k+1)}}.$$ \hspace{1cm} (5)

$$\mathbb{E}_{\theta^{(k+1/2)}}[Z_{ig} | U_{ig} | S] = z_{ig}^{(k+1)} \gamma_g (x_i - \mu_g^{(k+1)})$$

$$\mathbb{E}_{\theta^{(k+1/2)}}[Z_{ig} | U_{ig} | S] = z_{ig}^{(k+1)} \Theta_g (k),$$

where

$$\gamma_g = A_g^{(k)} (A_g^{(k)} A_g^{(k)}' + \Psi_g^{(k)})^{-1}$$

$$\Theta_g = I_n - \gamma_g (k) A_g^{(k)} + \gamma_g (k) S_g^{(k-1)} \gamma_g (k)'$$

Now, the $g$-th term of the expected complete-data log-likelihood $Q_2(\theta_2; \theta^{(k+1/2)})$ becomes
\[
Q_2(\mathbf{A}_g, \Psi_g) = C(\theta_1^{(k+1)}) - \frac{1}{2} n_g^{(k+1)} \ln |\Psi_g^{-1}| \\
- \frac{1}{2} n_g^{(k+1)} \text{tr} \left[ S_g^{(k+1)} \Psi_g^{-1} \right] \\
+ n_g^{(k+1)} \text{tr} \left[ A_g \Psi_g^{(k+1)} S_g^{(k+1)} \Psi_g^{-1} \right] \\
- \frac{1}{2} n_g^{(k+1)} \text{tr} \left[ A_g' \Psi_g^{-1} A_g \Theta_g \right] 
\]

where \(C(\theta_1^{(k+1)})\) denotes the terms that do not depend on \(\theta_2\). Then, in the second M-step (6) is maximized for \((\mathbf{A}_g, \Psi_g)\) satisfying

\[
\frac{\partial Q_2}{\partial \mathbf{A}_g} = n_g^{(k+1)} \Psi_g^{-1} S_g^{(k+1)} \gamma_g^{(k)} - n_g^{(k+1)} \Psi_g^{-1} A_g \theta_g^{(k)} = 0
\]

\[
\frac{\partial Q_2}{\partial \Psi_g} = \frac{1}{2} n_g^{(k+1)} \Psi_g - \frac{1}{2} n_g^{(k+1)} S_g^{(k+1)} \\
+ n_g^{(k+1)} S_g^{(k)} \gamma_g^{(k)} A_g' - \frac{1}{2} n_g^{(k+1)} A_g \theta_g A_g' = 0.
\]

Some algebras lead to the following estimate of \(\{(\mathbf{A}_g, \Psi_g), g = 1, \ldots, G\}\)

\[
\hat{\mathbf{A}}_g = S_g^{(k+1)} \gamma_g^{(k)} \left[ \Theta_g^{(k)} \right]^{-1}
\]

\[
\hat{\Psi}_g = \text{diag} \left[ S_g^{(k+1)} - \Lambda_g \gamma_g^{(k)} S_g^{(k+1)} \right].
\]

Hence the maximum likelihood estimates \(\hat{\mathbf{A}}_g\) and \(\hat{\Psi}_g\) for \(\mathbf{A}\) and \(\Psi\) can be obtained by alternatively computing the update estimates \(\Lambda_g^+\) and \(\Psi_g^+\) by

\[
\Lambda_g^+ = S_g^{(k+1)} \gamma_g^{(k)} \left[ \Theta_g^{(k)} \right]^{-1}
\]

\[
\Psi_g^+ = \text{diag} \left[ S_g^{(k+1)} - \Lambda_g^+ \gamma_g^{(k)} S_g^{(k+1)} \right].
\]

and, from the latter, evaluating the update estimates \(\gamma_g^+\) and \(\Theta_g^+\) by

\[
\gamma_g^+ = \Lambda_g^+ \left( \Lambda_g^+ \gamma_g^+ + \Psi_g^+ \right)^{-1}
\]

\[
\Theta_g^+ = I_g - \gamma_g^+ A_g + \Psi_g^+ S_g^{(k+1)} \gamma_g^+.
\]

iterating steps (7) and (8) until convergence on \(\hat{\mathbf{A}}_g\) and \(\hat{\Psi}_g\), so giving \(\Lambda_g^{(k+1)}\) and \(\Psi_g^{(k+1)}\).

In summary, the procedure can be described as follows. For a given initial random clustering \(z^{(0)}\), on the \((k+1)\)-th iteration, the algorithm carries out the following steps, for \(g = 1, \ldots, G\):

**Procedure 1 (EM algorithm for MFA)**

1. Compute \(z_g^{(k+1)}\) and consequently obtain \(\pi_g^{(k+1)}, \mu_g^{(k+1)}, n_g^{(k+1)}\) and \(S_g^{(k+1)}\);

2. Set a starting value for \(\mathbf{A}_g\) and \(\Psi_g\) from \(S_g^{(k+1)}\);

3. Repeat the following steps, until convergence on \(\hat{\mathbf{A}}_g\) and \(\hat{\Psi}_g\):

   (a) Compute \(\gamma_g^+\) and \(\Theta_g^+\) from (8);

   (b) Set \(\gamma_g \leftarrow \gamma_g^+\) and \(\Theta_g \leftarrow \Theta_g^+\);

   (c) Compute \(A_g^+ \leftarrow S_g^{(k+1)} \gamma_g^+ \left( \Theta_g^{-1} \right)\) and \(\Psi_g^+ \leftarrow\)

   \[\text{diag} \left( S_g^{(k+1)} - A_g^+ \gamma_g^+ S_g^{(k+1)} \right)\];

   (d) Set \(A_g \leftarrow A_g^+\) and \(\Psi_g \leftarrow \Psi_g^+\);

To completely describe the algorithm, here we give more details on how to specify the starting values for \(\mathbf{A}_g\) and \(\Psi_g\) from \(S_g^{(k+1)}\), as it is needed in Step 2.

Starting from the eigen-decomposition of \(S_g^{(k+1)}\), say \(S_g^{(k+1)} = W_g \Sigma_g W_g\), computed on the base of \(z_g^{(k+1)}\), the main idea is that \(\mathbf{A}_g\) has to “capture” the “more important” relations between the \(d\) observed features, see McNicholas and Murphy (2008). Therefore, looking at the equality \(\Sigma_g = \Gamma_g \Gamma_g' + \Psi_g\), the initial values of \(\mathbf{A}_g\) can be set as

\[
\lambda_{ij} = \sqrt{l_j} w_{ij}
\]

where \(l_j\) is the \(j\)-th largest eigenvalue of \(S_g^{(k+1)}\) and \(w_{ij}\) is the \(i\)-th element of the corresponding eigenvector \(w_j\) (the \(j\)-th column in \(W_g\)), for \(i \in \{1, 2, \ldots, p\}\) and \(j \in \{1, 2, \ldots, q\}\). Finally the \(\Psi_g\) matrices can be initialized by the position \(\Psi_g = \text{diag} \left( S_g^{(k+1)} - \mathbf{A}_g \mathbf{A}_g' \right)\).

### 4 Likelihood maximization in constrained parametric spaces

Properties of maximum likelihood estimation for normal mixture models have been deeply investigated. It is well known that \(L(\theta)\) is unbounded on \(\Theta\) and may present many local maxima. Day (1969) was perhaps the first noting that any small number of sample points, grouped sufficiently close together, or almost collinear, can give rise to spurious maximizers, corresponding to parameter points with greatly differing component standard deviation. To overcome this issue and to prevent \(L(\theta)\) from singularities, Hathaway (1985) proposed a constrained maximum likelihood formulation for mixtures of univariate normal distributions, suggesting a natural extension to the multivariate case. Let \(c \in (0, 1]\), then the following constraints

\[
\min_{1 \leq h \neq j \leq k} \lambda \left( \Sigma_h \Sigma_j^{-1} \right) \geq c
\]

on the eigenvalues \(\lambda\) of \(\Sigma_h \Sigma_j^{-1}\) leads to properly defined, scale-equivariant, consistent ML-estimators for the mixture-of-normal case, see Hennig (2004). It can be proved that a
sufficient condition for (10) is
\[ a \leq \lambda_{ig} \leq b, \quad i = 1, \ldots, d; \quad g = 1, \ldots, G \] (11)
where \( \lambda_{ig} \) denotes the \( i \)-th eigenvalue of \( \Sigma_g \) i.e. \( \lambda_{ig} = \lambda_i(\Sigma_g) \), and for \( a, b \in \mathbb{R}^+ \) such that \( a/b \geq c \), see Ingrassia (2004). Differently from (10), condition (11) can be easily implemented, also in the wider context of mixtures of \( t \) distributions, see Greleng (2010) for details. At the same time, it still keeps the algorithm far from singularities and avoids huge imbalances between the estimated component covariances. We recall that the latter is a tell-tale sign for spurious maximizers, when observed jointly with very small values for the mixing proportions.

Let us consider the constrained parameter space of \( \Theta_g \):
\[
\Theta_g = \{(\pi_1, \ldots, \pi_G, \mu_1, \ldots, \mu_G, \Sigma_1, \ldots, \Sigma_G) \in \mathbb{R}^{(1+d+(d^2+d)/2)} : \pi_g \geq 0, \pi_1 + \cdots + \pi_G = 1, a \leq \lambda_{ig} \leq b, \quad g = 1, \ldots, G, \quad i = 1, \ldots, d \}. \]
Due to the structure of the covariance matrix \( \Sigma_g \) given in (3), bound in (11) yields
\[ a \leq \lambda(\Lambda_g \Lambda'_g + \Psi_g) \leq b, \quad g = 1, \ldots, G. \] (12)

Concerning the square \( d \times d \) matrix \( \Lambda_g \Lambda'_g \) \((g = 1, \ldots, G)\), we can get its eigenvalue decomposition, i.e. we can find \( \Lambda_g \) and \( \Gamma_g \) such that
\[ \Lambda_g \Lambda'_g = \Gamma_g \Lambda_g \Gamma'_g \] (13)
where \( \Gamma_g \) is the orthonormal matrix whose columns are the eigenvectors of \( \Lambda_g \Lambda'_g \) and \( \Lambda_g = \text{diag}(\delta_{1g}, \ldots, \delta_{dg}) \) is the diagonal matrix of the eigenvalues of \( \Lambda_g \Lambda'_g \), sorted in non increasing order, i.e. \( \delta_{1g} \geq \delta_{2g} \geq \cdots \geq \delta_{dg} \geq 0 \), and \( \delta_{(q+1)g} = \cdots = \delta_{dg} = 0 \).

Now, let us consider the singular value decomposition of the \( d \times q \) rectangular matrix \( \Lambda_g \), so giving \( \Lambda_g = U_g D_g V'_g \), where \( U_g \) is a \( d \times d \) unitary matrix (i.e., such that \( U'_g U_g = I_d \)) and \( D_g \) is a \( d \times q \) rectangular diagonal matrix with \( q \) nonnegative real numbers on the diagonal, known as singular values, and \( V_g \) is a \( q \times q \) unitary matrix. The \( d \) columns of \( U_g \) and the \( q \) columns of \( V_g \) are called the left singular vectors and right singular vectors of \( \Lambda_g \), respectively. Now we have that
\[ \Lambda_g \Lambda'_g = (U_g D_g V'_g)(V'_g D'_g U'_g) = U_g D_g I_d D'_g U'_g \]
where
\[ D_g = U_g D_g V'_g \]
and equating (13) and (14) we get \( \Gamma_g = U_g \) and \( \Lambda_g = D_g \), that is
\[ \text{diag}(\delta_{1g}, \ldots, \delta_{dg}) = \text{diag}(d^2_{1g}, \ldots, d^2_{dg}), \] (15)
with \( d_{1g} \geq d_{2g} \geq \cdots \geq d_{dg} \geq 0 \). In particular, it is known that only the first \( q \) values of \( D_g \) are non negative, and the remaining \( d - q \) terms are null.

Denoting now by \( \psi_{ig} \) the \( i \)-th eigenvalue of \( \Psi_g \) then constraint (12) is satisfied when
\[ d^2_{ig} + \psi_{ig} \geq a \quad i = 1, \ldots, d \] (16)
\[ d^2_{ig} \leq \sqrt{b^2 - \psi_{ig}} \quad i = 1, \ldots, q \] (17)
\[ \psi_{ig} \leq b \quad i = q + 1, \ldots, d \] (18)
for \( g = 1, \ldots, G \). In particular, we remark that condition (16) reduces to \( \psi_{ig} \geq a \) for \( i = (q + 1), \ldots, d \).

The two-fold (eigenvalue and singular value) decomposition of the \( \Lambda_g \) presented above, suggests how to modify the EM algorithm in such a way that the eigenvalues of the covariances \( \Sigma_g \) \((g = 1, \ldots, G)\) are confined into suitable ranges. To this aim we have to implement constraints (16), (17) and (18). Procedure 2 summarizes the steps to be performed on the \((k + 1)\)-th iteration, when dealing with group \( g \), for \( g = 1, \ldots, G \).

Procedure 2 (Constrained EM algorithm for MFA)
1. Decompose \( \Lambda_g \) according to the singular value decomposition as \( \Lambda_g = U_g D_g V'_g \);
2. Create a copy \( D_g \) of \( D_g^{(k+1)} \) and a copy \( \Psi_g \) of \( \psi_g^{(k+1)} \);
3. For \( i = 1 \) to \( q \),
   - if \( d^2_{ig} + \psi_{ig} > b \), then
     set \( d^+_{ig} \leftarrow d_{ig} \cdot \sqrt{b/(d^2_{ig} + \psi_{ig})} \) in \( \Psi_g \), and
     set \( \psi^+_{ig} \leftarrow \psi_{ig} \cdot b/(d^2_{ig} + \psi_{ig}) \) in \( \Psi_g \);
   - For \( i = q + 1 \) to \( d \),
     if \( \psi_{ig} > b \), then
       set \( \psi^+_{ig} \leftarrow b \) in \( \Psi_g \);
   - For \( i = 1 \) to \( q \),
     if \( d^2_{ig} + \psi_{ig} < a \), then
       if \( d^2_{ig} + \psi_{ig} < 0 \), then
         set \( d^+_{ig} \leftarrow d_{ig} \cdot \sqrt{a/(d^2_{ig} + \psi_{ig})} \) in \( \Psi_g \), and
         set \( \psi^+_{ig} \leftarrow \psi_{ig} \cdot a/(d^2_{ig} + \psi_{ig}) \) in \( \Psi_g \),
       otherwise
         set \( d^+_{ig} \leftarrow \sqrt{a/2} \) in \( \Psi_g \), and
         set \( \psi^+_{ig} \leftarrow a/2 \) in \( \Psi_g \);
   - For \( i = q + 1 \) to \( d \),
     if \( \psi_{ig} < a \), then
       set \( \psi_{ig} \leftarrow a \) in \( \Psi_g \);
4. Set \( \Lambda_g^{(k+1)} \leftarrow U_g D_g V'_g \);
5. Set \( \Psi_g^{(k+1)} \leftarrow \Psi_g ; \)
9. Stop.
It is important to remark that the resulting EM algorithm is monotonic once the initial guess, say $\Sigma^0$, satisfies the constraints. Further, as shown in the case of Gaussian mixtures in Ingrassia and Rocci (2007), the maximization of the complete log-likelihood is guaranteed.

In order to select the appropriate bounds $a$, $b$, the idea is to extract information from the covariance matrix of $S$, say $\text{Cov}(S)$. Let $\lambda_{\min}$, $\lambda_{\max}$ be the smallest and the largest eigenvalue of $\text{Cov}(S)$, respectively, that is

$$
\lambda_s = \lambda_{\min}(\text{Cov}(S)) \text{ and } \lambda^* = \lambda_{\max}(\text{Cov}(S)).
$$

Consider the intervals $I_s = [\epsilon, \lambda_s]$ and $I^* = [\lambda_s, \lambda^*]$. Since we have no a priori information on $a$, $b$, we expect that the constrained algorithm, run with different values of the bounds, can give us a hint on how to choose them properly, by observing the final likelihood. Optimal values of the bounds should correspond to some agreement, over different random starts, on optimal values of the likelihood. Conversely, a simultaneous drop in the likelihood observed for a new bound, over different random starts, indicates that the new constraint is too strong for the data at hand. Motivated by these considerations, we set up the following procedure.

**Procedure 3 (Choice of the bounds $a$ and $b$)**

1. Set $\lambda_a = \lambda_{\min}(\text{Cov}(S))$ and $\lambda^* = \lambda_{\max}(\text{Cov}(S))$;
2. For a fixed integer $m$, compute the vector $a = (a_1, \ldots, a_m) \in \mathbb{R}^m$, defined as follows. Fix a small number $\epsilon > 0$ and divide the interval $[\epsilon, \lambda_s]$ into $2^m$ subintervals $[e_{h-1}, e_h]$ of size $e_h - e_{h-1} = (\lambda_s - \epsilon)/2^m$ for $h = 1, \ldots, 2^m$, with $e_0 = \epsilon$ and $e_{2^m} = \lambda_s$. Then set $a_j = e_{2j-1}$ for $j = 1, \ldots, m$.
3. Compute the vector $b = (b_1, \ldots, b_m) \in \mathbb{R}^m$, defined as follows. Divide the interval $[\lambda_s, \lambda^*]$ into $2^m$ subintervals $[e_{h-1}, e_h]$ of size $e_h - e_{h-1} = (\lambda^* - \lambda_s)/2^m$ for $h = 1, \ldots, 2^m$, with $e_0 = \lambda_s$ and $e_{2^m} = \lambda^*$. Then set $b_j = e_{2j-1}$ for $j = 1, \ldots, m$.
4. For $j = 1, 2$ run the constrained EM algorithm with lower bound $a = a_j$ and upper bound $b = b_j$; let $L_j$ be the corresponding maximum of the log-likelihood function;
5. While $j < m$ and $L_{j-1} \leq L_j$
   (i) Set $j \leftarrow j + 1$;
   (ii) Run the constrained EM algorithm with lower bound $a = a_j$ and upper bound $b = b_j$; let $L_j$ be the corresponding maximum of the log-likelihood function;
6. Set $b = b_{j-1}$ and $L^*_{j} \leftarrow L_{j-1}$;
7. Set $k = 1$ and $L^*_{k} \leftarrow L_{j-1}$;
8. While $k < m$ and $L^*_{k-1} \leq L^*_k$
   (i) Set $k \leftarrow k + 1$;
   (ii) Run the constrained EM algorithm with lower bound $a = a_k$ and upper bound $b = b_{j-1}$; let $L_k$ be the corresponding maximum of the log-likelihood function;
9. Set $\hat{\theta} = \arg\max \ L_{k-1}^*(\theta)$.

In the following, we set $m = 5$ and $\epsilon = 10^{-6}$. We remark that, in steps 2 and 3, the lengths of the subintervals increase following a geometric series.

## 5 Numerical studies

In this section we present numerical studies, based on simulated and real data sets, in order to show the performance of the constrained EM algorithm with respect to unconstrained approaches.

### 5.1 Artificial data

We consider here three mixtures of $G$ components of $d$-variate normal distributions, for different values of the parameter $\theta_0$. In the foremost, the parameter $\theta$ provided by the EM algorithm when it is started from the true classification, will be referred to as the “right maximum” of the likelihood function. To begin with, we generate a set of 100 different random initial clusterings to initialize the algorithm at each run: for a fixed number $G$ of components of the mixture, we draw a set of random starting values for the $z_{ig}$ from the multinomial distribution with values in $(1, 2, \ldots, G)$ with parameters $(p_1, p_2, \ldots, p_G) = (1/G, 1/G, \ldots, 1/G)$. Then we run a hundred times the unconstrained and the constrained AECM algorithms using the same set of initial clusterings in both cases. The needed routines have been written in R-code (R Core Team 2013), and they are available from the authors upon request.

The stopping criterion is based on the Aitken acceleration procedure (Aitken 1926), to estimate the asymptotic maximum of the log-likelihood at each iteration of the EM algorithm (in such a way, a decision can be made regarding whether or not the log-likelihood is sufficiently close to its estimated asymptotic value). The Aitken acceleration at iteration $k$ is given by

$$
a^{(k)} = \frac{L^{(k+1)} - L^{(k)}}{L^{(k)} - L^{(k-1)}},
$$

where $L^{(k+1)}$, $L^{(k)}$, and $L^{(k-1)}$ are the log-likelihood values from iterations $k + 1$, $k$, and $k - 1$, respectively. Then, the asymptotic estimate of the log-likelihood at iteration $k + 1$ is given by

$$
L^{(k+1)} = L^{(k)} + \frac{1}{1 - a^{(k)}}(L^{(k+1)} - L^{(k)}),
$$

see Böhning et al. (1994). In our analyses, the algorithms stopped when $L^{(k+1)} - L^{(k)} < \epsilon$, with $\epsilon = 0.001$. 

\[ \text{Springer} \]
Mixture 1: $G = 4$, $d = 7$, $q = 2$, $N = 100$. The sample $S$ has been generated with proportions $\pi = (0.2, 0.3, 0.35, 0.15)^t$ according to the following parameters:

$\mu_1 = (0, 0, 0, 0, 0, 0)^t$

$\Psi_1 = \text{diag}(0.2, 0.2, 0.2, 0.2, 0.2, 0.2)$

$\mu_2 = (5, 5, 5, 5, 5, 5)^t$

$\Psi_2 = \text{diag}(0.25, 0.25, 0.25, 0.25, 0.25, 0.25)$

$\mu_3 = (10, 10, 10, 10, 10, 10)^t$

$\Psi_3 = \text{diag}(0.15, 0.15, 0.15, 0.15, 0.15, 0.15)$

$\mu_4 = (15, 15, 15, 15, 15, 15)^t$

$\Psi_4 = \text{diag}(0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$

$$
\begin{pmatrix}
0.30 & 0.60 \\
0.60 & 0.27 \\
0.03 & -0.30 \\
-0.36 & 0.30 \\
0.30 & 0.06 \\
0.60 & -0.09 \\
-0.63 & 1.50
\end{pmatrix} =
\begin{pmatrix}
0.08 & 0.16 \\
0.16 & 0.40 \\
0.80 & -0.80 \\
-0.16 & 0.40 \\
0.80 & 0.56 \\
0.96 & -0.24 \\
1.60 & -0.24
\end{pmatrix}
$$

$$
\begin{pmatrix}
0.07 & 0.14 \\
0.14 & 0.00 \\
0.70 & 0.00 \\
-0.14 & 0.00 \\
0.70 & 0.00 \\
0.00 & -0.91 \\
0.70 & -0.70
\end{pmatrix} =
\begin{pmatrix}
0.04 & 0.08 \\
0.08 & 0.00 \\
0.40 & 0.00 \\
-0.08 & 0.00 \\
0.40 & 0.00 \\
0.00 & -0.52 \\
-0.40 & 0.80
\end{pmatrix}
$$

We began our analysis by running the unconstrained EM algorithm on the generated sample $S$: over 100 runs, it never attained the perfect classification. Afterwards, we considered the constrained algorithm based on Procedure 3; in particular the perfect classification was attained in 66% of cases. Figure 1 shows the boxplots of the distribution of the misclassification error, comparing results over 100 runs of the unconstrained and the constrained EM algorithm for MFA.

To illustrate Procedure 3, we will now give detailed comments on how to apply it to the sample $S$, drawn from Mixture 1. Firstly, we computed the smallest and the largest eigenvalues of the covariance matrix of $S$, say $\lambda_1 = 0.1983$ and $\lambda^* = 170.3176$, respectively. According to step 2 and 3 of Procedure 3, we got the following sequences of bounds $a$ and $b$

$$a = (0.00621, 0.01240, 0.02479, 0.04957, 0.09913, 0.19826)^t$$

$$b = (5.51, 10.83, 21.46, 42.73, 85.28, 170.32)^t.$$

Hence we run the algorithm for fixed $a = 0.00621$ (i.e. the smallest lower bound) and for increasing values of $b$, until the largest log-likelihood is attained: in the present case it happened for $b = 10.83$, see Fig. 2(left). Afterwards, we further tuned the estimation of the bounds by considering increasing values of $a$, till setting $a = 0.0124$, see Fig. 2(right). Table 2 add more details on results obtained by applying Procedure 3. At the end of the data-driven procedure to select the constraints, the bounds have been set to $a = 0.0124$ and $b = 10.83$. The corresponding maximum log-likelihood resulted $L^*(\theta) = -540.4518$, which has to be compared with $L(\theta_0) = -542.8097$, obtained after the first EM iteration, starting from the true assignment of units.

Finally, to give an idea of the computational cost of implementing Procedure 3, we remark that for each starting point, the unconstrained algorithm took about 27 iterations in average; the constrained procedure tried about 3–5 different combinations of lower and upper bound, and each run required 12 iterations in average. The constrained procedure required quite larger computational resources, in particular the CPU time was about 8.7 times larger than the time required by the unconstrained algorithm (0.52 versus 0.06 system CPU time, in seconds, on an Intel 2500K processor, 3.3 GHz, 4 GB DDR3). In case of larger datasets, this negligible difference could become relevant. Due to the nature of Procedure 3, however, it is quite easy to devise how to parallelize its computation. For example, having set a first value for the lower bound $a$, then the different runs for each value of $b$ could be implemented in parallel, providing final log-likelihood estimates for the subsequent choice of $b$. Then the same idea could be re-applied, running in parallel the algorithm with the selected value of $b$ and different values of $a$, providing log-likelihood estimates for the final choice of $a$. This would greatly improve the computational time.
Mixture 1: behaviour of Procedure 3: (A) (left) first, fix $a$ and increase $b$ till reaching a local maximum of the log-likelihood; (B) (right) then, the estimation is tuned by increasing the lower bound $a$ until a local maximum of the log-likelihood is attained.

First we run the unconstrained algorithm: over 100 runs, the perfect classification has been attained 23% of the time. Afterwards, we considered the constrained algorithm based on Procedure 3, obtaining perfect classification in all runs. Figure 3 shows and compares the boxplots of the distribution of the misclassification error over 100 runs of the unconstrained and constrained EM algorithms.

Finally, to give an idea of the computational cost of implementing the data-driven constraints, compared to the unconstrained version, let us remark that, for each starting point, the unconstrained algorithm took about 44 iterations in average while the constrained procedure tried about 4–5 different combinations of the lower and upper bounds and each run required 8 iterations in average. Also in this case, the constrained procedure required quite larger computational resources: the system CPU time was about 20 times larger than the time needed by the unconstrained algorithm.

Mixture 2: $G = 3$, $d = 6$, $q = 2$, $N = 150$ The sample has been generated with proportions $\boldsymbol{\pi} = (0.3, 0.4, 0.3)'$ according to the following parameters:

$$\begin{align*}
\boldsymbol{\mu}_1 &= (0, 0, 0, 0, 0, 0)' \\
\boldsymbol{\psi}_1 &= \text{diag}(0.1, 0.1, 0.1, 0.1, 0.1, 0.1) \\
\boldsymbol{\mu}_2 &= (5, 5, 5, 5, 5)' \\
\boldsymbol{\psi}_2 &= \text{diag}(0.4, 0.4, 0.4, 0.4, 0.4, 0.4) \\
\boldsymbol{\mu}_3 &= (10, 10, 10, 10, 10)' \\
\boldsymbol{\psi}_3 &= \text{diag}(0.2, 0.2, 0.2, 0.2, 0.2, 0.2)
\end{align*}$$

$$\begin{align*}
\boldsymbol{A}_1 &= \begin{pmatrix} 0.50 & 1.00 \\ 1.00 & 0.45 \\ 0.05 & -0.50 \\ -0.60 & 0.50 \\ 0.50 & 0.10 \\ 1.00 & -0.15 \end{pmatrix},
\boldsymbol{A}_2 &= \begin{pmatrix} 0.10 & 0.20 \\ 0.20 & 0.50 \\ 1.00 & -1.00 \\ -0.20 & 0.50 \\ 1.00 & 0.70 \\ 1.20 & -0.30 \end{pmatrix} \\
\boldsymbol{A}_3 &= \begin{pmatrix} 0.10 & 0.20 \\ 0.20 & 0.00 \\ 1.00 & 0.00 \\ -0.20 & 0.00 \\ 1.00 & 0.00 \\ 0.00 & -1.30 \end{pmatrix}.
\end{align*}$$

Figure 3 shows and compares the boxplots of the distribution of the misclassification error over 100 runs of the unconstrained and constrained EM algorithms.

Mixture 3: $G = 2$, $d = 3$, $q = 2$, $N = 200$ The third study concerns an artificial dataset proposed in Baek et al. (2010), in the framework of Mixtures of Common Factor Analyzers (MCFA), which are special cases of MFA with additional restrictions. To illustrate the accuracy of MCFA with respect to MFA, Baek et al. (2010) consider a sample generated with proportions $\boldsymbol{\pi} = (0.5, 0.5)'$ according to the

### Table 2 Mixture 1: (A) maximum log-likelihood estimates for $a = 0.00621$ and different values of $b$ (in bold the selected value); (B) maximum log-likelihood estimates for $b = 10.83$ and different values of $a$ (in bold the selected value)

| $b$   | 5.51 | 10.83 | 21.46 | 42.73 | 85.26 | 170.32 |
|-------|------|-------|-------|-------|-------|--------|
| $\mathcal{L}$ | -738.1761 | -540.5757 | -621.4624 | - | - | - |

| $a$   | 0.00621 | 0.01240 | 0.02479 | 0.04957 | 0.09913 | 0.19826 |
|-------|---------|---------|---------|---------|---------|---------|
| $\mathcal{L}$ | -540.5757 | -540.4518 | -540.5757 | - | - | - |
Table 3 Mixture 3: Simulated data. Frequency distribution of the misclassification error over 100 runs of both unconstrained and constrained EM algorithms for MFA

| Misclassification error | Unconstrained EM | Constrained EM |
|-------------------------|------------------|----------------|
| 3 (1.5 %)               | 72 %             | 72 %           |
| 4 (2.0 %)               | 19 %             | 28 %           |
| 75 (37.5 %)             | 9 %              | 0 %            |

following parameters:

\[ \mu_1 = (0, 0, 0)' \quad \mu_2 = (2, 2, 6)' \]

\[ \Sigma_1 = \begin{pmatrix} 4 & -1.8 & -1 \\ -1.8 & 2 & 0.9 \\ -1 & 0.9 & 2 \end{pmatrix} \]

\[ \Sigma_2 = \begin{pmatrix} 4 & 1.8 & .8 \\ 1.8 & 2 & 0.5 \\ 0.8 & 0.5 & 2 \end{pmatrix} \]

In Baek et al. (2010) a sample of a hundred units has been clustered: 2 misclassifications for MFA and 4 for MCFA were obtained, respectively. In our simulations, we considered a sample of \( N = 200 \). We run the constrained and the unconstrained algorithms, considering 95 random starting and 5 \( k \)-means initializations. After running the unconstrained EM we got three results: 3 misclassifications (1.5 %), 4 misclassifications (2.0 %) and 75 misclassifications (37.5 %) over the 200 units. The constrained EM improved the results, yielding to 3 or 4 misclassified units. Our findings are summarized in Table 3. The dataset was also modeled using MCFA, through the package mcfa provided at http://www.maths.uq.edu.au/~gjm/: after 20 random starts, 6 misclassifications were detected.

5.2 Real data

The Flea Beetles Data Set The flea beetles data, introduced by Lubischew (1962), refer to 74 specimens of flea beetle of the genus Chaetocnema, which contains three species: concinna, heptapotamica, or heikertingeri, so that \( G = 3 \). Measurements were collected on \( d = 6 \) variables concerning: the width (in the fore-part and from the side) and angle of the aedeagus, the width of the first and second joint of the tarsus, and the width of the head between the external edges of the eyes of each beetle.

The goal of the original study was to form a classification rule to distinguish the three species. To this aim, we consider \( q = 2 \) factors, according to the results of Andrews and McNicholas (2011). Like in the previous case of Mixture 3, we considered 95 random startings and 5 \( k \)-means initializations. The constrained algorithm provided the perfect classification in 43 % of cases, while the unconstrained algorithm provided the right classification in 5 % of cases, only when the initialization was performed through a \( k \)-means algorithm. In Fig. 4 we show the boxplots of the distribution of the misclassification error over 100 runs of both unconstrained and constrained EM algorithms.

Finally, we remark that the unconstrained algorithm required about 26 iteration for each run, on average; on the other side, the constrained algorithm carried out about 4–6 combinations of bounds and, on average, it performed about 15 iterations.
THE WINE DATA SET  The Wine data, proposed in Fornia et al. (1986), consists of 178 observations regarding $d = 27$ chemical and physical properties of three different cultivars of Italian wine: Barolo, Grignolino and Barbera. In our analysis, we first considered a reduced dataset containing $d = 13$ (scaled) variables, modeled through PGMM with $q = 2$ in McNicholas and Murphy (2008). Then, we applied the constrained and unconstrained EM for MFA with $q = 2$ and $G = 3$, from 100 random initializations. In particular, the constrained algorithm yielded just one misclassification. Figure 5 shows the boxplots of the distribution of the misclassification error over 100 runs of both unconstrained and constrained EM algorithms, with 95 random startings and 5 $k$-means initializations.

Afterwards, we considered the complete dataset, with $d = 27$. Parsimonious Gaussian mixture models, with $q = 4$ and $G = 3$ estimated through the pgmm package (McNicholas et al. 2011), led to two misclassifications. When initialized through $k$-means, both the unconstrained and the constrained algorithm led to two misclassifications. Using random initialization, the misclassification was larger. For the sake of completeness, we ran also mcfa (with 100 random starts) and obtained two misclassifications.

6 Concluding remarks

Mixtures of factor analyzers are commonly used to explain data, in particular, correlation between variables in multivariate observations, allowing also for dimensionality reduction. For these models, as well as for Gaussian mixtures, however, the MLE presents some drawbacks. Indeed, when no more than $q$ observations are unequivocally assigned to the $g$-th component of the mixture, in terms of the fitted posterior probabilities of component membership, this will lead to spikes or near singularities in the likelihood.

It is known, from the literature, that a constrained formulation of the EM algorithm considerably reduces such issues for Gaussian mixtures. Motivated by these considerations, in this paper we propose to maximize the likelihood function in a constrained parameter space, having no singularities and a reduced number of spurious local maxima. To avoid huge imbalances between the estimated component covariances, which are a tell-tale for spurious maximizers, we imposed that the eigenvalues of the covariance matrices lie into suitable ranges.

Since we have no a priori information on such bounds, we devise a procedure that extracts information from the data at hand, and performs a tuning of the upper and lower bounds, by observing the final likelihood over a small set of runs. The performance of the newly introduced estimation approach has been shown and compared to the usual non-constrained one, as well as to some more parsimonious approaches in the literature, based on common Gaussian factors or on patterned covariance matrices for Gaussian factors. To this purpose, we present numerical simulations on synthetic samples and applications to real data sets.

Our findings show that the problematic convergence of the EM, even more critical when dealing with factor analyzers, can be greatly improved. Remarkably, comparing the results obtained through the unconstrained EM and constrained EM for MFA, the data-driven procedure converges to a model with better fitting to the data and, at the same time, it improves the classification of the units. In some sense, the constraints actuate as a driving method to conduct the estimation process toward a better result.

In comparison with other methods of stronger constrained estimation, the milder assumption here proposed provides a better model both in terms of goodness of fit and in terms of classification. Finally, we remark that the constrained procedure requires larger computational resources; however, it can be easily parallelized and this gives ideas for future research.

Acknowledgements  The authors sincerely thank the Associate Editor and the referees for very helpful comments and valuable suggestions. Their pertinent comments helped us also in fixing some not-so-minor details in the exposition, which greatly improved the quality of the final version.

References

Aitken, A.: On Bernoulli’s numerical solution of algebraic equations. In: Proceedings of the Royal Society of Edinburgh, vol. 46, pp. 289–305 (1926)
Andrews, J., McNicholas, P.: Extending mixtures of multivariate t-factor analyzers. Stat. Comput. 21(3), 361–373 (2011)

Baek, J., McLachlan, G., Flack, L.: Mixtures of factor analyzers with common factor loadings: applications to the clustering and visualization of high-dimensional data. IEEE Trans. Pattern Anal. Mach. Intell. 32(7), 1298–1309 (2010)

Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics 49(3), 803–821 (1993)

Bishop, C.M., Tippin, M.E.: A hierarchical latent variable model for data visualization. IEEE Trans. Pattern Anal. Mach. Intell. 20(3), 281–293 (1998)

Böhnning, D., Dietz, E., Schaub, R., Schlattmann, P., Lindsay, B.: The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. Ann. Inst. Stat. Math. 46(2), 373–388 (1994)

Day, N.E.: Estimating the components of a mixture of normal distributions. Biometrika 56(3), 463–474 (1969)

Forina, M., Armanino, C., Castino, M., Ubigli, M.: Multivariate data analysis as a discriminating method of the origin of wines. Vitis 25, 189–201 (1986)

Ghahramani, Z., Hilton, G.: The EM algorithm for mixture of factor analyzers. Technical report CRG-TR-96-1 (1997)

Greselin, F., Ingrassia, S.: Constrained monotone EM algorithms for mixtures of multivariate t distributions. Stat. Comput. 20(1), 9–22 (2010)

Hathaway, R.: A constrained formulation of maximum-likelihood estimation for normal mixture distributions. Ann. Stat. 13(2), 795–800 (1985)

Hennig, C.: Breakdown points for maximum likelihood estimators of location-scale mixtures. Ann. Stat. 32(4), 1313–1340 (2004)

Hoff, P.: Subset clustering of binary sequences, with an application to genomic abnormality data. Biometrics 61, 1027–1036 (2005)

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

Ingrassia, S., Rocci, R.: Constrained monotone EM algorithms for finite mixture of multivariate Gaussians. Comput. Stat. Data Anal. 51, 5339–5351 (2007)

Liu, J., Zhang, J., Palumbo, M., Lawrence, C.: Bayesian clustering with variable and transformation selection (with discussion). Bayesian Stat. 7, 249–275 (2003)

Lubischew, A.: On the use of discriminant functions in taxonomy. Biometrics 18, 455–477 (1962)

McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000a)

McLachlan, G.J., Peel, D.: Mixtures of factor analyzers. In: Langley, P. (ed.) Proceedings of the Seventeenth International Conference on Machine Learning, pp. 599–606. Morgan Kaufmann, San Francisco (2000b)

McNicholas, P., Murphy, T.: Parsimonious Gaussian mixture models. Stat. Comput. 18(3), 285–296 (2008)

McNicholas, P.D., Jampani, K.R., McDaid, A.F., Murphy, T.B., Banks, L.: pgmm: parsimonious Gaussian mixture models. R package version 1.0 (2011)

Meng, X., van Dyk, D.: The EM algorithm—an old folk-song sung to a fast new tune. J. R. Stat. Soc., Ser. B, Stat. Methodol. 59(3), 511–567 (1997)

Pan, W., Shen, X.: Penalized model-based clustering with application to variable selection. J. Mach. Learn. Res. 8, 1145–1164 (2007)

R Core Team: R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2013). http://www.R-project.org

Raftery, A., Dean, N.: Variable selection for model-based clustering. J. Am. Stat. Assoc. 101(473), 168–178 (2006)