




Automated learning of mixtures of factor analysis models with missing information

Wan-Lun Wang¹ · Tsung-I Lin^{2,3} 

Received: 23 May 2019 / Accepted: 10 January 2020 / Published online: 29 January 2020

© Sociedad de Estadística e Investigación Operativa 2020

Abstract

The mixture of factor analyzers (MFA) model has emerged as a useful tool to perform dimensionality reduction and model-based clustering for heterogeneous data. In seeking the most appropriate number of factors (q) of a MFA model with the number of components (g) fixed a priori, a two-stage procedure is commonly implemented by firstly carrying out parameter estimation over a set of prespecified numbers of factors, and then selecting the best q according to certain penalized likelihood criteria. When the dimensionality of data grows higher, such a procedure can be computationally prohibitive. To overcome this obstacle, we develop an automated learning scheme, called the automated MFA (AMFA) algorithm, to effectively merge parameter estimation and selection of q into a one-stage algorithm. The proposed AMFA procedure that allows for much lower computational cost is also extended to accommodate missing values. Moreover, we explicitly derive the score vector and the empirical information matrix for calculating standard errors associated with the estimated parameters. The potential and applicability of the proposed method are demonstrated through a number of real datasets with genuine and synthetic missing values.

Keywords Automated learning · Factor analysis · Maximum likelihood estimation · Missing values · Model selection · One-stage algorithm

Mathematics Subject Classification 62H12 · 62H25 · 62H30

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11749-020-00702-6>) contains supplementary material, which is available to authorized users.

✉ Tsung-I Lin
tilin@nchu.edu.tw

¹ Department of Statistics, Graduate Institute of Statistics and Actuarial Science, Feng Chia University, Taichung 40724, Taiwan

² Institute of Statistics, National Chung Hsing University, Taichung 402, Taiwan

³ Department of Public Health, China Medical University, Taichung 404, Taiwan

1 Introduction

The mixture of factor analyzers (MFA) model, initially introduced by Ghahramani and Hinton (1997), has attracted considerable attention over the past three decades and been broadly applied in diverse fields, see the monograph by McLachlan and Peel (2000) for a comprehensive overview. The MFA combines the advantages of Gaussian mixture model and factor analysis (FA) and has now been taken as a promising tool for simultaneously performing model-based clustering and linear dimensionality reduction. More precisely, it provides a global nonlinear approach to dimension reduction via the adoption a factor-analytic representation for the covariance matrices of component distributions. Ghahramani and Beal (2000) presented a novel variational inference for a Bayesian treatment of MFA models. Ueda et al. (2000) proposed a split-and-merge expectation maximization (SMEM) algorithm for the MFA model and showed its real-world applications to image compression and handwritten digits recognition. In clustering microarray gene-expression profiles, McLachlan et al. (2002, 2003) illustrated the effectiveness of the MFA approach for reducing the dimension of the feature space.

A computational feasible EM algorithm (Dempster et al. 1977) has been suggested by Ghahramani and Hinton (1997) for fitting the MFA model. McLachlan et al. (2003) developed an alternating expectation conditional maximization (AECM) algorithm (Meng and van Dyk 1997) for fitting MFA and further investigated its practical use for modeling high-dimensional data. The convergence of AECM can be moderately faster than EM due to less amount of missing data in some CM-steps. Zhao and Yu (2008) further provided a much more efficient procedure, which is developed under an expectation conditional maximization (ECM; Meng and Rubin 1993) scheme by treating only membership indicators as missing data. Its appealing efficiency can be attributable to the fact that the latent factors are not taken into account in the complete-data space, while all estimators in CM-steps still have closed forms.

The occurrence of missing data that may complicate data analysis is a ubiquitous problem in nearly all fields of scientific research. There exist many strategies for dealing with incomplete data under various missing-data mechanisms. Little and Rubin (2002) outlined a taxonomy of techniques for handling missing values. The maximum likelihood (ML) methods of imputing missing values under mixtures of multivariate normal distributions have been well studied, see, for example, Ghahramani and Jordan (1994) and Lin et al. (2006). For learning FA models with possibly missing values, Zhao and Shi (2014) proposed a novel automated factor analysis (AFA) algorithm that allows for determining the number of factors (q) in an automated manner.

In this paper, we establish a generalization of AFA algorithm, called the automated MFA (AMFA) algorithm, which performs parameter estimation and determination of the number of factors (q) simultaneously for fitting the MFA with known mixture component size (g). When g is treated as unknown, the AMFA algorithm can also be applied to automatically determine an appropriate number of q for each value of g within a given range, still being much more efficient than the two-stage methods. The computational cost of AMFA can be substantially faster than the two-stage EM-based algorithms when the dimension of variables (p) or the proportion of missing information becomes high. Moreover, the proposed learning procedure allows for

handling the data in the presence of missing values under the assumption of missing at random (MAR) mechanism. Notably, the two-stage ECM algorithm can be treated as a simplified case of AMFA without updating q . To facilitate implementation, two auxiliary permutation matrices that exactly extract the observed and missing portions of an individual are incorporated into the estimating procedure. The Hessian matrix of the MFA model with incomplete data is also explicitly derived for obtaining the asymptotic standard errors of parameter estimators.

The rest of the paper is organized as follows. Section 2 formulates an incomplete-data specification of MFA model and presents some of its essential properties. In Sect. 3, we briefly describe how to perform the EM and AECM algorithms for fitting MFA models under a MAR mechanism. In Sect. 4, we propose a one-stage AMFA algorithm for parameter estimation and automatic determination of q . In Sect. 5, we explicitly derive the Hessian matrix of the observed log-likelihood function for computing the asymptotic standard errors of the ML estimators. We illustrate the usefulness and ability of the proposed method in Sect. 6 through two real-data examples and two simulation studies. Section 7 offers some concluding remarks and highlights possible directions for further work. The detailed derivations of lengthy technical results are sketched in Online Supplementary Appendices.

2 MFA model with missing data

The MFA model is a global nonlinear approach by postulating a finite mixture of g FA models for the representation of the data in a lower-dimensional subspace. Let Y_j denote a p -dimensional random vector of the j th individual for $j = 1, \dots, n$. In the MFA formulation, each observation Y_j is modeled as

$$Y_j = \mu_i + A_i F_{ij} + \varepsilon_{ij} \quad \text{with probability } \pi_i \quad (i = 1, \dots, g), \quad (1)$$

where π_i 's are known as mixing proportions which are constrained to be positive and sum to one, g is the number of mixture components, μ_i is a $p \times 1$ mean vector, A_i is a $p \times q$ matrix of factor loadings, $F_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}_q(\mathbf{0}, I_q)$ is a q -dimensional vector ($q < p$) of factors and $\varepsilon_{ij} \stackrel{\text{ind}}{\sim} \mathcal{N}_p(\mathbf{0}, \Psi_i)$ is a p -dimensional vector of errors and independent of F_{ij} . Besides, I_q is an identity matrix of size q , and Ψ_i is a $p \times p$ diagonal matrix with positive diagonal elements referred to as *uniqueness variances*.

Let $\Theta = \{\pi_i, \theta_i\}_{i=1}^g$ be the entire unknown parameters subject to $\sum_{i=1}^g \pi_i = 1$, where $\theta_i = \{\mu_i, A_i, \Psi_i\}$ denotes the vector of the unknown parameters within the i th component. Thus, the MFA model defined as (1) has the probability density function (pdf) of Y_j :

$$f(y_j; \Theta) = \sum_{i=1}^g \pi_i \phi_p(y_j; \mu_i, \Sigma_i),$$

where $\phi_p(\cdot; \mu, \Sigma)$ indicates the pdf of $\mathcal{N}_p(\mu, \Sigma)$ and Σ_i is the i th component covariance matrix taking the form of $\Sigma_i = A_i A_i^\top + \Psi_i$.

To handle the data with possibly missing values, we introduce two indicator matrices for identifying the observed and missing locations of each individual, denoted by $\mathbf{O}_j(p_j^o \times p)$ and $\mathbf{M}_j((p - p_j^o) \times p)$, such that

$$\mathbf{Y}_j^o = \mathbf{O}_j \mathbf{Y}_j \text{ and } \mathbf{Y}_j^m = \mathbf{M}_j \mathbf{Y}_j, \tag{2}$$

where $\mathbf{Y}_j^o(p_j^o \times 1)$ and $\mathbf{Y}_j^m((p - p_j^o) \times 1)$ are the observed and missing components of \mathbf{Y}_j , respectively. To identify the original group of individual \mathbf{Y}_j , an unobservable allocation vector $\mathbf{Z}_j = \{Z_{1j}, \dots, Z_{gj}\}$ is introduced, for $j = 1, \dots, n$, where $Z_{ij} \in \{0, 1\}$ are binary outcomes with constraint $\sum_{i=1}^g Z_{ij} = 1$. The role of \mathbf{Z}_j is to encode which component has brought into \mathbf{Y}_j . That is, $Z_{ij} = 1$ if \mathbf{Y}_j belongs to the i th group, and $Z_{ij} = 0$ otherwise. It follows that $\mathbf{Z}_j \sim \mathcal{M}(1, \boldsymbol{\pi})$ has a multinomial distribution with prior probabilities $\boldsymbol{\pi} = (\pi_1, \dots, \pi_g)$.

Incorporating missing information (2) into (1) leads to $\mathbf{Y}_j^o \mid (Z_{ij} = 1) \sim \mathcal{N}_{p_j^o}(\boldsymbol{\mu}_{ij}^o, \boldsymbol{\Sigma}_{ij}^{oo})$, where $\boldsymbol{\mu}_{ij}^o = \mathbf{O}_j \boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_{ij}^{oo} = \mathbf{O}_j \boldsymbol{\Sigma}_i \mathbf{O}_j^\top$. Therefore, the marginal pdf of \mathbf{Y}_j^o is given by

$$f(\mathbf{y}_j^o; \boldsymbol{\Theta}) = \sum_{i=1}^g \pi_i \phi_{p_j^o}(\mathbf{y}_j^o; \boldsymbol{\mu}_{ij}^o, \boldsymbol{\Sigma}_{ij}^{oo}). \tag{3}$$

Furthermore, we obtain a hierarchical representation of \mathbf{Y}_j^o under the MFA framework:

$$\begin{aligned} \mathbf{Y}_j^o \mid (\mathbf{F}_{ij}, Z_{ij} = 1) &\sim \mathcal{N}_{p_j^o}(\mathbf{O}_j(\boldsymbol{\mu}_i + \mathbf{A}_i \mathbf{F}_{ij}), \mathbf{O}_j \boldsymbol{\Psi}_i \mathbf{O}_j^\top), \\ \mathbf{F}_{ij} \mid (Z_{ij} = 1) &\sim \mathcal{N}_q(\mathbf{0}, \mathbf{I}_q), \quad \mathbf{Z}_j \sim \mathcal{M}(1, \boldsymbol{\pi}). \end{aligned} \tag{4}$$

As a consequence, we can establish the following theorem, which is useful for the evaluation of some conditional expectations involved in the EM algorithm discussed in Sect. 3.

Theorem 1 *Given the hierarchical representation of the MFA model specified by (4), we obtain the following conditional distributions:*

(a) *The conditional distribution of \mathbf{Y}_j^m given \mathbf{y}_j^o , \mathbf{F}_{ij} and $Z_{ij} = 1$ is*

$$\mathbf{Y}_j^m \mid (\mathbf{y}_j^o, \mathbf{F}_{ij}, Z_{ij} = 1) \sim \mathcal{N}_{p-p_j^o}(\boldsymbol{\mu}_{ij}^{m \cdot o}, \boldsymbol{\Sigma}_{ij}^{mm \cdot o}),$$

where $\boldsymbol{\mu}_{ij}^{m \cdot o} = \mathbf{M}_j \{ \boldsymbol{\mu}_i + \mathbf{A}_i \mathbf{F}_{ij} + \boldsymbol{\Psi}_i \mathbf{C}_{ij}^{oo} (\mathbf{y}_j - \boldsymbol{\mu}_i - \mathbf{A}_i \mathbf{F}_{ij}) \}$ and $\boldsymbol{\Sigma}_{ij}^{mm \cdot o} = \mathbf{M}_j (\mathbf{I}_p - \boldsymbol{\Psi}_i \mathbf{C}_{ij}^{oo}) \boldsymbol{\Psi}_i \mathbf{M}_j^\top$ with $\mathbf{C}_{ij}^{oo} = \mathbf{O}_j^\top (\mathbf{O}_j \boldsymbol{\Psi}_i \mathbf{O}_j^\top)^{-1} \mathbf{O}_j$.

(b) *The conditional distribution of \mathbf{F}_{ij} given \mathbf{y}_j^o and $Z_{ij} = 1$ is*

$$\mathbf{F}_{ij} \mid (\mathbf{y}_j^o, Z_{ij} = 1) \sim \mathcal{N}_q(\mathbf{A}_i^\top \mathbf{S}_{ij}^{oo} (\mathbf{y}_j - \boldsymbol{\mu}_i), \mathbf{I}_q - \mathbf{A}_i^\top \mathbf{S}_{ij}^{oo} \mathbf{A}_i),$$

where $\mathbf{S}_{ij}^{oo} = \mathbf{O}_j^\top (\mathbf{O}_j \boldsymbol{\Sigma}_i \mathbf{O}_j^\top)^{-1} \mathbf{O}_j$.

(c) The conditional distribution of \mathbf{Y}_j^m given \mathbf{y}_j^o and $Z_{ij} = 1$ is

$$\mathbf{Y}_j^m \mid (\mathbf{y}_j^o, Z_{ij} = 1) \sim \mathcal{N}_{p-p_j^o}(\boldsymbol{\mu}_{2.1}, \boldsymbol{\Sigma}_{22.1}),$$

where $\boldsymbol{\mu}_{2.1} = \mathbf{M}_j \{ \boldsymbol{\mu}_i + \boldsymbol{\Sigma}_i \mathbf{S}_{ij}^{oo} (\mathbf{y}_j - \boldsymbol{\mu}_i) \}$ and $\boldsymbol{\Sigma}_{22.1} = \mathbf{M}_j \{ \mathbf{I}_p - \boldsymbol{\Sigma}_i \mathbf{S}_{ij}^{oo} \} \boldsymbol{\Sigma}_i \mathbf{M}_j^\top$.

(d) The conditional distribution of \mathbf{F}_{ij} given \mathbf{y}_j and $Z_{ij} = 1$ is

$$\mathbf{F}_{ij} \mid (\mathbf{y}_j, Z_{ij} = 1) \sim \mathcal{N}_q(\mathbf{A}_i^\top \boldsymbol{\Sigma}_i^{-1} (\mathbf{y}_j - \boldsymbol{\mu}_i), \mathbf{I}_q - \mathbf{A}_i^\top \boldsymbol{\Sigma}_i^{-1} \mathbf{A}_i).$$

Proof See Supplementary Appendix A. □

We further establish the following corollary, which is useful for evaluating the Q -function in the EM algorithm discussed in Sect. 3.1.

Corollary 1 Given the conditional distributions in Theorem 1, we have

- (a) $E(\mathbf{Y}_j \mid \mathbf{y}_j^o, Z_{ij} = 1) = \boldsymbol{\mu}_i + \boldsymbol{\Sigma}_i \mathbf{S}_{ij}^{oo} (\mathbf{y}_j - \boldsymbol{\mu}_i),$
- (b) $E \left\{ \mathbf{M}_j^\top \text{cov}(\mathbf{Y}_j^m \mid \mathbf{y}_j^o, \mathbf{F}_{ij}, Z_{ij} = 1) \mathbf{M}_j \mid \mathbf{y}_j^o, Z_{ij} = 1 \right\} = (\mathbf{I}_p - \boldsymbol{\Psi}_i \mathbf{C}_{ij}^{oo}) \boldsymbol{\Psi}_i,$
- (c) $\text{cov} \left\{ \mathbf{M}_j^\top E(\mathbf{Y}_j^m \mid \mathbf{y}_j^o, \mathbf{F}_{ij}, Z_{ij} = 1) - \mathbf{A}_i \mathbf{F}_{ij} \mid \mathbf{y}_j^o, Z_{ij} = 1 \right\}$
 $= (\boldsymbol{\Omega}_{ij} - \mathbf{A}_i) \boldsymbol{\Phi}_{ij} (\boldsymbol{\Omega}_{ij} - \mathbf{A}_i)^\top,$

where $\boldsymbol{\Phi}_{ij} = \text{cov}(\mathbf{F}_{ij} \mid \mathbf{y}_j^o, Z_{ij} = 1) = \mathbf{I}_q - \mathbf{A}_i^\top \mathbf{S}_{ij}^{oo} \mathbf{A}_i$ and $\boldsymbol{\Omega}_{ij} = (\mathbf{I}_p - \boldsymbol{\Psi}_i \mathbf{C}_{ij}^{oo}) \mathbf{A}_i$.

Proof See Supplementary Appendix B. □

Furthermore, we have the following conditional expectations which are summarized in Corollary 2 and required for the development of the AECM algorithm described in Sect. 3.2.

Corollary 2 Given the hierarchical specification of (4), we can get

- (a) $\text{cov}(\mathbf{Y}_j \mid \mathbf{y}_j^o, Z_{ij} = 1) = (\mathbf{I}_p - \boldsymbol{\Sigma}_i \mathbf{S}_{ij}^{oo}) \boldsymbol{\Sigma}_i,$
- (b) $E\{ \mathbf{A}_i \mathbf{F}_{ij} (\mathbf{Y}_j - \boldsymbol{\mu}_i)^\top \mid \mathbf{y}_j^o, Z_{ij} = 1 \} = \mathbf{A}_i \boldsymbol{\Gamma}_i^\top \mathbf{V}_{ij},$
- (c) $E(\mathbf{F}_{ij} \mathbf{F}_{ij}^\top \mid \mathbf{y}_j^o, Z_{ij} = 1) = \boldsymbol{\Gamma}_i^\top \mathbf{V}_{ij} \boldsymbol{\Gamma}_i + \mathbf{I}_q - \boldsymbol{\Gamma}_i^\top \mathbf{A}_i,$

where $\boldsymbol{\Gamma}_i = \boldsymbol{\Sigma}_i^{-1} \mathbf{A}_i$ and

$$\begin{aligned} \mathbf{V}_{ij} &= E\{ (\mathbf{Y}_j - \boldsymbol{\mu}_i)(\mathbf{Y}_j - \boldsymbol{\mu}_i)^\top \mid \mathbf{y}_j^o, Z_{ij} = 1 \} \\ &= \boldsymbol{\Sigma}_i \mathbf{S}_{ij}^{oo} (\mathbf{y}_j - \boldsymbol{\mu}_i)(\mathbf{y}_j - \boldsymbol{\mu}_i)^\top \mathbf{S}_{ij}^{oo} \boldsymbol{\Sigma}_i + (\mathbf{I}_p - \boldsymbol{\Sigma}_i \mathbf{S}_{ij}^{oo}) \boldsymbol{\Sigma}_i. \end{aligned}$$

Proof See Supplementary Appendix C. □

3 ML estimation via the EM and AECM algorithms

3.1 The EM algorithm

The EM algorithm (Dempster et al. 1977) is a popular tool for carrying out ML estimation in a variety of incomplete-data problems. Each iteration of the EM algorithm is composed of two processes, alternating between an E-step in which the missing data are estimated by their conditional expectations, and an M-step which simultaneously maximizes the conditional expectation of complete-data log-likelihood function computed in the E-step with respect to all unknown parameters. The EM procedure is particularly useful when the M-step is computationally simpler than the maximization of the original likelihood.

We offer a feasible EM procedure for learning the MFA model with incomplete data. For notational convenience, we denote the allocation variables by $\mathbf{Z} = (\mathbf{Z}_1, \dots, \mathbf{Z}_n)$, latent factors by $\mathbf{F} = \{\mathbf{F}_{i1}, \dots, \mathbf{F}_{in}\}_{i=1}^g$ and missing portion of the data by $\mathbf{y}^m = (\mathbf{y}_1^m, \dots, \mathbf{y}_n^m)$. Let $\mathbf{y}^o = (\mathbf{y}_1^o, \dots, \mathbf{y}_n^o)$ be the observed portion of the data and $\mathbf{Y}_c^{[1]} = (\mathbf{y}, \mathbf{F}, \mathbf{Z})$ be the complete data, where $\mathbf{y} = (\mathbf{y}^o, \mathbf{y}^m)$.

The log-likelihood function of Θ for complete data $\mathbf{Y}_c^{[1]} = (\mathbf{y}, \mathbf{F}, \mathbf{Z})$, omitting additive constant terms, is

$$\ell_c^{[1]}(\Theta | \mathbf{Y}_c^{[1]}) = \sum_{i=1}^g \sum_{j=1}^n Z_{ij} \left\{ \ln \pi_i - \frac{1}{2} \ln |\Psi_i| - \frac{1}{2} (\mathbf{y}_j - \boldsymbol{\mu}_i - \mathbf{A}_i \mathbf{F}_{ij})^\top \Psi_i^{-1} (\mathbf{y}_j - \boldsymbol{\mu}_i - \mathbf{A}_i \mathbf{F}_{ij}) \right\}.$$

Let $\hat{\Theta}^{(k)} = \{\hat{\pi}_i^{(k)}, \hat{\boldsymbol{\mu}}_i^{(k)}, \hat{\mathbf{A}}_i^{(k)}, \hat{\Psi}_i^{(k)}\}_{i=1}^g$ denote the estimates of Θ at the k th iteration. On the E-step, we compute the expected log-likelihood for the complete data (the so-called Q -function), where the expectation is taken with respect to the conditional distributions of the missing data $(\mathbf{y}^m, \mathbf{F}, \mathbf{Z})$ given the observed data \mathbf{y}^o and the current estimates of parameters $\hat{\Theta}^{(k)}$. On the M-step, we maximize the Q -function to compute the updated parameter estimates, say $\hat{\Theta}^{(k+1)}$. The detailed implementation of the proposed EM algorithm is summarized as follows:

E-step: Compute the following conditional expectations:

$$\begin{aligned} \hat{z}_{ij}^{(k)} &= E(Z_{ij} | \mathbf{y}_j^o, \hat{\Theta}^{(k)}) = \frac{\hat{\pi}_i^{(k)} \phi_{p_j^o}(\mathbf{y}_j^o; \hat{\boldsymbol{\mu}}_{ij}^{(k)}, \hat{\boldsymbol{\Sigma}}_{ij}^{oo(k)})}{\sum_{h=1}^g \hat{\pi}_h^{(k)} \phi_{p_j^o}(\mathbf{y}_j^o; \hat{\boldsymbol{\mu}}_{hj}^{(k)}, \hat{\boldsymbol{\Sigma}}_{hj}^{oo(k)})}, \\ \hat{\mathbf{F}}_{ij}^{(k)} &= E(\mathbf{F}_{ij} | \mathbf{y}_j^o, Z_{ij} = 1, \hat{\Theta}^{(k)}) = \hat{\mathbf{A}}_i^{(k)\top} \hat{\boldsymbol{\Sigma}}_{ij}^{oo(k)} (\mathbf{y}_j - \hat{\boldsymbol{\mu}}_i^{(k)}), \\ \hat{\mathbf{y}}_{ij}^{(k)} &= E(\mathbf{Y}_j | \mathbf{y}_j^o, Z_{ij} = 1, \hat{\Theta}^{(k)}) = \hat{\boldsymbol{\mu}}_i^{(k)} + \hat{\boldsymbol{\Sigma}}_i^{(k)} \hat{\boldsymbol{\Sigma}}_{ij}^{oo(k)} (\mathbf{y}_j - \hat{\boldsymbol{\mu}}_i^{(k)}), \\ \hat{\boldsymbol{\Phi}}_{ij}^{(k)} &= \text{cov}(\mathbf{F}_{ij} | \mathbf{y}_j^o, Z_{ij} = 1, \hat{\Theta}^{(k)}) = \mathbf{I}_q - \hat{\mathbf{A}}_i^{(k)\top} \hat{\boldsymbol{\Sigma}}_{ij}^{oo(k)} \hat{\mathbf{A}}_i^{(k)}, \end{aligned} \tag{5}$$

and $\hat{\mathbf{A}}_{ij}^{(k)} = E(\mathbf{F}_{ij} \mathbf{F}_{ij}^\top \mid \mathbf{y}_j^o, Z_{ij} = 1, \hat{\boldsymbol{\theta}}^{(k)}) = \hat{\mathbf{F}}_{ij}^{(k)} \hat{\mathbf{F}}_{ij}^{(k)\top} + \hat{\boldsymbol{\phi}}_{ij}^{(k)}$, where $\hat{\boldsymbol{\mu}}_{ij}^{o(k)} = \mathbf{O}_j \hat{\boldsymbol{\mu}}_i^{(k)}$, $\hat{\boldsymbol{\Sigma}}_{ij}^{oo(k)} = \mathbf{O}_j \hat{\boldsymbol{\Sigma}}_i^{(k)} \mathbf{O}_j^\top$ and $\hat{\boldsymbol{\Sigma}}_{ij}^{oo(k)} = \mathbf{O}_j^\top (\mathbf{O}_j \hat{\boldsymbol{\Sigma}}_i^{(k)} \mathbf{O}_j^\top)^{-1} \mathbf{O}_j$.

Therefore, the resulting Q -function is obtained as

$$Q(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}^{(k)}) = \sum_{i=1}^g \sum_{j=1}^n \hat{z}_{ij}^{(k)} \left\{ \ln \pi_i - \frac{1}{2} \ln |\boldsymbol{\Psi}_i| - \frac{1}{2} \text{tr} \left[\boldsymbol{\Psi}_i^{-1} \right. \right. \\ \times \left\{ (\hat{\mathbf{y}}_{ij}^{(k)} - \boldsymbol{\mu}_i - \mathbf{A}_i \hat{\mathbf{F}}_{ij}^{(k)}) (\hat{\mathbf{y}}_{ij}^{(k)} - \boldsymbol{\mu}_i - \mathbf{A}_i \hat{\mathbf{F}}_{ij}^{(k)})^\top \right. \\ \left. \left. + (\mathbf{I}_p - \hat{\boldsymbol{\Psi}}_i^{(k)} \hat{\mathbf{C}}_{ij}^{oo(k)}) \hat{\boldsymbol{\Psi}}_i^{(k)} + (\hat{\boldsymbol{\Sigma}}_{ij}^{(k)} - \mathbf{A}_i) \hat{\boldsymbol{\phi}}_{ij}^{(k)} (\hat{\boldsymbol{\Sigma}}_{ij}^{(k)} - \mathbf{A}_i)^\top \right\} \right\},$$

where $\hat{\mathbf{C}}_{ij}^{oo(k)} = \mathbf{O}_j^\top (\mathbf{O}_j \hat{\boldsymbol{\Psi}}_i^{(k)} \mathbf{O}_j^\top)^{-1} \mathbf{O}_j$ and $\hat{\boldsymbol{\Sigma}}_{ij}^{(k)} = (\mathbf{I}_p - \hat{\boldsymbol{\Psi}}_i^{(k)} \hat{\mathbf{C}}_{ij}^{oo(k)}) \hat{\mathbf{A}}_i^{(k)}$.

M-step: Find $\hat{\boldsymbol{\theta}}^{(k+1)}$ by maximizing Q -function, leading to

$$\hat{\pi}_i^{(k+1)} = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(k)}}{n}, \quad \hat{\boldsymbol{\mu}}_i^{(k+1)} = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(k)} (\hat{\mathbf{y}}_j^{(k)} - \hat{\mathbf{A}}_i^{(k)} \hat{\mathbf{F}}_{ij}^{(k)})}{\sum_{j=1}^n \hat{z}_{ij}^{(k)}}, \\ \hat{\mathbf{A}}_i^{(k+1)} = \left[\sum_{j=1}^n \hat{z}_{ij}^{(k)} \left\{ (\hat{\mathbf{y}}_j^{(k)} - \hat{\boldsymbol{\mu}}_i^{(k+1)}) \hat{\mathbf{F}}_{ij}^{(k)\top} + \hat{\boldsymbol{\Sigma}}_{ij}^{(k)} \hat{\boldsymbol{\phi}}_{ij}^{(k)} \right\} \right] \\ \times \left[\sum_{j=1}^n \hat{z}_{ij}^{(k)} (\hat{\mathbf{F}}_{ij}^{(k)} \hat{\mathbf{F}}_{ij}^{(k)\top} + \hat{\boldsymbol{\phi}}_{ij}^{(k)}) \right]^{-1}, \\ \hat{\boldsymbol{\Psi}}_i^{(k+1)} = \frac{\text{Diag} \left(\sum_{j=1}^n \hat{z}_{ij}^{(k)} \hat{\mathbf{Y}}_{ij}^{(k)} \right)}{\sum_{j=1}^n \hat{z}_{ij}^{(k)}},$$

where $\hat{\mathbf{Y}}_{ij}^{(k)} = (\hat{\mathbf{y}}_j^{(k)} - \hat{\boldsymbol{\mu}}_i^{(k+1)} - \hat{\mathbf{A}}_i^{(k+1)} \hat{\mathbf{F}}_{ij}^{(k)}) (\hat{\mathbf{y}}_j^{(k)} - \hat{\boldsymbol{\mu}}_i^{(k+1)} - \hat{\mathbf{A}}_i^{(k+1)} \hat{\mathbf{F}}_{ij}^{(k)})^\top + (\mathbf{I}_p - \hat{\boldsymbol{\Psi}}_i^{(k)} \hat{\mathbf{C}}_{ij}^{oo(k)}) \hat{\boldsymbol{\Psi}}_i^{(k)} + (\hat{\boldsymbol{\Sigma}}_{ij}^{(k)} - \hat{\mathbf{A}}_i^{(k+1)}) \hat{\boldsymbol{\phi}}_{ij}^{(k)} (\hat{\boldsymbol{\Sigma}}_{ij}^{(k)} - \hat{\mathbf{A}}_i^{(k+1)})^\top$.

3.2 The AECM algorithm

The AECM algorithm (Meng and van Dyk 1997) is a flexible extension of the ECM algorithm (Meng and Rubin 1993) in which the specification of the complete data on each CM-step is allowed to be different. With the adoption of AECM, each iteration consists of several cycles and each cycle has its own E-step and CM-steps. Indeed, two main advantages of AECM lie on its mathematical simplicity and less computational cost, as compared with the EM algorithm derived in the preceding subsection.

To employ the AECM algorithm to the fitting of MFA models with missing values, we partition the unknown parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, where $\boldsymbol{\theta}_1 = \{\pi_i, \boldsymbol{\mu}_i\}_{i=1}^g$ and

$\Theta_2 = \{A_i, \Psi_i\}_{i=1}^g$. In the first cycle, given $\Theta_2 = \hat{\Theta}_2^{(k)}$, the log-likelihood function of Θ_1 for complete data $Y_c^{[2]} = (y, Z)$, excluding constant terms, takes the form of

$$\ell_c^{[2]}(\Theta_1 | Y_c^{[2]}) = \sum_{i=1}^g \sum_{j=1}^n Z_{ij} \left\{ \ln \pi_i - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (y_j - \mu_i)^\top \Sigma_i^{-1} (y_j - \mu_i) \right\}. \tag{6}$$

The implementation of the proposed AECM algorithm, as detailed below, consists of one E-step followed by one CM-step in each of two cycles across iterations.

E-step of cycle 1: Compute the Q -function corresponding to (6) as follows:

$$Q^{[2]}(\Theta_1 | \hat{\Theta}^{(k)}) = \sum_{i=1}^g \sum_{j=1}^n \hat{z}_{ij}^{(k)} \left[\ln \pi_i - \frac{1}{2} \text{tr} \left\{ \hat{\Sigma}_i^{(k)-1} (\hat{y}_{ij}^{(k)} - \mu_i)(\hat{y}_{ij}^{(k)} - \mu_i)^\top \right\} \right], \tag{7}$$

where $\hat{\Sigma}_i^{(k)} = \hat{A}_i^{(k)} \hat{A}_i^{(k)\top} + \hat{\Psi}_i^{(k)}$ and $\hat{y}_{ij}^{(k)}$ is the same as (5) given in the E-step of EM.

CM-steps of cycle 1: Find $\hat{\Theta}_1^{(k+1)}$ by maximizing (7). The resulting estimators are

$$\hat{\pi}_i^{(k+1)} = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(k)}}{n} \quad \text{and} \quad \hat{\mu}_i^{(k+1)} = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(k)} \hat{y}_{ij}^{(k)}}{\sum_{j=1}^n \hat{z}_{ij}^{(k)}} \tag{8}$$

in which $\hat{\mu}_i^{(k+1)}$ has a simpler expression than that of EM.

In the second cycle, we take (y, F, Z) to be the complete data and estimate Θ_2 given $\Theta_1 = \hat{\Theta}_1^{(k+1)}$. Hence, the complete-data log-likelihood function of Θ_2 for $Y_c^{[1]} = (y, F, Z)$, omitting terms irrelevant to Θ_2 , is

$$\begin{aligned} \ell_c^{[1]}(\Theta_2 | Y_c^{[1]}) &= \sum_{i=1}^g \sum_{j=1}^n Z_{ij} \\ &\times \left\{ -\frac{1}{2} \ln |\Psi_i| - \frac{1}{2} (y_j - \hat{\mu}_i^{(k+1)} - A_i F_{ij})^\top \Psi_i^{-1} (y_j - \hat{\mu}_i^{(k+1)} - A_i F_{ij}) \right\}. \end{aligned} \tag{9}$$

E-step of cycle 2: Similarly, obtain the Q -function corresponding to (9), given by

$$\begin{aligned} Q^{[1]}(\Theta_2 | \hat{\Theta}_1^{(k+1)}, \hat{\Theta}_2^{(k)}) &= \sum_{i=1}^g \sum_{j=1}^n \hat{z}_{ij}^{(k)} \\ &\times \left\{ -\frac{1}{2} \ln |\Psi_i| - \frac{1}{2} \text{tr} \left(\Psi_i^{-1} \left[(I_p - A_i \hat{F}_i^{(k)\top}) \hat{V}_i^{(k)} (I_p - A_i \hat{F}_i^{(k)\top})^\top + A_i \hat{\xi}_i^{(k)} A_i^\top \right] \right) \right\}, \end{aligned} \tag{10}$$

where $\hat{\Gamma}_i^{(k)} = \hat{\Sigma}_i^{(k-1)} \hat{A}_i^{(k)}$, $\hat{\xi}_i^{(k)} = \sum_{j=1}^n \hat{z}_{ij}^{(k)} (\mathbf{I}_q - \hat{\Gamma}_i^{(k)\top} \hat{A}_i^{(k)}) / \sum_{j=1}^n \hat{z}_{ij}^{(k)}$ and

$$\hat{V}_i^{(k)} = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(k)} \left\{ (\hat{y}_{ij}^{(k)} - \hat{\mu}_i^{(k+1)}) (\hat{y}_{ij}^{(k)} - \hat{\mu}_i^{(k+1)})^\top + (\mathbf{I}_p - \hat{\Sigma}_i^{(k)} \hat{S}_{ij}^{\text{oo}(k)}) \hat{\Sigma}_i^{(k)} \right\}}{\sum_{j=1}^n \hat{z}_{ij}^{(k)}}. \tag{11}$$

CM-step of cycle 2: Maximizing (10) over A_i and Ψ_i gives

$$\begin{aligned} \hat{A}_i^{(k+1)} &= \hat{V}_i^{(k)} \hat{\Gamma}_i^{(k)} \left(\hat{\Gamma}_i^{(k)\top} \hat{V}_i^{(k)} \hat{\Gamma}_i^{(k)} + \hat{\xi}_i^{(k)} \right)^{-1} \quad \text{and} \\ \hat{\Psi}_i^{(k+1)} &= \text{Diag}(\hat{V}_i^{(k)} - \hat{A}_i^{(k+1)} \hat{\Gamma}_i^{(k)\top} \hat{V}_i^{(k)}). \end{aligned}$$

4 Automated learning of MFA with missing information

In the EM and AECM algorithms, the values of g and q are considered to be fixed and known. The most popular measure for model selection in mixture models is the Bayesian information criterion (BIC; Schwarz 1978) due to its satisfactory theoretical properties (Keribin 2000) and empirical performances (Fraley and Raftery 1998, 2002). The BIC is calculated as

$$\text{BIC} = m \ln n - 2\ell_{\max},$$

where m is the number of parameters and ℓ_{\max} is the maximized log-likelihood value. However, it is typically a time-consuming process to perform a grid search of a range of (g, q) pairs. To cope with this obstacle, we develop a faster learning procedure, called the AMFA algorithm in short, to determine the best q in MFA despite that g is still assumed to be known. When $g = 1$, our procedure includes the one-stage AFA algorithm (Zhao and Shi 2014) as a special case.

Meng and Rubin (1993) have shown that the asymptotic convergence rate of the EM-type algorithm is inversely related the amount of missing data. More exactly, the speeds of the EM-type algorithms are governed by the fractions of observed information in the respective data-augmentation space. Zhao and Yu (2008) proposed a fast ECM algorithm for the MFA model without the presence of missing values under a smaller data augmentation $\mathbf{Y}_c^{[2]}$ as utilized in the first cycle of AECM.

From (6), unlike the AECM algorithm, we utilize the smaller complete data $\mathbf{Y}_c^{[2]} = (\mathbf{y}, \mathbf{Z})$ to update $\hat{A}_i^{(k)}$ and $\hat{\Psi}_i^{(k)}$ by maximizing

$$Q^*(\Theta_2) = -\frac{1}{2} \sum_{i=1}^g \sum_{j=1}^n \hat{z}_{ij}^{(k)} \left\{ \ln |A_i A_i^\top + \Psi_i| + \text{tr}((A_i A_i^\top + \Psi_i)^{-1} \hat{V}_i^{(k)}) \right\}, \tag{12}$$

where $\Theta_2 = \{A_i, \Psi_i\}_{i=1}^g$ and $\hat{V}_i^{(k)}$ is the local covariance matrix as defined by (11). Let $\tilde{A}_i \triangleq [\hat{\Psi}_i^{(k)}]^{-1/2} A_i$ and $\tilde{V}_i \triangleq [\hat{\Psi}_i^{(k)}]^{-1/2} \hat{V}_i^{(k)} [\hat{\Psi}_i^{(k)}]^{-1/2}$. To locally maximize

(12) under smaller augmentation data space with (y, \mathbf{Z}) , it follows straightforwardly from Zhao and Yu (2008) that each $\tilde{\mathbf{A}}_i$ satisfies

$$\tilde{\mathbf{A}}_i \left(\mathbf{I}_q + \tilde{\mathbf{A}}_i^\top \tilde{\mathbf{A}}_i \right) = \tilde{\mathbf{V}}_i \tilde{\mathbf{A}}_i. \tag{13}$$

Let $\mathbf{U}_{iq_i} \mathbf{D}_i \mathbf{R}_i$ be the singular value decomposition of $\tilde{\mathbf{A}}_i$, where \mathbf{U}_{iq_i} is a $p \times q_i$ matrix satisfying $\mathbf{U}_{iq_i}^\top \mathbf{U}_{iq_i} = \mathbf{I}_{q_i}$, $\mathbf{D}_i = \text{Diag}(d_{i1}, \dots, d_{iq_i})$ is a diagonal matrix with elements $d_{i1} \geq d_{i2} \geq \dots \geq d_{iq_i} > 0$, and \mathbf{R}_i is an arbitrarily chosen $q_i \times q$ matrix satisfying $\mathbf{R}_i \mathbf{R}_i^\top = \mathbf{I}_{q_i}$. Therefore, the decomposition of (13) is equivalent to

$$\mathbf{U}_{iq_i} \left(\mathbf{I}_{q_i} + \mathbf{D}_i^2 \right) = \tilde{\mathbf{V}}_i \mathbf{U}_{iq_i},$$

where $(\mathbf{u}_{i1}, 1 + d_{i1}^2), \dots, (\mathbf{u}_{iq_i}, 1 + d_{iq_i}^2)$ are the corresponding eigenvector–eigenvalue pairs of $\tilde{\mathbf{V}}_i$.

Consider the decomposition for $\tilde{\mathbf{V}}_i = \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i^\top$, where $\mathbf{U}_i = [\mathbf{u}_{i1} \ \dots \ \mathbf{u}_{ip}]$ is a $p \times p$ orthogonal matrix and $\mathbf{\Lambda}_i = \text{Diag}(\lambda_{i1}, \dots, \lambda_{ip})$. Using the facts of

$$\ln |\mathbf{I}_p + \tilde{\mathbf{A}}_i \tilde{\mathbf{A}}_i^\top| = \sum_{r=1}^{q_i} \ln \lambda_{ir} \quad \text{and} \quad \text{tr} \left\{ (\mathbf{I}_p + \tilde{\mathbf{A}}_i \tilde{\mathbf{A}}_i^\top)^{-1} \tilde{\mathbf{V}}_i \right\} = \sum_{r=1}^{q_i} 1 + \sum_{r=q_i+1}^p \lambda_{ir},$$

it can be therefore established from (12) that

$$\begin{aligned} & Q^*(\mathbf{A}_1, \dots, \mathbf{A}_g, \{\hat{\boldsymbol{\Psi}}_i^{(k)}\}_{i=1}^g) \\ &= -\frac{1}{2} \sum_{i=1}^g \hat{n}_i^{(k)} \left\{ \ln |\hat{\boldsymbol{\Psi}}_i^{(k)}| + \sum_{r=1}^{q_i} (\ln \lambda_{ir} - \lambda_{ir} + 1) + \sum_{r=1}^p \lambda_{ir} \right\}, \end{aligned} \tag{14}$$

where $\hat{n}_i^{(k)} = \sum_{j=1}^n \hat{z}_{ij}^{(k)}$. Obviously, the sum $\sum_{i=1}^g n_i^{(k)} (\ln |\hat{\boldsymbol{\Psi}}_i^{(k)}| + \sum_{r=1}^p \lambda_{ir})$ is irrelevant to the determination of q_i , and the function $f(\lambda) = \ln \lambda - \lambda + 1$ is negative and strictly decreasing over the interval $(1, \infty)$. So, it is clear to see that the more the eigenvalue λ_r is larger than one, the more the factor r contributes (14).

Following Zhao and Shi (2014), we incorporate the penalty term of BIC, say $m(q) \ln(n)$, into (14) to aid the selection of q . Accordingly, the optimal q can be calculated as an integer solution satisfying the equation:

$$\hat{q}^{(k+1)} = \underset{q \leq q_{\max}}{\text{argmin}} \left[\sum_{i=1}^g \hat{n}_i^{(k)} \left\{ \sum_{r=1}^q (\ln \lambda_{ir} - \lambda_{ir} + 1) \right\} + m(q) \ln n \right], \tag{15}$$

where q_{\max} is the greatest integer satisfying the following identifiability constraint (Ledermann 1937):

$$q_{\max} \leq p + (1 - \sqrt{1 + 8p})/2. \tag{16}$$

Letting $q = \hat{q}^{(k+1)}$, it can be shown that the solution for \mathbf{A}_i that globally maximizes (12) is

$$\hat{\mathbf{A}}_i^{(k+1)} = [\hat{\Psi}_i^{(k)}]^{1/2} \mathbf{U}_{iq'_i} (\mathbf{A}_{iq'_i} - \mathbf{I}_{q'_i})^{1/2} \mathbf{R}_i, \tag{17}$$

where $\mathbf{A}_{iq'_i} = \text{Diag}(\lambda_{i1}, \dots, \lambda_{iq'_i})$ for which $q'_i = q$ if $\lambda_{iq'_i} > 1$; otherwise, q'_i is set to be the unique integer satisfying $\lambda_{iq'_i} > 1 \geq \lambda_{iq'_i+1}$. As such, $\mathbf{A}_{iq'_i} - \mathbf{I}_{q'_i}$ is guaranteed to be positive definite. For simplicity, the rotation matrix \mathbf{R}_i is chosen as the first q'_i rows of \mathbf{I}_q to satisfy the requirement of $\mathbf{R}_i \mathbf{R}_i^\top = \mathbf{I}_{q'_i}$ in our analysis.

Next, we devise an element-wise scheme in a similar spirit of Zhao et al. (2008) for updating the component uniqueness variances sequentially. First, we define

$$\hat{\Psi}_{ir}^{(k)} \triangleq \text{Diag}(\hat{\psi}_{i1}^{(k+1)}, \dots, \hat{\psi}_{i,r-1}^{(k+1)}, \psi_{ir}, \hat{\psi}_{i,r+1}^{(k)}, \dots, \hat{\psi}_{ip}^{(k)}).$$

Given $\mathbf{A}_i = \hat{\mathbf{A}}_i^{(k+1)}$ and $\Psi_i = \hat{\Psi}_{ir}^{(k)}$, Eq. (12) is obviously a function of ψ_{ir} , so we simply denote it by $\bar{\ell}(\psi_{ir})$. Letting $\Sigma_{ir} \triangleq \hat{\Psi}_{ir}^{(k)} + \hat{\mathbf{A}}_i^{(k+1)} \hat{\mathbf{A}}_i^{(k+1)\top}$, maximization of $\bar{\ell}(\psi_{ir})$ is equivalent to solving the following equation:

$$\frac{-2}{\hat{n}_i^{(k)}} \frac{\partial \bar{\ell}(\psi_{ir})}{\partial \psi_{ir}} = (\Sigma_{ir}^{-1} - \Sigma_{ir}^{-1} \mathbf{V}_i \Sigma_{ir}^{-1})_{rr} = 0, \tag{18}$$

where $(\cdot)_{rr}$ indicates the (r, r) th entry of the matrix given in the parenthesis. Multiplying both sides by $[\hat{\Psi}_i^{(k)}]^{-1/2}$, Eq. (18) can be equivalently rewritten as

$$(\tilde{\Sigma}_{ir}^{-1} - \tilde{\Sigma}_{ir}^{-1} \tilde{\mathbf{V}}_i \tilde{\Sigma}_{ir}^{-1})_{rr} = 0, \tag{19}$$

where $\tilde{\Sigma}_{ir} = [\hat{\Psi}_i^{(k)}]^{-1/2} \Sigma_{ir} [\hat{\Psi}_i^{(k)}]^{-1/2}$.

To calculate the inverse of $\tilde{\Sigma}_{ir}$ easily, we adopt the following notation:

$$\tilde{\Psi}_{ir} \triangleq \hat{\Psi}_{ir}^{(k)} [\hat{\Psi}_i^{(k)}]^{-1} = \omega_{ir} \mathbf{e}_r \mathbf{e}_r^\top + \sum_{h=1}^{r-1} \hat{\omega}_{ih}^{(k+1)} \mathbf{e}_h \mathbf{e}_h^\top + \mathbf{I}_p,$$

where $\omega_{ir} = \psi_{ir}/\psi_{ir}^{(k)} - 1$ and $\hat{\omega}_{ih}^{(k+1)} = \hat{\psi}_{ih}^{(k+1)}/\hat{\psi}_{ih}^{(k)} - 1$. Combining the above definition, we obtain

$$\tilde{\Sigma}_{ir} = \tilde{\Psi}_{ir} + \tilde{\mathbf{A}}_i \tilde{\mathbf{A}}_i^\top = \omega_{ir} \mathbf{e}_r \mathbf{e}_r^\top + \mathbf{B}_{ir},$$

where \mathbf{e}_r is the r th column of \mathbf{I}_p , for $r = 1, \dots, p$, and $\mathbf{B}_{ir} = \sum_{h=1}^{r-1} \hat{\omega}_{ih}^{(k+1)} \mathbf{e}_h \mathbf{e}_h^\top + \mathbf{I}_p + \tilde{\mathbf{A}}_i \tilde{\mathbf{A}}_i^\top$. Using the matrix inversion formula (Golub and Van Loan 1989), we obtain

$$\tilde{\Sigma}_{ir}^{-1} = (\mathbf{B}_{ir} + \omega_{ir} \mathbf{e}_r \mathbf{e}_r^\top)^{-1} = \mathbf{B}_{ir}^{-1} - \frac{\omega_{ir} \mathbf{B}_{ir}^{-1} \mathbf{e}_r \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1}}{1 + \omega_{ir} \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r}, \tag{20}$$

where the last equality holds when \mathbf{B}_{ir} is a nonsingular matrix and $1 + \omega_{ir} \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r \neq 0$.

Substituting (20) into (19) yields the following result:

$$(\tilde{\Sigma}_{ir}^{-1} - \tilde{\Sigma}_{ir}^{-1} \tilde{\mathbf{V}}_i \tilde{\Sigma}_{ir}^{-1})_{rr} = \frac{\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r + \omega_{ir} (\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r)^2 - \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \tilde{\mathbf{V}}_i \mathbf{B}_{ir}^{-1} \mathbf{e}_r}{(1 + \omega_{ir} \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r)^2}, \tag{21}$$

where the right-hand side is a function of ω_{ir} . Because the denominator of (21) is greater than zero, equating the equation to zero gives the unique solution

$$\hat{\omega}_{ir}^{(k+1)} = (\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r)^{-2} (\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \tilde{\mathbf{V}}_i \mathbf{B}_{ir}^{-1} \mathbf{e}_r - \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r). \tag{22}$$

The corresponding solution of ψ_{ir} is given by

$$\hat{\psi}_{ir}^{(k+1)} = [\hat{\omega}_{ir}^{(k+1)} + 1] \hat{\psi}_{ir}^{(k)}. \tag{23}$$

However, the solution (23) is not always kept positive under certain circumstances. To ensure that Ψ_i is positive definite, one common way is to choose a very small number $\eta > 0$ and assume $\psi_{ir} > \eta$. That is, we set $\hat{\psi}_{ir}^{(k+1)} = \eta$ if $\hat{\omega}_{ir}^{(k+1)} \leq -1$. In such a way, the uniqueness variances are guaranteed to be positive and $\hat{\omega}_{ih}^{(k+1)} = (\hat{\psi}_{ih}^{(k+1)} / \hat{\psi}_{ih}^{(k)} - 1) > -1$, for $h = 1, \dots, r - 1$. Thus, it is trivial to see that $\mathbf{B}_{ir} = \text{diag}(\hat{\omega}_{i1}^{(k+1)} + 1, \dots, \hat{\omega}_{i,r-1}^{(k+1)} + 1, 1, \dots, 1) + \tilde{\mathbf{A}}_i \tilde{\mathbf{A}}_i^\top$ is positive definite and so is invertible. It is worth to note that the inverse of \mathbf{B}_{ir} can be easily calculated because of $1 + \omega_{ir} \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r > 0$ as $\omega_{ir} > -1$. This fact is justified in Proposition 2 of Zhao et al. (2008) for the single-factor analysis model.

Afterward, we discuss in more detail about the unimodal property of $\bar{\ell}(\psi_{ir})$. As we have shown previously,

$$\begin{aligned} -\frac{2}{\hat{n}_i^{(k)}} \frac{\partial \bar{\ell}(\psi_{ir})}{\partial \psi_{ir}} &= (\tilde{\Sigma}_{ir}^{-1} - \tilde{\Sigma}_{ir}^{-1} \tilde{\mathbf{V}}_i \tilde{\Sigma}_{ir}^{-1})_{rr} \times [\hat{\psi}_{ir}^{(k)}]^{-1} \\ &= \frac{\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r + \omega_{ir} (\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r)^2 - \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \tilde{\mathbf{V}}_i \mathbf{B}_{ir}^{-1} \mathbf{e}_r}{(1 + \omega_{ir} \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r)^2} [\hat{\psi}_{ir}^{(k)}]^{-1}. \end{aligned}$$

From (23), we obtain $\hat{\omega}_{ir}^{(k+1)} = \hat{\psi}_{ir}^{(k+1)} / \hat{\psi}_{ir}^{(k)} - 1$ and $\omega_{ir} = \psi_{ir} / \hat{\psi}_{ir}^{(k)} - 1$. Now, if $\hat{\psi}_{ir}^{(k+1)} > \psi_{ir}$, it implies that

$$\hat{\omega}_{ir}^{(k+1)} = (\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r)^{-2} (\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \tilde{\mathbf{V}}_i \mathbf{B}_{ir}^{-1} \mathbf{e}_r - \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r) > \omega_{ir},$$

or equivalently $\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r + \omega_{ir} (\mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \mathbf{e}_r)^2 - \mathbf{e}_r^\top \mathbf{B}_{ir}^{-1} \tilde{\mathbf{V}}_i \mathbf{B}_{ir}^{-1} \mathbf{e}_r < 0$. This will lead to $\bar{\ell}'(\psi_{ir}) > 0$, since $\hat{\psi}_{ir}^{(k)}$ is always positive. Similarly, we can verify that $\bar{\ell}'(\psi_{ir}) < 0$ if $\hat{\psi}_{ir}^{(k+1)} < \psi_{ir}$. Hence, we can conclude that $\hat{\psi}_{ir}^{(k+1)}$ is a global maximizer of $\bar{\ell}(\psi_{ir})$ at iteration $k + 1$.

In summary, the proposed AMFA algorithm proceeds as follows:

E-step: Same as the E-step of cycle 1 of the AECM algorithm.

CM-step 1: Compute $\hat{\pi}_i^{(k+1)}$ and $\hat{\mu}_i^{(k+1)}$ using the same estimators (8) as in the AECM algorithm.

CM-step 2: Compute $\hat{q}^{(k+1)}$ using Eq. (15).

CM-step 3: Compute $\hat{A}_i^{(k+1)}$ using Eq. (17).

CM-step 4: Compute $\hat{\psi}_{ir}^{(k+1)}$ using (23) for $i = 1, \dots, g$ and $r = 1, \dots, p$. To avoid an improper solution of ψ_{ir} , the estimated value of ω_{ir} in (22) can be rendered as

$$\hat{\omega}_{ir}^{(k+1)} = \max \left\{ \frac{\mathbf{b}_{i,r}^\top \tilde{\mathbf{V}}_i \mathbf{b}_{i,r} - b_{i,rr}}{b_{i,rr}^2}, \frac{\eta}{\hat{\psi}_{ir}^{(k)}} - 1 \right\},$$

where $\mathbf{b}_{i,r}$ and $b_{i,rr}$ denote the r th column and the (r, r) th element of \mathbf{B}_{ir}^{-1} , respectively.

Notably, when CM-step 2 is skipped, the AMFA algorithm is referred to as the ECM procedure (Zhao and Yu 2008) for a fixed q . As will be demonstrated in our illustrations, the AMFA algorithm is much more efficient than the tandem approach implemented by EM, AECM and ECM algorithms.

5 Provision of standard error estimates

The sampling-based bootstrap technique (Efron and Tibshirani 1993) is a simple procedure for estimating the sampling distribution of estimator of interest. Although conceptually simple, one major obvious drawback of this approach is that it can be either computational intractable or unrealistically time-consuming, especially for mixture models (Basford et al. 1997). We offer a simple and effective information-based method for obtaining the standard errors of the ML estimates of the parameters after convergence of the AMFA algorithm. Following the notation used in Boldea and Magnus (2009) and Montanari and Viroli (2011), we explicitly derive the score vector and the Hessian matrix of the MFA model with possible missing values. The standard errors for the parameter estimates can be obtained by evaluating the inverse of the observed information matrix.

From (3), the log-likelihood for the MFA model with incomplete data is

$$\ell(\Theta \mid \mathbf{y}^o) = \sum_{j=1}^n \ln f(\mathbf{y}_j^o) = \sum_{j=1}^n \ln \left(\sum_{i=1}^g \pi_i \phi_{ij}^o \right), \tag{24}$$

where $\phi_{ij}^o = \phi_{pj}^o(\mathbf{y}_j^o; \boldsymbol{\mu}_{ij}^o, \boldsymbol{\Sigma}_{ij}^{o0})$.

Let $\mathbf{f}_i = \text{vec}(\mathbf{A}_i)$ denote a $pq \times 1$ column vector obtained by stacking all columns of \mathbf{A}_i , and $\boldsymbol{\psi}_i$ a $p \times 1$ containing the main diagonal of $\boldsymbol{\Psi}_i$. Thus, there exists a unique $p^2 \times p$ duplication matrix \mathbf{D} such that $\boldsymbol{\psi}_i = \mathbf{D}^\top \text{vec}(\boldsymbol{\Psi}_i)$. Moreover, it can be shown that the first and second derivatives of $\ln \phi_{ij}^o$ are, respectively, given by

$$d \ln \phi_{ij}^0 = \mathbf{b}_{ij}^\top d\boldsymbol{\mu}_i - (\text{vec}(\mathbf{A}_i))^\top (\mathbf{I}_q \otimes \mathbf{B}_{ij}) d\text{vec}(\mathbf{A}_i) - \frac{1}{2} \text{diag}(\mathbf{B}_{ij}) d\boldsymbol{\psi}_i \tag{25}$$

and

$$\begin{aligned} d^2 \ln \phi_{ij}^0 &= -d\boldsymbol{\mu}_i^\top \mathbf{S}_{ij}^{00} d\boldsymbol{\mu}_i - 4(d \text{vec}(\mathbf{A}_i))^\top (\mathbf{A}_i^\top \mathbf{b}_{ij} \otimes \mathbf{S}_{ij}^{00}) d\boldsymbol{\mu}_i \\ &\quad - 2(d\boldsymbol{\psi}_i)^\top \mathbf{D}^\top (\mathbf{b}_{ij} \otimes \mathbf{S}_{ij}^{00}) d\boldsymbol{\mu}_i - (d \text{vec}(\mathbf{A}_i))^\top (\mathbf{I}_q \otimes \mathbf{S}_{ij}^{00}) d \text{vec}(\mathbf{A}_i) \\ &\quad - 2(d \text{vec}(\mathbf{A}_i))^\top (\mathbf{A}_i^\top \boldsymbol{\Upsilon}_{ij} \mathbf{A}_i \otimes \mathbf{S}_{ij}^{00}) d \text{vec}(\mathbf{A}_i) \\ &\quad - 2(d\boldsymbol{\psi}_i)^\top \mathbf{D}^\top (\boldsymbol{\Upsilon}_{ij} \mathbf{A}_i \otimes \mathbf{S}_{ij}^{00}) d \text{vec}(\mathbf{A}_i) - \frac{1}{2} (d\boldsymbol{\psi}_i)^\top \mathbf{D}^\top (\boldsymbol{\Upsilon}_{ij} \otimes \mathbf{S}_{ij}^{00}) \mathbf{D} d\boldsymbol{\psi}_i, \end{aligned} \tag{26}$$

where $\mathbf{b}_{ij} = \mathbf{S}_{ij}^{00} (\mathbf{y}_j - \boldsymbol{\mu}_i)$, $\mathbf{B}_{ij} = \mathbf{S}_{ij}^{00} - \mathbf{b}_{ij} \mathbf{b}_{ij}^\top$ and $\boldsymbol{\Upsilon}_{ij} = \mathbf{S}_{ij}^{00} - 2\mathbf{B}_{ij}$. For the sake of concise representation, the following lemma presents the compact forms of (25) and (26) whose detailed proof is sketched in Supplementary Appendix D.

Lemma 1 *The first two derivatives of $\ln \phi_{ij}^0$ in MFA models that allow for missing values are given by*

$$d \ln \phi_{ij}^0 = \mathbf{c}_{ij}^\top d\boldsymbol{\theta}_i \text{ and } d^2 \ln \phi_{ij}^0 = -(d\boldsymbol{\theta}_i)^\top \mathbf{C}_{ij} (d\boldsymbol{\theta}_i),$$

where

$$\begin{aligned} \mathbf{c}_{ij} &= \begin{pmatrix} \mathbf{b}_{ij} \\ -(\mathbf{I}_q \otimes \mathbf{B}_{ij}) \text{vec}(\mathbf{A}_i) \\ -\frac{1}{2} \text{diag}(\mathbf{B}_{ij}) \end{pmatrix}, \text{ and} \\ \mathbf{C}_{ij} &= \begin{pmatrix} \mathbf{S}_{ij}^{00} & 2(\mathbf{b}_{ij}^\top \mathbf{A}_i \otimes \mathbf{S}_{ij}^{00}) & (\mathbf{b}_{ij}^\top \otimes \mathbf{S}_{ij}^{00}) \mathbf{D} \\ 2(\mathbf{A}_i^\top \mathbf{b}_{ij} \otimes \mathbf{S}_{ij}^{00}) & 2\mathbf{A}_i^\top \boldsymbol{\Upsilon}_{ij} \mathbf{A}_i \otimes \mathbf{S}_{ij}^{00} + (\mathbf{I}_q \otimes \mathbf{B}_{ij}) & (\mathbf{A}_i^\top \boldsymbol{\Upsilon}_{ij} \otimes \mathbf{S}_{ij}^{00}) \mathbf{D} \\ \mathbf{D}^\top (\mathbf{b}_{ij} \otimes \mathbf{S}_{ij}^{00}) & \mathbf{D}^\top (\boldsymbol{\Upsilon}_{ij} \mathbf{A}_i \otimes \mathbf{S}_{ij}^{00}) & \frac{1}{2} \mathbf{D}^\top (\boldsymbol{\Upsilon}_{ij} \otimes \mathbf{S}_{ij}^{00}) \mathbf{D} \end{pmatrix}. \end{aligned} \tag{27}$$

Proof The proof is straightforward and hence omitted. □

Following the same notation used in Boldea and Magnus (2009), we write $\mathbf{a}_i = \mathbf{e}_i / \pi_i$, for $i = 1, \dots, g - 1$, where \mathbf{e}_i denotes the i th column of \mathbf{I}_{g-1} , and $\mathbf{a}_g = -\mathbf{1}_{g-1} / \pi_g$ with $\mathbf{1}_{g-1}$ being a $(g - 1) \times 1$ vector of ones. The score vector is defined as the first derivative of (24), denoted by $\mathbf{s}(\boldsymbol{\Theta} \mid \mathbf{y}^0) = \sum_{j=1}^n \mathbf{s}_j(\boldsymbol{\Theta} \mid \mathbf{y}_j^0)$, where

$$\mathbf{s}_j(\boldsymbol{\Theta} \mid \mathbf{y}_j^0) = \frac{\partial \ln f(\mathbf{y}_j^0)}{\partial \boldsymbol{\Theta}} = \text{vec}(\mathbf{s}_j^\pi, \mathbf{s}_j^1, \dots, \mathbf{s}_j^g).$$

Using Lemma 1 and

$$d \ln f(\mathbf{y}_j^0) = \sum_{i=1}^g \frac{\pi_i \phi_{ij}^0 d \ln(\pi_i \phi_{ij}^0)}{\sum_{h=1}^g \pi_h \phi_{hj}^0}, \tag{28}$$

we obtain the explicit expressions for the elements of $s_j(\Theta | y_j^0)$, which contain $s_j^\pi = \sum_{i=1}^g \alpha_{ij}^0 a_i = \bar{a}_j$ and $s_j^i = \alpha_{ij}^0 c_{ij}$ for $i = 1, \dots, g$, where

$$\alpha_{ij}^0 = \frac{\pi_i \phi_{ij}^0}{\sum_{h=1}^g \pi_h \phi_{hj}^0} \tag{29}$$

is the posterior probability that y_j^0 belongs to the i th group.

Consequently, the second derivative of the log-likelihood function, called the Hessian matrix, is $H(\Theta | y^0) = \sum_{j=1}^n H_j(\Theta | y_j^0)$, where

$$H_j(\Theta | y_j^0) = \frac{\partial s_j(\Theta | y_j^0)}{\partial \Theta^\top} = \begin{bmatrix} H_j^{\pi\pi} & H_j^{\pi 1} & \dots & H_j^{\pi g} \\ H_j^{1\pi} & H_j^{11} & \dots & H_j^{1g} \\ \vdots & \vdots & & \vdots \\ H_j^{g\pi} & H_j^{g1} & \dots & H_j^{gg} \end{bmatrix}. \tag{30}$$

From (28), we can deduce

$$d^2 \ln f(y_j^0) = \sum_{i=1}^g \alpha_{ij}^0 [d^2 \ln(\pi_i \phi_{ij}^0) + \{d \ln(\pi_i \phi_{ij}^0)\}^2] - \left\{ \sum_{i=1}^g \alpha_{ij}^0 d \ln(\pi_i \phi_{ij}^0) \right\}^2. \tag{31}$$

Using Lemma 1 in conjunction with (31), we establish the following theorem, which allows a direct calculation of the Hessian matrix defined in (30).

Theorem 2 *The Hessian matrix in (30) is composed of the elements of*

$$\begin{aligned} H_j^{\pi\pi} &= -\bar{a}_j \bar{a}_j^\top, \quad H_j^{\pi i} = \alpha_{ij}^0 (a_i - \bar{a}_j) c_{ij}^\top, \quad H_j^{i\pi} = H_j^{\pi i \top}, \\ H_j^{ik} &= -\alpha_{ij}^0 \alpha_{kj}^0 c_{ij} c_{kj}^\top \quad (i \neq k), \quad H_j^{ki} = (H_j^{ik})^\top, \\ H_j^{ii} &= -\alpha_{ij}^0 C_{ij} + \alpha_{ij}^0 (1 - \alpha_{ij}^0) c_{ij} c_{ij}^\top, \end{aligned}$$

where c_{ij} and C_{ij} are defined as in (27), and α_{ij}^0 is given in (29).

Proof The proof is straightforward and hence omitted. □

It is well known that the mixture models may suffer from the label-switching problem (Stephens 2000) such that the parameters are not identifiable. A common strategy of alleviating this problem is to impose a constraint that makes the components unique, e.g., $\pi_1 > \dots > \pi_g$. Suppose that the mixture parameters are identifiable and have a bounded likelihood. Redner and Walker (1994) showed that the ML estimator $\hat{\Theta}$ can converge in probability to the true values of Θ and in distribution to a multivariate normal distribution with mean vector Θ and covariance matrix being the inverse of

Fisher information matrix. That is, as $n \rightarrow \infty$,

$$\hat{\boldsymbol{\theta}} \xrightarrow{P} \boldsymbol{\theta}, \quad \text{and} \quad \sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \xrightarrow{d} \mathcal{N}_m(\mathbf{0}, \mathcal{I}^{-1}(\boldsymbol{\theta})), \quad (32)$$

where $m = \text{Dim}(\boldsymbol{\theta})$. Under suitable regularity conditions (Cramér 1946), the expected information matrix $\mathcal{I}(\boldsymbol{\theta})$ can be approximated by

$$\mathcal{I}_o(\hat{\boldsymbol{\theta}}) \approx - \sum_{j=1}^n \mathbf{H}_j(\hat{\boldsymbol{\theta}} | y_j^o). \quad (33)$$

Thus, the large sample properties of (32) and (33) are useful for proceeding with hypothesis testing and constructions of confidence intervals. In practice, standard errors of parameter estimates can be extracted from the square root of the diagonal elements of $\mathcal{I}_o^{-1}(\hat{\boldsymbol{\theta}})$.

6 Numerical illustrations

6.1 Example 1: Ozone day detection data

Ozone is an allotrope of oxygen and is known to be much less stable than the diatomic allotrope O_2 . It is generated by binding dioxygen and oxygen atom together which is decomposed by high-energy radiation from the effect of ultraviolet light and atmospheric electrical discharges. Most of ozone is in stratosphere, and the consistency is highest at distance 20 km to 30 km above the earth's surface called the ozone layer region, which absorbs ultraviolet radiation (UV) that is especially harmful to living creatures.

Our first example concerns the one-hour (1-h) and eight-hour (8-h) peak of ground ozone level data with genuine missing values, which were collected by Zhang and Fan (2008) from 1998 to 2004 at the Houston, Galveston and Brazoria (HGB) area of Texas, USA. Both of which consist of more than 2500 instances with 72 continuous attributes containing various measures of air pollutant and meteorological information for the HGB area. Moreover, there is a nominal variable whose label equals 1 for the ozone day and 0 for the normal day. In July 1997, the Environmental Protection Agency (EPA) announced a new standard 8-h 0.08 parts per million (ppm) in place of the previous 0.12 ppm 1-h standard. The difference between the two datasets is the measure of air pollution at peaks during 1 h and 8 h. It is interesting to notice that only a few percent belongs to ozone days. Specifically, there are 73 (2.88%) ozone days versus 2463 normal days in the 1-h samples and 160 (6.31%) ozone days versus 2374 normal days in the 8-h samples. The data are publicly available from the UCI machine learning database repository (Frank and Asuncion 2010). There are plenty of missing values in these data. A more detailed account of the fractions of missing values is summarized in Table 1. From the table, we found that both datasets exhibit similar missing patterns across different percentage ranges. In addition, the normal days tend to have greater proportions of missing values than ozone days.

Table 1 Number of normal and ozone days with different percentage ranges of missing values for 1-h and 8-h ozone data

Missing rate (%)	One hour		Eight hour	
	Normal	Ozone	Normal	Ozone
0	1791 (72.72%)	57 (78.08%)	1719 (72.41%)	128 (80.00%)
> 0–5	261 (10.60%)	6 (8.22%)	209 (8.80%)	12 (7.50%)
> 5–10	70 (2.84%)	1 (1.37%)	67 (2.82%)	4 (2.50%)
> 10–15	12 (0.49%)	2 (2.74%)	11 (0.46%)	3 (1.88%)
> 15–20	11 (0.45%)	0 (0.00%)	9 (0.38%)	2 (1.25%)
> 20–25	6 (0.24%)	0 (0.00%)	6 (0.25%)	0 (0.00%)
> 25–30	82 (3.33%)	3 (4.11%)	82 (3.45%)	3 (1.88%)
> 30	275 (11.17%)	4 (5.48%)	271 (11.42%)	8 (5.00%)
Total	2463 (100%)	73 (100%)	2374 (100%)	160 (100%)

Values in parentheses represent the associated percentage

To compare the convergence behavior of the EM, AECM and ECM algorithms, we fit the MFA model to the 1-h and 8-h datasets. The number of components g is fixed at 2 to reflect two intrinsic clusters (normal versus ozone days), whereas the number of factors q varies from 1 to 40, though in principle the maximum on the basis of (16) should be 60. Such a consideration arises from the fact that the three algorithms may fail to converge due to an over-fitting of MFA when q is over 40. Table 2 compares the top three models for getting the highest $BIC^a = -BIC/2$ and the required CPU time. Comparing the resulting fitting performances for the 1-h and 8-h datasets, the optimal numbers of factors chosen by the three algorithms are between 24 to 28. It can be observed that the ECM algorithm converges significantly faster than EM and AECM and attains higher BIC^a values. Accordingly, a larger BIC^a value indicates a better-fitting model. Even though the ECM algorithm is more efficient, it takes nearly 2 machine hours to complete the learning. From a practical viewpoint, the running time of the two-stage procedures seems a bit too long.

To speed up the learning process, we adopt the AMFA algorithm which generally takes within 3 minutes to converge (Table 2). Note that different initial values $\hat{q}^{(0)}$ may yield slightly different optimal q ; they all ended up with similar BIC^a values to those of ECM. Figure 1 displays the evolvement of BIC^a values against number of iterations starting by $\hat{q}^{(0)} = 10, 12$ and 14. It is obvious that the AMFA algorithm performs similarly to get the final $q \in (24, 28)$ (Table 2), but it converges very quickly in the sense of fewer iterations as well as the CPU time as compared to the two-stage algorithms.

6.2 Example 2: Diabetes in Pima Indian women

Diabetes mellitus (DM), called diabetes for short, is a disease that occurs when patients' blood sugar is at high level for a long time. Symptoms of diabetes include frequent urination, increased thirst, increased hunger and decreased weight. There are

Table 2 Performance comparison of the AMFA algorithm and three two-stage procedures

Data	Output	EM			AECM			ECM			AMFA		
		<i>q</i>	<i>m</i>	BIC ^a	<i>q</i>	<i>m</i>	BIC ^a	<i>q</i>	<i>m</i>	BIC ^a	<i>q</i>	<i>m</i>	BIC ^a
1 h	1st	26	3383	-8950.089	25	3289	-8868.546	26	3383	-8674.283	26	3383	-8694.113
	2nd	27	3475	-9003.579	24	3193	-9055.903	27	3475	-8686.561	27	3475	-8702.277
	3rd	28	3565	-9036.587	26	3383	-9155.406	28	3565	-8702.450	28	3565	-8746.067
	CPU time	≈ 26 (h)			≈ 23 (h)			≈ 2 (h)			≈ 3 (min)		
8 h	1st	27	3475	-8954.971	24	3193	-9074.767	26	3383	-8690.646	26	3383	-8670.046
	2nd	26	3383	-8963.410	26	3383	-9145.713	27	3475	-8707.358	27	3475	-8718.522
	3rd	25	3289	-9041.546	25	3289	-9170.087	28	3565	-8745.635	28	3565	-8761.077
	CPU time	≈ 26 (h)			≈ 23 (h)			≈ 2 (h)			≈ 3 (min)		

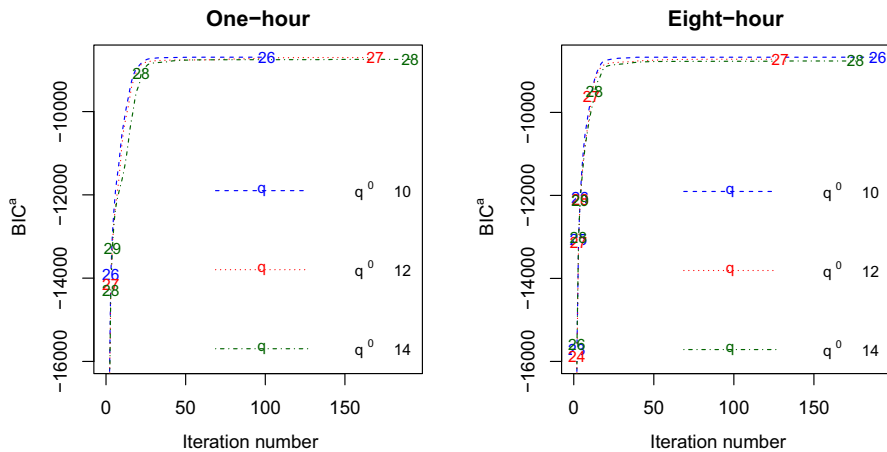


Fig. 1 The evolution of BIC^a for fitting the AMFA algorithm

three main types of diabetes; no matter which one of types, if keep untreated, they will cause complications. Type-1 DM is known as insulin-dependent diabetes mellitus (IDDM), resulting from autoimmune destruction of the β -cells that will cause shortage of insulin. Type-2 DM, known as non-insulin-dependent diabetes mellitus (NIDDM), is a long-term metabolic disorder due to obesity and lack of exercise. Gestational diabetes mellitus (GDM) is a situation that women without diabetes have high blood sugar levels during pregnancy.

The second example to which we testify the usefulness of our methods is the Pima Indian women data, publicly available at the UCI machine learning database repository (Frank and Asuncion 2010). The dataset comprised $p = 8$ attributes, including number of times pregnant (x_1), plasma glucose concentration (x_2), diastolic blood pressure in mmHg (x_3), triceps skinfold thickness in mm (x_4), 2-h serum insulin in μ U per ml (x_5), body mass index (x_6), diabetes pedigree function (x_7) and age (x_8), for 268 diabetic and 500 non-diabetic female patients. A detailed description of the eight attributes and their numbers of missing units is summarized in Table 2 of Wang and Lin (2015). Overall, there are 652 unobserved measurements over the total of $8 \times 768 = 6144$ measurements, leading to a missing rate of 10.61%.

Because there are two known clusters labeled by ‘diabetic’ and ‘non-diabetic,’ we therefore focus on the comparison under a two-component MFA model with $q = 1-4$. We implement the EM, AECM, ECM and AMFA algorithms to fit MFA models to the data. Figure 2 displays the typical evolution of the $BIC^a = -BIC/2$ values trained by the four algorithms. As can be seen, all algorithms achieve nearly the same BIC^a values except for EM which suffers from premature convergence for $q = 2$ and 3. The premature convergence means that evolutionary process gets stagnated too early to obtain the optimal value or results in a drop of successive log-likelihood values.

Table 3 shows the performance of the best chosen model along with the required CPU time obtained by running the three two-stage EM-based algorithms and the one-stage AMFA algorithm. Observing the table, the AECM, ECM and AMFA algorithms

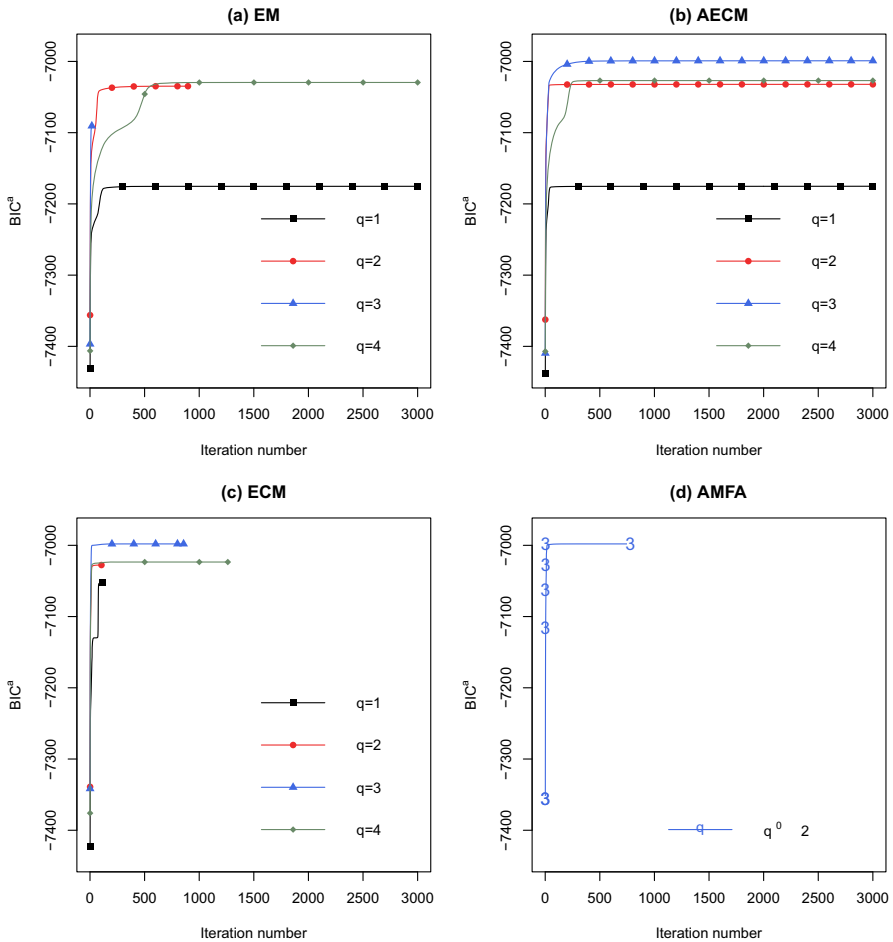


Fig. 2 The typical evolutions of BIC^a for EM, AECM, ECM and AMFA algorithms fitted to the Pima Indian women data

Table 3 The performance of the three two-stage procedures and the proposed AMFA algorithm for the fitting of MFA models to the Pima Indian women data

	EM	AECM	ECM	AMFA
q	4	3	3	3
m	85	75	75	75
ℓ_{\max}	-6747.159	-6749.935	-6748.867	-6748.867
BIC^a	-7029.521	-6999.077	-6998.010	-6998.009
CPU time (in s)	55.8	79.28	15.92	5.16

select the same number of factors ($q = 3$) and attain approximately the same BIC^a value. The EM algorithm offers an inferior solution ($q = 4$) due to premature conver-

gence. It is also readily seen that the AMFA algorithm takes the least amount of CPU time to reach the optimal solution ($q = 3$). The empirical results provide evidence that the AMFA algorithm is more efficient than the two-stage algorithms.

Table 4 summarizes the resulting parameter estimates together with the associated standard errors obtained by inverting (33) for the chosen three-factor MFA model. The results indicate that all the mean estimates of eight variables are significantly less than zero for component 1 and greater than zero for component 2, suggesting that all variables contribute equally important to the separation between groups.

6.3 Example 3: Simulation based on cereal data with synthetic missing values

Our third example concerns the cereal data reported originally by Lattin et al. (2003). Among these data, 116 participants appraise 12 cereals on $p = 25$ attributes, including filling, salt, soggy and so on. Some of respondents assessed more than one cereal so that there are $n = 235$ observations. Participants used the five-point scale to demonstrate the level to each characteristic of cereal brands. Zhao and Shi (2014) implemented the AFA algorithm to fit a single-component MFA model to the cereal data and concluded the most appropriate number of factors attains at $q = 4$.

We consider the fitting of the MFA models to the cereal data with g varying from 1 to 3 and q from 1 to $q_{\max} = 18$. Table 5 compares the performance of the AMFA and three two-stage procedures for estimating 12 different MFA models, and outputs include the optimal choice of q , final BIC^a value as well as the CPU time. As one can see, all algorithms give the same final ML solution, among the AMFA demands substantially smaller CPU time (less than 1 s). It is also worth noting that the two-component MFA model with three factors provides the best fit. Contrary to the previous studies, the underlying distribution of the cereal data should be multimodal rather than unimodal.

To study the computational efficiency of the AMFA algorithm relative to the two-stage procedures for learning MFA models in the presence of missing values, artificial missing data were generated by deleting randomly from the complete cereal data studied under three proportions of missingness: 5%, 10% and 20%. In this study, we focus on comparing the learning of two-component MFA models as g is *a priori* determined to be 2 based on the best fitting of the complete cereal data. Table 6 summarizes the results, including the CPU time, BIC^a values and the selected number of factors q averaged over 100 Monte Carlo replications together with their corresponding standard deviations. The computational results of EM are not included because it urges to converge prematurely, especially when the missing rate rises. We observe that both AEEM and ECM procedures give somewhat similar BIC^a values, whereas the AMFA algorithm is prone to provide slightly inferior BIC^a scores. Apparently, the numbers of factors selected by the three estimating procedures are all getting smaller as the missing rate increases. In terms of computational efficiency, the AMFA algorithm represents a much faster computational speed than the two-stage procedures. Consequently, the optimal values of q determined by the three considered procedures are quite close, while the AMFA algorithm is shown to be more reliable as it yields lower standard deviations.

Table 4 ML results from fitting the MFA model with $g = 2$ and $q = 3$ to the Pima Indian women data

Variable	Component 1			Component 2			Ψ_2
	μ_1	A_1	Ψ_1	μ_2	A_2	Ψ_2	
x_1	-0.5671 (0.0311)	-0.0619 (0.0196)	0.2857 (0.0272)	0.4549 (0.0550)	-0.0996 (0.0380)	0.1421 (0.0428)	0.8884 (0.0737)
x_2	-0.4428 (0.0446)	0.1499 (0.0294)	-0.0221 (0.0334)	0.3522 (0.0547)	0.1735 (0.0373)	-0.9140 (0.0926)	0.2671 (0.1606)
x_3	-0.3571 (0.0560)	0.1368 (0.0401)	0.1142 (0.0460)	0.2808 (0.0485)	0.3104 (0.0340)	0.0028 (0.0378)	0.6654 (0.0544)
x_4	-0.2548 (0.0610)	0.6980 (0.0424)	0.0439 (0.0516)	0.1609 (0.0602)	0.5526 (0.0416)	-0.0435 (0.0445)	0.6703 (0.0644)
x_5	-0.4401 (0.0327)	0.1857 (0.0225)	-0.0672 (0.0289)	0.3046 (0.0906)	0.1403 (0.0566)	-0.6220 (0.0698)	1.0079 (0.1284)
x_6	-0.2347 (0.0572)	0.9572 (0.0530)	-0.0012 (0.0446)	0.1814 (0.0509)	0.9903 (0.0558)	0.0037 (0.0404)	0.0050 (0.0859)
x_7	-0.2732 (0.0385)	0.0910 (0.0249)	-0.0458 (0.0290)	0.2192 (0.0592)	0.1200 (0.0412)	-0.0754 (0.0469)	1.2623 (0.0923)
x_8	-0.7245 (0.0174)	0.0510 (0.0109)	0.2131 (0.0201)	0.5812 (0.0526)	-0.2305 (0.0354)	-0.0084 (0.0401)	0.4843 (0.0711)

Values within parentheses are standard errors of ML estimates

Table 5 Comparison of the AMFA and two-stage algorithms for the fitting of cereal data with MFA models

g	Output	EM	AECM	ECM	AMFA
1	q	4	4	4	4
	BIC ^a	-7418.10	-7418.10	-7418.10	-7418.10
	CPU time (in s)	43.41	33.40	12.72	0.07
2	q	3	3	3	3
	BIC ^a	-7250.00	-7250.00	-7250.00	-7250.00
	CPU time (in s)	119.67	94.56	18.73	0.69
3	q	2	2	2	2
	BIC ^a	-7297.19	-7297.19	-7297.19	-7297.19
	CPU time (in s)	181.28	133.30	17.57	0.25

Table 6 Simulation results based on 100 replications for the cereal data with synthetic missing values

Missing rate	Output	AECM	ECM	AMFA
5%	q	2.87 (0.68)	2.80 (0.68)	2.99 (0.10)
	BIC ^a	-6944.12 (30.35)	-6944.56 (28.20)	-7000.77 (105.89)
	CPU time (in s)	3060.14 (205.38)	306.30 (54.34)	9.62 (3.51)
10%	q	2.82 (0.80)	2.75 (0.81)	2.84 (0.37)
	BIC ^a	-6636.07 (44.07)	-6634.15 (43.04)	-6715.71 (108.85)
	CPU time (in s)	5287.64 (325.97)	580.40 (89.73)	16.64 (6.89)
20%	q	2.58 (0.78)	2.64 (0.80)	2.30 (0.46)
	BIC ^a	-5996.88 (43.95)	-5999.94 (47.25)	-6049.52 (87.29)
	CPU time (in s)	6390.59 (356.57)	1029.74 (184.92)	18.08 (13.07)

6.4 Example 4: Simulation based on artificial data

A simulation experiment is conducted to examine the performance of the EM, AECM, ECM and AMFA algorithms in recovering the true underlying parameter values when the number of components is correctly specified. We generate 500 Monte Carlo data of $p = 8$ attributes, and sample sizes $n=150$ (small), 300 (moderate) and 600 (relatively large) form a MFA model with $g = 3$ components and $q = 3$ factors. The setup of true values of parameters is

$$\pi_i = 1/3, \quad \mu_i = 2i \cdot \mathbf{1}_8, \quad \mathbf{B}_i = \text{Unif}(8, 3), \quad \Psi_i = 0.3i \cdot \mathbf{I}_8, \quad \text{for } i = 1, 2, 3,$$

where $\text{Unif}(p, q)$ denotes a $p \times q$ matrix of random numbers drawn from a uniform distribution on the unit interval $(0, 1)$, $\mathbf{1}_8$ indicates a column vector of length 8 with all entries equal to 1, and \mathbf{I}_8 denotes a 8×8 identity matrix.

For each simulated dataset, we compare the estimation accuracy and efficiency of the four considered algorithms on the fitting of the three-component MFA model with q ranging from 1 to $q_{\max} = 4$. To investigate the estimation accuracies, we report in Table 7 the average norm of the bias of parameters $\{\pi_i, \mathbf{B}_i, \Psi_i\}_i^3$, since their true values are known, summarized over 500 replications along with the standard deviations in parentheses. For a specific parameter vector θ of length d with true values $(\theta_1, \dots, \theta_d)$, the norm of the bias is defined as $\{(\hat{\theta}_1^{(r)} - \theta_1)^2 + \dots + (\hat{\theta}_d^{(r)} - \theta_d)^2\}^{1/2}$, where $\hat{\theta}_s^{(r)}$ is the estimate of θ_s , for $s = 1, \dots, d$, obtained at the r th replication. The final converged BIC values and consumed CPU time are also shown in the last two columns for the sake of comparison. From the numerical results listed in Table 7, the four algorithms produce very similar estimation accuracy and the AMFA algorithm demands the lowest computational cost in all cases. Notice that the EM algorithm is slowest, while the AECM algorithm gives slightly lower accuracy for some cases. It is also noteworthy that the norm of the bias as well as the standard deviations for all parameter vectors is getting smaller when the sample size increases. This indicates empirical evidence that the estimates obtained by the four estimating procedures have desirable asymptotic properties. Although the above simulation experiment is somewhat limited, it demonstrates the AMFA algorithm can yield comparable parameter estimates and converged log-likelihood to the two-stage methods which may demand a prohibitively higher computational burden.

7 Conclusion

We have devised a one-stage AMFA procedure that seamlessly integrates the selection of the number of factors into parameter estimation for fast learning MFA model with possibly missing values. The efficiency of such an automatic scheme stems from less amount of missing information and quick determination of factor dimensions based on the eigendecomposition of local sample covariance matrices. Two auxiliary permutation matrices are incorporated into all considered estimating procedures that allow for ease of algorithmic representation and computer coding. We have further explicitly derived the Hessian matrix of the MFA model with missing information; thereby, the asymptotic standard errors of the ML estimates can be directly obtained without having to resort to computationally intensive bootstrap methods (Efron and Tibshirani 1986). Experiments with real and synthetic examples reveal that the AMFA algorithm performs comparably to the two-stage methods, but the computational demands are much lower, particularly for data involving a higher proportion of missing outcomes.

The proposed one-stage approach is limited to fast learning MFA under a given number of components (g). Although pointed out by many researchers, it is still an open question to develop an automated algorithm for fast determination of the best

Table 7 Simulation results for assessing the performance of the EM, AECM, ECM and AMFA algorithms summarized over 500 trials

Sample size (n)	Algorithm	π ($\frac{1}{3}\mathbf{I}_3$)	μ_1 ($2\mathbf{I}_8$)	μ_2 ($4\mathbf{I}_8$)	μ_3 ($6\mathbf{I}_8$)	ψ_1 ($0.3\mathbf{I}_8$)	ψ_2 ($0.6\mathbf{I}_8$)	ψ_3 ($0.9\mathbf{I}_8$)	BIC	CPU time (s)
150	EM	0.0836	0.5389	0.7424	0.7665	0.4140	0.8688	1.1870	-1973.12	26.41
		(0.0558)	(0.3123)	(0.3701)	(0.3700)	(0.1056)	(0.1877)	(0.2465)	(36.08)	(7.318)
	AECM	0.0886	0.5470	0.7869	0.7717	0.4140	0.8715	1.1900	-1973.01	21.70
		(0.0573)	(0.3211)	(0.4386)	(0.3661)	(0.1063)	(0.1918)	(0.2450)	(36.06)	(6.242)
	ECM	0.0836	0.5430	0.7491	0.7571	0.4143	0.8707	1.1920	-1973.22	1.15
		(0.0524)	(0.3148)	(0.3963)	(0.3536)	(0.1061)	(0.1887)	(0.2481)	(36.08)	(0.812)
300	AMFA	0.0836	0.5430	0.7491	0.7571	0.4143	0.8707	1.1920	-1973.22	0.37
		(0.0524)	(0.3148)	(0.3963)	(0.3536)	(0.1061)	(0.1887)	(0.2481)	(36.08)	(0.258)
	EM	0.0645	0.3808	0.5634	0.5638	0.3373	0.7693	1.0820	-3798.61	48.71
		(0.0450)	(0.2121)	(0.3011)	(0.2830)	(0.0889)	(0.1713)	(0.2369)	(47.06)	(8.499)
	AECM	0.0658	0.3829	0.5719	0.5694	0.3370	0.7692	1.0800	-3798.61	37.46
		(0.0461)	(0.2157)	(0.3092)	(0.2833)	(0.0887)	(0.1708)	(0.2375)	(46.97)	(4.923)
ECM	0.0637	0.3817	0.5671	0.5608	0.3409	0.7784	1.1010	-3798.52	4.38	
	(0.0419)	(0.2132)	(0.3188)	(0.2780)	(0.0925)	(0.1731)	(0.2481)	(47.08)	(4.607)	
600	AMFA	0.0644	0.3800	0.5697	0.5664	0.3317	0.7604	1.0600	-3798.30	1.15
		(0.0447)	(0.2119)	(0.3190)	(0.2911)	(0.0961)	(0.1836)	(0.2740)	(47.57)	(1.242)
	EM	0.0389	0.2456	0.3719	0.3824	0.2453	0.6337	0.8945	-7386.96	55.67
		(0.0310)	(0.1197)	(0.1857)	(0.2012)	(0.0874)	(0.1745)	(0.2439)	(75.15)	(6.143)
	AECM	0.0389	0.2456	0.3722	0.3831	0.2454	0.6335	0.8959	-7386.97	44.15
		(0.0311)	(0.1197)	(0.1860)	(0.2021)	(0.0873)	(0.1747)	(0.2454)	(75.16)	(6.575)
ECM	0.0389	0.2455	0.3722	0.3830	0.2508	0.6535	0.9448	-7386.80	5.48	
	(0.0308)	(0.1199)	(0.1839)	(0.2033)	(0.0953)	(0.1845)	(0.2654)	(75.24)	(5.649)	
AMFA	0.0389	0.2452	0.3718	0.3818	0.2608	0.6590	0.9555	-7386.86	1.86	
	(0.0305)	(0.1170)	(0.1822)	(0.2006)	(0.1025)	(0.1915)	(0.2803)	(75.68)	(2.509)	

True parameter values and the associated Monte Carlo standard deviations are shown in parentheses

pair (g, q) without sacrificing too much computational complexity. The BIC is adopted to choose the most plausible q due to its consistence in model selection. However, other criteria such as the integrated completed likelihood (ICL; Biernacki et al. 2000) or approximate weight of evidence (AWE; Banfield and Raftery 1993) can also be used and might perform better than BIC in discovering the true numbers of factors or clusters. Nevertheless, all these criteria suffer from the problem of over-penalization in dealing with incomplete data when the rate of missingness gets large (Ibrahim et al. 2008). It would be a worthwhile future task to develop an improved procedure that can take the aforementioned issues into account. In the formulation of MFA, the number of local factors per component is assumed to be equal to guarantee the global structural identifiability. However, such a restriction might cause an overlearning effect when the number of component (g) or factors (q) increases. To remedy this deficiency, an attempt on the extension of our current approach allowing for automatic determination of varying dimensions of local factors $\{q_i\}_{i=1}^g$ deserves further attention. Finally, it is of interest to extend the current approach to the mixture of t factor analyzers (McLachlan et al. 2007; Wang and Lin 2013) for modeling incomplete high-dimensional data that have heavy-tailed behavior in a more efficient manner.

Acknowledgements The authors gratefully acknowledge the editors and two anonymous referees for their comments and suggestions that greatly improved the quality of this paper. We are also grateful to Ms. Ying-Ting Lin for her assistance in initial simulations. This research was supported by the Ministry of Science and Technology of Taiwan under Grant Nos. 107-2628-M-035-001-MY3 and 107-2118-M-005-002-MY2.

References

- Banfield JD, Raftery AE (1993) Model-based Gaussian and non-Gaussian clustering. *Biometrics* 49:803–821
- Basford KE, Greenway DR, McLachlan GJ, Peel D (1977) Standard errors of fitted means under normal mixture models. *Comput Stat* 12:1–17
- Biernacki C, Celeux G, Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans Pattern Anal Mach Intell* 22:719–725
- Boldea O, Magnus JR (2009) Maximum likelihood estimation of the multivariate normal mixture model. *J Am Stat Assoc* 104:1539–1549
- Cramér H (1946) *Mathematical methods of statistics*. Princeton University Press, Princeton
- Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J R Stat Soc B* 39:1–38
- Efron B, Tibshirani R (1986) Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Stat Sci* 1:54–75
- Efron B, Tibshirani R (1993) *An introduction to the bootstrap*. Chapman & Hall, London
- Fraley C, Raftery AE (1998) How many clusters? Which clustering method? Answers via model-based cluster analysis. *Comput J* 41:578–588
- Fraley C, Raftery AE (2002) Model-based clustering, discriminant analysis, and density estimation. *J Am Stat Assoc* 97:611–612
- Frank A, Asuncion A (2010) UCI machine learning repository. School of Information and Computer Science, University of California, Irvine, CA. <http://archive.ics.uci.edu/ml>
- Ghahramani Z, Beal MJ (2000) Variational inference for Bayesian mixture of factor analysers. In: Solla S, Leen T, Muller K-R (eds) *Advances in neural information processing systems 12*. MIT Press, Cambridge, pp 449–455
- Ghahramani Z, Jordan MI (1994) Supervised learning from incomplete data via an EM approach. In: Cowan JD, Tesarro G, Alspector J (eds) *Advances in neural information processing systems, vol 6*. Morgan Kaufmann, San Francisco, pp 120–127

- Ghahramani Z, Hinton GE (1997) The EM algorithm for mixtures of factor analyzers. Technical report no. CRG-TR-96-1, University of Toronto
- Golub GH, Van Loan CF (1989) Matrix computations, 2nd edn. Johns Hopkins University Press, Baltimore, MD
- Ibrahim JG, Zhu H, Tang N (2008) Model selection criteria for missing data problems via the EM algorithm. *J Am Stat Assoc* 103:1648–1658
- Keribin C (2000) Consistent estimation of the order of mixture models. *Sankhya* 62:49–66
- Lattin J, Carrol JD, Green PE (2003) Analyzing multivariate data. Brooks/Cole, Pacific Grove, CA
- Ledermann W (1937) On the rank of the reduced correlational matrix in multiple-factor analysis. *Psychometrika* 2:85–93
- Lin TI, Lee JC, Ho HJ (2006) On fast supervised learning for normal mixture models with missing information. *Pattern Recogn* 39:1177–1187
- Little RJA, Rubin DB (2002) Statistical analysis with missing data, 2nd edn. Wiley, New York
- McLachlan GJ, Peel D (2000) Finite mixture models. Wiley, New York
- McLachlan GJ, Bean RW, Peel D (2002) A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics* 18:413–422
- McLachlan GJ, Peel D, Bean RW (2003) Modelling high-dimensional data by mixtures of factor analyzers. *Comput Stat Data Anal* 41:379–388
- McLachlan GJ, Bean RW, Jones LBT (2007) Extension of the mixture of factor analyzers model to incorporate the multivariate t -distribution. *Comput Stat Data Anal* 51:5327–5338
- Meng XL, van Dyk D (1997) The EM algorithm—an old folk-song sung to a fast new tune. *J Roy Stat Soc B* 59:511–567
- Meng XL, Rubin DB (1993) Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika* 80:267–278
- Montanari A, Viroli C (2011) Maximum likelihood estimation of mixtures of factor analyzers. *Comput Stat Data Anal* 55:2712–2723
- Redner RA, Walker HF (1984) Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev* 26:195–239
- Schwarz G (1978) Estimating the dimension of a model. *Ann Stat* 6:461–464
- Stephens M (2000) Dealing with label switching in mixture models. *J Roy Stat Soc B* 62:795–809
- Ueda N, Nakano R, Ghahramani Z, Hinton GE (2000) SMEM algorithm for mixture models. *Neural Comput* 12:2109–2128
- Wang WL, Lin TI (2013) An efficient ECM algorithm for maximum likelihood estimation in mixtures of t -factor analyzers. *Comput Stat* 28:751–769
- Wang WL, Lin TI (2015) Robust model-based clustering via mixtures of skew- t distributions with missing information. *Adv Data Anal Classif* 9:423–445
- Zhang K, Fan W (2008) Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond. *Knowl Inf Syst* 14:299–326
- Zhao JH, Shi L (2014) Automated learning of factor analysis with complete and incomplete data. *Comput Stat Data Anal* 72:205–218
- Zhao JH, Yu PLH (2008) Fast ML estimation for the mixture of factor analyzers via an ECM algorithm. *IEEE Trans Neural Netw* 19:1956–1961
- Zhao JH, Yu PLH, Jiang Q (2008) ML estimation for factor analysis: EM or non-EM? *Stat Comput* 18:109–123