mathematics

MDPI

Article

Simultaneous Bayesian Clustering and Model Selection with Mixture of Robust Factor Analyzers

Shan Feng [1,2,*], Wenxian Xie [1] and Yufeng Nie [1,*]

[1] School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China; wenxianxie@nwpu.edu.cn

[2] College of Statistics, Xi'an University of Finance and Economics, Xi'an 710100, China

[*] Correspondence: fengshan1912@mail.nwpu.edu.cn (S.F.); yfnie@nwpu.edu.cn (Y.N.)

Abstract: Finite Gaussian mixture models are powerful tools for modeling distributions of random phenomena and are widely used for clustering tasks. However, their interpretability and efficiency are often degraded by the impact of redundancy and noise, especially on high-dimensional datasets. In this work, we propose a generative graphical model for parsimonious modeling of the Gaussian mixtures and robust unsupervised learning. The model assumes that the data are generated independently and identically from a finite mixture of robust factor analyzers, where the features' salience is adjusted by an active set of latent factors to allow a violation of the local independence assumption. For the model inference, we propose a structured variational Bayes inference framework to realize simultaneous clustering, model selection and outlier processing. Performance of the proposed algorithm is evaluated by conducting experiments on artificial and real-world datasets. Moreover, an application on the high-dimensional machine learning task of handwritten alphabet recognition is introduced.

Keywords: Bayesian inference; feature selection; mixture of factor analyzers; robust clustering; structured variational Bayes

MSC: 62H30; 62F15; 62H22

check for updates

Citation: Feng, S.; Xie, W.; Nie, Y. Simultaneous Bayesian Clustering and Model Selection with Mixture of Robust Factor Analyzers. *Mathematics* **2024**, *12*, 1091. https://doi.org/10.3390/math12071091

Academic Editor: Sanku Dey

Received: 29 February 2024
Revised: 29 March 2024
Accepted: 2 April 2024
Published: 4 April 2024

1. Introduction

Finite Gaussian mixture models are powerful tools for modeling distributions of random phenomena. They are widely used for unsupervised classification tasks and lay the foundation for many deep learning-based clustering algorithms, e.g., [1,2]. However, competitive performance of the Gaussian mixture model cannot be expected on high-dimensional datasets due to the curse of dimensionality [3]. The impact of redundancy and noise can degrade the model's interpretability and efficiency, which is crucial in many application fields such as molecular biology and the clinical medicine [4]. Since the intrinsic dimensions of high-dimensional data are usually much less than their original feature space, it is possible to improve the clustering performance via dimension-reduction methods [5].

Feature-selection approaches are designed to retain a subset of features that are informative and discriminant for clustering. The two-stage methods implement the feature selection and clustering separately, which consider the preselected features as input without regarding the subsequent clustering algorithms [6,7]. But as choosing the feature subset and clustering are highly dependent problems, to circumvent the loss of information, incorporating the feature selection in the clustering algorithms and constructing an integrated objective function is suggested [3,4]. Pan and Shen [8] proposed a penalized likelihood approach for unsupervised feature selection where they used an L_1 penalty to shrink the component means. A similar approach was also suggested in [9], where the feature-selection consistency via the penalization was further studied. However, as both approaches are highly dependent on the choosing of penalization parameters, cross-validation or criterion-based model selection is required to tune the parameters.

A different stream of research casts the feature selection as parameter-estimation problems [4,10–12], where the random variable "feature saliency" is introduced to quantify the relevance of features to class assignment. This approach is efficient as neither combinatorial search through the feature subsets nor tuning of parameters is required. The feature selection and the clustering can be performed simultaneously in a principled and automatic way. Zhang et al. [13] extended this method to the Student's t mixture model, which has higher tolerance to outliers and therefore is more robust for clustering and feature selection. As an extension to the work of Zhang et al., Sun and Zhou [14] made a full-Bayesian treatment for the model and proposed the structured variational Bayesian (VB) approach, which takes into consideration the estimation uncertainty of all model parameters and can deliver a tighter bound to the marginal likelihood than the mean-field approximated VB algorithms. Their model was extended further in [3] to consider the class-specific feature saliency for Bayesian feature selection.

Feature-selection approaches commonly assume that the features are conditionally independent given the latent class variable, which is equivalent to adopting a diagonal component covariance matrix structure in the Gaussian mixture model. While this assumption greatly facilitates computational efficiency, it can be easily violated in real-world datasets [15]. For the linear regression analysis, Fan et al. [16] conducted a synthetic study indicating that when the covariates are highly correlated exact recovery of the active set from the solution path of LASSO will be difficult. For classification, as has been mentioned in [17], ignoring the dependence relationship across features may undermine the reliability of the algorithms and lead to misleading conclusions about the features' salience.

In [18], the local independence assumption for the Gaussian mixture model was relaxed by a block-diagonal specification of the component covariance matrix, where the features are partitioned into several disconnected groups in each class. However, as the total number of block-diagonal structures increases as the Bell number [19], searching for the optimal model can be difficult, especially in the high-dimensional cases. Galimberti and Soffritti [18] proposed a hierarchical aggregative strategy based on the BIC criterion to perform a nonexhaustive search of the structures. But this method cannot promise to find the optimal model. Ruan et al. [20] extended the graphical LASSO method to the context of the Gaussian mixture model and proposed a penalized likelihood approach for a sparse solution of the component covariance matrices. But the penalization parameters still need to be selected.

Different to the block-diagonal specification, the model of mixture of factor analyzers assumes a factor analysis-based decomposition structure for the component covariance matrices, where the local dependence between features is explained by a few latent factors [21–24]. Typically, the number of factors for each component needs to be specified in advance of the model fitting or the model-selection criterion used to select the optimal number of factors. However, while presetting of the number may lead to over-fitted or over-simplified models, conducting exhaustive searches over the model space is computationally expensive. The shrinkage prior methods were proposed to achieve an automatic latent dimension reduction. Wang and Lan [25] imposed automatic relevance determination prior [26] on the factor loading matrices. Multiplicative Gamma process shrinkage priors [27] in the infinite factor analysis was used by Murphy et al. [28]. They also suggested an adaptive Gibbs sampling algorithm where the factors with negligible loadings are removed gradually during the iterations. Therefore, the computational efficiency can be improved.

In this paper, we develop further the Student's t mixture model of Sun and Zhou [14] for Bayesian clustering and feature selection to tackle the cases where features are correlated in the mixture component. While the model in [14] defines the feature saliency under the local independence assumption, we introduce the factor-adjusted feature saliency, where the salience of each feature is evaluated by conditioning on the latent factors. Taken as a whole, the extension produces parsimonious, flexible and robust modeling for the mixture of factor analyzers. Moreover, motivated by the Bayesian model selection method in [29] for the linear regression analysis, instead of using the shrinkage priors, we propose an automatic inference scheme for the number of factors by introducing the random variable of factor activity. Then, the problems of feature selection, latent dimension reduction, outlier processing and clustering can be integrated together as the inference for a Bayesian hierarchical latent variable model. We continue the work in [14] to adopt a full Bayesian treatment, where proper prior distributions are assumed for the model parameters. The structured VB inference framework that improves the evidence lower bound (ELBO) for the proposed model is presented, where a "drop-out" sampling technique [30] can be applied immediately to ease the computation.

The rest of this paper is organized as follows: In Section 2, we introduce the Student's *t* mixture model proposed by Sun and Zhou [14], which has provided the base of our study. In Section 3, we develop the proposed mixture of robust factor analyzers, which is present as a hierarchical latent variable model for Bayesian inference. In Section 4, the structured VB inference framework for the proposed model is established. Section 5 justifies the performance of the developed model and algorithm on the synthetic data and presents the evaluation results based on some real-world datasets. Section 6 concludes this paper, points out the limitations and suggests future research directions.

2. The Student's *t* Mixture Model for Feature Selection

We present the proposed hierarchical latent variable model starting from the Student's *t* mixture model defined in [14]. Denote the set of i.i.d. observations as $\mathbf{Y} = \{\boldsymbol{y}_n\}_{n=1}^N$, where $\boldsymbol{y}_n = (y_{n1}, y_{n2}, \ldots, y_{nd})^T \in \mathbb{R}^d$ is the d-dimensional feature data for the nth individual. The finite mixture model for clustering assumes that the data for each individual are generated from a class-specific distribution but with the class label missing, then it marginally follows a finite mixture distribution. Throughout the paper, we denote the number of mixture components or equally the number of classes as K. The latent class label for the nth individual is denoted as z_n which takes value in $\{1, 2, \ldots, K\}$. The clustering is realized by assigning each individual the class label where it has the highest posterior probability of belonging.

The Student's *t* mixture model given in [14] assumes that the features are conditionally independent given the hidden class label and each follows a Student's *t* distribution. Moreover, the relevance or irrelevance of feature to data separation is taken into account by introducing the Bernoulli latent variables $\boldsymbol{\phi}_n = (\phi_{n1}, \phi_{n2}, \ldots, \phi_{nd})^T$, which gives the mixture density of $\boldsymbol{y}_n$ as

$$p(\boldsymbol{y}_n|\boldsymbol{\phi}_n, \Theta_1) = \sum_{k=1}^{K} \pi_k \prod_{l=1}^{d} \Big[\mathcal{S}_t(y_{nl}|\mu_{kl}, \sigma_{kl}, v_{kl})^{\phi_{nl}} \mathcal{S}_t(y_{nl}|\mu_{0l}, \sigma_{0l}, v_{0l})^{1-\phi_{nl}} \Big]. \quad (1)$$

$\mathcal{S}_t(y|\mu, \sigma, v)$ is the density function of the Student's t distribution with mean, precision and degrees of freedom as μ, σ and v, respectively. For $l = 1, 2, \ldots, d$, $\phi_{nl} \in \{0, 1\}$, if $\phi_{nl} = 1$, then the lth feature is relevant to the class assignment; if $\phi_{nl} = 0$, then the lth feature is irrelevant and follows a common distribution independent of the class assignment. For $k = 1, 2, \ldots, K$, the parameter π_k ($\pi_k > 0$ and $\sum_{k=1}^K \pi_k = 1$) is the mixing proportion of class k. Let $\Theta_1 = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{v}\}$ denote the set of unknown parameters in model (1), where $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)^T$, $\boldsymbol{\mu} = \{\mu_{0l}, \{\mu_{kl}\}_{k=1}^K\}_{l=1}^d$, $\boldsymbol{\sigma} = \{\sigma_{0l}, \{\sigma_{kl}\}_{k=1}^K\}_{l=1}^d$ and $\boldsymbol{v} = \{v_{0l}, \{v_{kl}\}_{k=1}^K\}_{l=1}^d$.

The prior distribution of $\boldsymbol{\phi}_n$ is given by

$$p(\boldsymbol{\phi}_n|\boldsymbol{\beta}) = \prod_{l=1}^{d} p(\phi_{nl}|\beta_l) = \prod_{l=1}^{d} \beta_l^{\phi_{nl}} (1-\beta_l)^{1-\phi_{nl}}, \quad (2)$$

where the ϕ_{nl}'s are assumed to be mutually independent. The parameter β_l for the Bernoulli distribution of ϕ_{nl} is called the feature saliency [4] of the lth feature. It measures the importance of the feature for class assignment and is estimated to realize a "soft" feature selection. Denote $\boldsymbol{\beta} = \{\beta_l\}_{l=1}^d$.

The observed-data likelihood function can be obtained by integrating over the latent variables $\boldsymbol{\phi}_n$ in model (1), which gives

$$p(\boldsymbol{y}_n|\Theta_2) = \sum_{k=1}^{K} \pi_k \prod_{l=1}^{d} \Big[\beta_l \mathcal{S}_t(y_{nl}|\mu_{kl}, \sigma_{kl}, v_{kl}) + (1-\beta_l)\mathcal{S}_t(y_{nl}|\mu_{0l}, \sigma_{0l}, v_{0l})\Big], \tag{3}$$

where $\Theta_2 = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{v}, \boldsymbol{\beta}\}$. Statistical inference directly on the observed-data likelihood is difficult. In [14], the VB inference method was adopted where the complete-data likelihood is given by

$$p(\boldsymbol{y}_n, \boldsymbol{u}_n, \boldsymbol{\phi}_n, z_n|\Theta_2) = \prod_{k=1}^{K} \Big[\pi_k p(\boldsymbol{y}_n|\boldsymbol{u}_n, \boldsymbol{\phi}_n, z_n = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(\boldsymbol{u}_n|\boldsymbol{\phi}_n, z_n = k, \boldsymbol{v}) p(\boldsymbol{\phi}_n|\boldsymbol{\beta})\Big]^{\delta_{z_n,k}}. \tag{4}$$

At the right-hand side of (4), $\delta_{z_n,k}$ is the Kronecker delta function. The latent variables $\boldsymbol{u_n} = (u_{n1}, u_{n2}, \dots, u_{nd})^T$ are introduced by noting that the Student's t distribution can be written as a convolution of a Gaussian and a Gamma distribution [3,14]. It follows that

$$\begin{aligned} p(\boldsymbol{y}_n|\boldsymbol{u}_n, \boldsymbol{\phi}_n, z_n = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) &= \prod_{l=1}^{d} p(y_{nl}|u_{nl}, \phi_{nl}, z_n = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) \\ &= \prod_{l=1}^{d} \Big[\mathcal{N}(y_{nl}|\mu_{kl}, \sigma_{kl}u_{nl})^{\phi_{nl}} \mathcal{N}(y_{nl}|\mu_{0l}, \sigma_{0l}u_{nl})^{1-\phi_{nl}}\Big], \end{aligned} \tag{5}$$

and

$$\begin{aligned} p(\boldsymbol{u}_n|\boldsymbol{\phi}_n, z_n = k, \boldsymbol{v}) &= \prod_{l=1}^{d} p(u_{nl}|\phi_{nl}, z_n = k, \boldsymbol{v}) \\ &= \prod_{l=1}^{d} \left[\mathcal{G}\left(u_{nl}\middle|\frac{v_{kl}}{2}, \frac{v_{kl}}{2}\right)^{\phi_{nl}} \mathcal{G}\left(u_{nl}\middle|\frac{v_{0l}}{2}, \frac{v_{0l}}{2}\right)^{1-\phi_{nl}}\right]. \end{aligned} \tag{6}$$

$\mathcal{N}(y|\mu, \sigma)$ represents the Gaussian density function with mean μ and precision σ and $\mathcal{G}(u|a, b)$ is the Gamma density function

$$\mathcal{G}(u|a, b) = \frac{b^a u^{a-1}}{\Gamma(a)} \exp(-bu). \tag{7}$$

Note that by integrating over $\boldsymbol{u}_n$, $\boldsymbol{\phi}_n$ and z_n in (4), the observed-data likelihood (3) can be recovered.

3. Towards the Mixture of Robust Factor Analyzers

To tackle the cases where features are correlated in the mixture component, we relax the local independence assumption in [14] by specifying for each class a latent factor model. Specifically, for class k, we introduce the latent factors $\boldsymbol{x}_{nk} = (x_{nk1}, x_{nk2}, \dots, x_{nkp_k})^T$, where x_{nkj}'s are i.i.d. from the distribution $\mathcal{N}(0, 1)$ and p_k is the number of latent factors. After conditioning on $\boldsymbol{x}_{nk}$, the features are assumed mutually independent within the class, which corresponds to a modification of model (1) as follows:

$$p(\boldsymbol{y}_n|\boldsymbol{\phi}_n, \boldsymbol{x}_n, \Theta_3) = \sum_{k=1}^{K} \pi_k \prod_{l=1}^{d} \Big[\mathcal{S}_t(y_{nl}|\boldsymbol{w}_{kl}^T\boldsymbol{x}_{nk} + \mu_{kl}, \sigma_{kl}, v_{kl})^{\phi_{nl}} \mathcal{S}_t(y_{nl}|\boldsymbol{w}_{kl}^T\boldsymbol{x}_{nk} + \mu_{0l}, \sigma_{0l}, v_{0l})^{1-\phi_{nl}}\Big], \tag{8}$$

where $\boldsymbol{w}_{kl} \in \mathbb{R}^{p_k}$ are the factor loadings for the lth feature in class k. Denote $\boldsymbol{x}_n = \{\boldsymbol{x}_{nk}\}_{k=1}^K$ and $\Theta_3 = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{v}, \boldsymbol{w}\}$ where $\boldsymbol{w} = \{\{\boldsymbol{w}_{kl}\}_{l=1}^d\}_{k=1}^K$. In model (8), ϕ_{nl} indicates the relevance of the lth feature to class assignment after adjustment by the latent factors. Correspondingly, β_l that defines the distribution of ϕ_{nl} in (2) represents the factor-adjusted feature saliency.

Typically, in each local factor model, the latent dimensions p_k need to be specified. With overly high dimensions, the model may over-fit the data, yielding poor interpretations and hardening the computation, while with low and inadequate dimensions, the model may not be flexible enough to capture the correlations between features in each class. To enable an automatic determination, we treat the problem as another feature-selection task, but now the "features" become the latent factors. Starting from a sufficiently large p_k, we introduce in class k the Bernoulli latent variables $\boldsymbol{r}_{nk} = (r_{nk1}, r_{nk2}, \ldots, r_{nkp_k})^T$, where $r_{nkj} \in \{0, 1\}$ with $r_{nkj} = 1$ indicating that the factor x_{nkj} is active and $r_{nkj} = 0$ inactive. Model (8) then becomes

$$p(\boldsymbol{y}_n|\boldsymbol{\phi}_n, \boldsymbol{x}_n, \boldsymbol{r}_n, \Theta_3) = \sum_{k=1}^K \pi_k \prod_{l=1}^d \left[\mathcal{S}_t(y_{nl}|\boldsymbol{w}_{kl}^T \mathbf{R}_{nk}\boldsymbol{x}_{nk} + \mu_{kl}, \sigma_{kl}, v_{kl})^{\phi_{nl}} \mathcal{S}_t(y_{nl}|\boldsymbol{w}_{kl}^T \mathbf{R}_{nk}\boldsymbol{x}_{nk} + \mu_{0l}, \sigma_{0l}, v_{0l})^{1-\phi_{nl}}\right], \tag{9}$$

where $\mathbf{R}_{nk} = \mathrm{diag}(\boldsymbol{r}_{nk})$ and we denote $\boldsymbol{r}_n = \{\boldsymbol{r}_{nk}\}_{k=1}^K$. When r_{nkj}'s all equal zero, the model reduces to the Student's t mixtures of model (1).

The prior distribution of $\boldsymbol{r}_{nk}$ is given by

$$p(\boldsymbol{r}_{nk}|\boldsymbol{\rho}_k) = \prod_{j=1}^{p_k} p(r_{nkj}|\rho_{kj}) = \prod_{j=1}^{p_k} \rho_{kj}{}^{r_{nkj}}(1-\rho_{kj})^{1-r_{nkj}}, \tag{10}$$

where we have assumed prior independence between the entries of $\boldsymbol{r}_{nk}$. Denote $\boldsymbol{\rho}_k = \{\rho_{kj}\}_{j=1}^{p_k}$ and $\boldsymbol{\rho} = \{\boldsymbol{\rho}_k\}_{k=1}^K$. In accordance with the concept of feature saliency, we call ρ_{kj} the factor activity. It is the probability that the jth factor in class k is active. The problem of finding latent dimensions then can be cast as a parameter-estimation problem, i.e., the estimation of $\boldsymbol{\rho}$.

Our modeling of $\boldsymbol{r}_{nk}, k = 1, 2, \ldots, K$ to select the active factors in each class is inspired by the normal-zero model proposed in [29], which introduces the indicators to select automatically the important covariates in linear regression. The difference is that we have defined the indicators as latent variables for each individual, while the normal-zero model introduces the indicators as model parameters. In our model, the parameters $\boldsymbol{\rho}$ that define the Bernoulli distributions of the indicators are the key quantities for model selection and will be inferred under a Bayesian inference framework, while following the normal-zero model $\boldsymbol{\rho}$ are treated as hyper-parameters and typically need to be specified.

Note that the conditional probability of (9) defines a mixture of robust factor analyzers. The factor model for class k can be written as

$$\boldsymbol{y}_n = \mathbf{W}_k \mathbf{R}_{nk} \boldsymbol{x}_{nk} + \boldsymbol{\Phi}_n \boldsymbol{\mu}_k + (\mathbf{I} - \boldsymbol{\Phi}_n)\boldsymbol{\mu}_0 + \boldsymbol{\varepsilon}_n, \tag{11}$$

where $\mathbf{W}_k = [\boldsymbol{w}_{k1}, \boldsymbol{w}_{k2}, \ldots, \boldsymbol{w}_{kd}]^T$ is the $d \times p_k$ factor loading matrix. Denote $\boldsymbol{\mu}_k = (\mu_{k1}, \mu_{k2}, \ldots, \mu_{kd})^T$, $\boldsymbol{\mu}_0 = (\mu_{01}, \mu_{02}, \ldots, \mu_{0d})^T$ and $\boldsymbol{\Phi}_n = \mathrm{diag}(\boldsymbol{\phi}_n)$. The latent factors $\boldsymbol{x}_{nk}$ follow the distribution of $\mathcal{N}(\mathbf{0}, \mathbf{I}_{p_k})$, where $\mathbf{I}_{p_k}$ is the identity matrix of order p_k. The distributions for $\boldsymbol{\phi}_n$ and $\boldsymbol{r}_{nk}$ are defined in (2) and (10), separately. $\boldsymbol{\varepsilon}_n = (\varepsilon_{n1}, \varepsilon_{n2}, \ldots, \varepsilon_{nd})^T$, where ε_{nl}'s are mutually independent given z_n and

$$p(\varepsilon_{nl}|\phi_{nl}, z_n = k, \boldsymbol{\sigma}, \boldsymbol{v}) = \mathcal{S}_t(\varepsilon_{nl}|0, \sigma_{kl}, v_{kl})^{\phi_{nl}} \mathcal{S}_t(\varepsilon_{nl}|0, \sigma_{0l}, v_{0l})^{1-\phi_{nl}}. \tag{12}$$

By introducing the latent variable u_{nl} distributed according to (6), we have

$$p(\varepsilon_{nl}|u_{nl}, \phi_{nl}, z_n = k, \boldsymbol{\sigma}) = \mathcal{N}(\varepsilon_{nl}|0, \sigma_{kl} u_{nl})^{\phi_{nl}} \mathcal{N}(\varepsilon_{nl}|0, \sigma_{0l} u_{nl})^{1-\phi_{nl}}. \tag{13}$$

The complete-data likelihood $p(\boldsymbol{y}_n, \boldsymbol{u}_n, \boldsymbol{\phi}_n, \boldsymbol{x}_n, \boldsymbol{r}_n, z_n|\Theta)$ for the proposed hierarchical latent variable model, where $\Theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, v, \boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{\rho}\}$ can be factorized as

$$p(\boldsymbol{y}_n, \boldsymbol{u}_n, \boldsymbol{\phi}_n, \boldsymbol{x}_n, \boldsymbol{r}_n, z_n|\Theta) = \prod_{k=1}^{K} \Big[\pi_k p(\boldsymbol{y}_n|\boldsymbol{u}_n, \boldsymbol{\phi}_n, \boldsymbol{x}_{nk}, \boldsymbol{r}_{nk}, z_n = k, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{w}) \\ \times p(\boldsymbol{u}_n|\boldsymbol{\phi}_n, z_n = k, v) p(\boldsymbol{\phi}_n|\boldsymbol{\beta}) p(\boldsymbol{x}_{nk}) p(\boldsymbol{r}_{nk}|\boldsymbol{\rho}_k)\Big]^{\delta_{z_n,k}}, \quad (14)$$

where

$$p(\boldsymbol{y}_n|\boldsymbol{u}_n, \boldsymbol{\phi}_n, \boldsymbol{x}_{nk}, \boldsymbol{r}_{nk}, z_n = k, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{w}) = \prod_{l=1}^{d} p(y_{nl}|u_{nl}, \phi_{nl}, \boldsymbol{x}_{nk}, \boldsymbol{r}_{nk}, z_n = k, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{w}) \\ = \prod_{l=1}^{d} \Big[\mathcal{N}(y_{nl}|\boldsymbol{w}_{kl}^T \mathbf{R}_{nk} \boldsymbol{x}_{nk} + \mu_{kl}, \sigma_{kl} u_{nl})^{\phi_{nl}} \mathcal{N}(y_{nl}|\boldsymbol{w}_{kl}^T \mathbf{R}_{nk} \boldsymbol{x}_{nk} + \mu_{0l}, \sigma_{0l} u_{nl})^{1-\phi_{nl}}\Big], \quad (15)$$

corresponding to a modification of conditional probability (5) for the Student's t mixture model.

In the following, we denote the set of latent variables as $\mathcal{H} = \{\boldsymbol{h}_n\}_{n=1}^N$ where $\boldsymbol{h}_n = \{\boldsymbol{u}_n, \boldsymbol{\phi}_n, \boldsymbol{x}_n, \boldsymbol{r}_n, z_n\}$. Then, the complete-data likelihood for the whole dataset can be written as

$$p(\mathbf{Y}, \mathcal{H}|\Theta) = \prod_{n=1}^{N} p(\boldsymbol{y}_n, \boldsymbol{h}_n|\Theta). \quad (16)$$

Full Bayesian treatment to the latent variable model requires specification of the prior distributions associated with the model parameters. We assume that

$$p(\Theta) = p(\boldsymbol{\pi}) p(\boldsymbol{\mu}) p(\boldsymbol{\sigma}) p(\boldsymbol{\beta}) p(\boldsymbol{w}) p(\boldsymbol{\rho}), \quad (17)$$

and

$$\begin{aligned} p(\boldsymbol{\pi}) &= \mathcal{D}ir(\boldsymbol{\pi}|\boldsymbol{\alpha}_0), \\ p(\boldsymbol{\mu}) &= \prod_{l=1}^{d}\Big[p(\mu_{0l})\prod_{k=1}^{K} p(\mu_{kl})\Big] = \prod_{l=1}^{d}\Big[\mathcal{N}(\mu_{0l}|s_{0l}, \lambda_0)\prod_{k=1}^{K}\mathcal{N}(\mu_{kl}|s_{0l}, \lambda_0)\Big], \\ p(\boldsymbol{\sigma}) &= \prod_{l=1}^{d}\Big[p(\sigma_{0l})\prod_{k=1}^{K} p(\sigma_{kl})\Big] = \prod_{l=1}^{d}\Big[\mathcal{G}\Big(\sigma_{0l}\Big|\frac{\eta_0}{2}, \frac{\xi_0}{2}\Big)\prod_{k=1}^{K}\mathcal{G}\Big(\sigma_{kl}\Big|\frac{\eta_0}{2}, \frac{\xi_0}{2}\Big)\Big], \\ p(\boldsymbol{\beta}) &= \prod_{l=1}^{d}\mathcal{B}eta(\beta_l|\kappa_1, \kappa_2), \\ p(\boldsymbol{w}) &= \prod_{k=1}^{K}\prod_{l=1}^{d} p(\boldsymbol{w}_{kl}) = \prod_{k=1}^{K}\prod_{l=1}^{d}\mathcal{N}(\boldsymbol{w}_{kl}|\mathbf{0}, m_0\mathbf{I}_{p_k}), \\ p(\boldsymbol{\rho}) &= \prod_{k=1}^{K}\prod_{j=1}^{p_k}\mathcal{B}eta(\rho_{kj}|\tau_1, \tau_2), \end{aligned} \quad (18)$$

where $\mathcal{B}eta(\beta|a, b)$ represents the Beta density function

$$\mathcal{B}eta(\beta|a, b) = \frac{\beta^{a-1}(1-\beta)^{b-1}}{\mathcal{B}(a, b)}, \quad (19)$$

and

$$\mathcal{D}ir(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}{\pi_k}^{\alpha_k - 1}, \quad (20)$$

is the Dirichlet density. In the above specifications, the conjugate priors are used. The parameters in the priors, including κ_1, κ_2, τ_1, τ_2, m_0, λ_0, η_0, ξ_0, $\boldsymbol{s}_0$ and $\boldsymbol{\alpha}_0$ where $\boldsymbol{s}_0 = \{s_{0l}\}_{l=1}^d$

and $\boldsymbol{\alpha}_0 = (\alpha_{01}, \alpha_{02}, \ldots, \alpha_{0K})^T$, are considered as hyperparameters. It is noticeable that we do not assume any prior for the degrees of freedom v_{0l}'s and v_{kl}'s. Since there are no conjugate priors, we follow the practice in [3,14] to seek for the point estimates for them.

The plate diagram of the proposed hierarchical latent variable model is shown in Figure 1. The arrows in the diagram indicate the dependencies. The original model of Sun and Zhou [14] is depicted in blue.

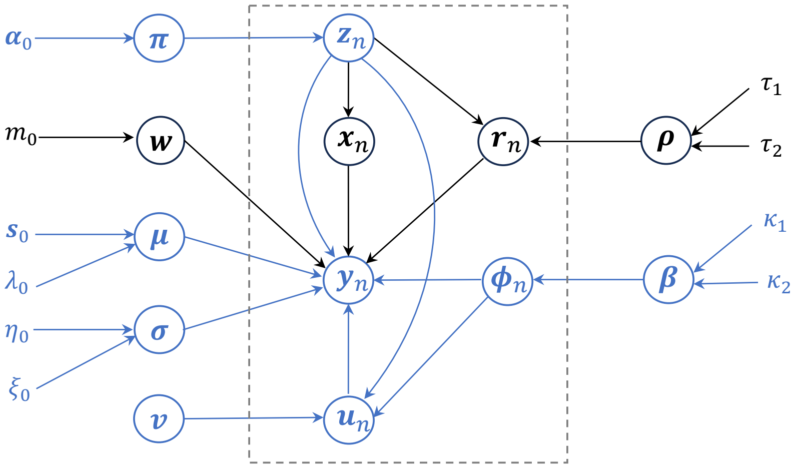

Figure 1. Plate diagram of the proposed hierarchical latent variable model.

4. Inference on the Model

4.1. Brief Introduction to VB Method

To infer from the posterior distribution of the latent variables and the parameters, computation of the evidence $p(\mathbf{Y})$ is required. However, the computation involves integration over the latent variables and the parameters, which is intractable for our model. In this paper, we resort to the VB method for model inference. It is designed to maximize a lower bound of $\log p(\mathbf{Y})$. Assuming posterior independence between the latent variables and the parameters, the evidence lower bound (ELBO) is defined by

$$\mathcal{L}(q(\mathcal{H}), q(\Theta), \mathbf{Y}) = \mathbb{E}_{q(\mathcal{H})q(\Theta)}\left[\log \frac{p(\mathbf{Y}, \mathcal{H}|\Theta)p(\Theta)}{q(\mathcal{H})q(\Theta)}\right] \leq \log p(\mathbf{Y}), \tag{21}$$

where $q(\mathcal{H})$ and $q(\Theta)$ are auxiliary posteriors for the latent variables and the parameters, respectively. A coordinate ascent search method [31] can be applied to iteratively maximize the ELBO. At the tth iteration, it implements the VB expectation (VB-E) step and the VB maximization (VB-M) step as follows:

$$\begin{aligned} \text{VB-E step:} \quad & q^{(t+1)}(\mathcal{H}) = \arg\max_{q(\mathcal{H})} \mathcal{L}(q(\mathcal{H}), q^{(t)}(\Theta), \mathbf{Y}); \\ \text{VB-M step:} \quad & q^{(t+1)}(\Theta) = \arg\max_{q(\Theta)} \mathcal{L}(q^{(t+1)}(\mathcal{H}), q(\Theta), \mathbf{Y}). \end{aligned} \tag{22}$$

4.2. Tree-Like Factorization of the Auxiliary Posterior

In this paper, we apply the tree-like factorization proposed in [3,14] to the auxiliary posterior of the latent variables. The resultant structured VB method can be viewed as a partially collapsed VB [32], which can reach a tighter lower bound for $\log p(\mathbf{Y})$ than the mean-filed approximated VB method [13].

As the observations are mutually independent, $q(\mathcal{H})$ has the form

$$q(\mathcal{H}) = \prod_{n=1}^{N} q(\boldsymbol{h}_n). \tag{23}$$

Tree-like factorization assumes that the auxiliary posterior $q(\boldsymbol{h}_n)$ can be factorized as

$$q(\boldsymbol{h}_n) = \prod_{k=1}^{K} \left\{ q(\boldsymbol{u}_n|\boldsymbol{\phi}_n, z_n = k) q(\boldsymbol{\phi}_n) q(\boldsymbol{x}_{nk}|\boldsymbol{r}_{nk}) q(\boldsymbol{r}_{nk}) q(z_n = k) \right\}^{\delta_{z_n,k}}. \tag{24}$$

As entries of the noise term in the local factor model are assumed to be mutually independent, $q(\boldsymbol{h}_n)$ can be further factorized as

$$\begin{aligned} q(\boldsymbol{h}_n) = \prod_{k=1}^{K} \Bigg\{ & \prod_{l=1}^{d} \Big[\big(q(u_{nl}|\phi_{nl} = 1, z_n = k) q(\phi_{nl} = 1)\big)^{\phi_{nl}} \big(q(u_{nl}|\phi_{nl} = 0) q(\phi_{nl} = 0)\big)^{1-\phi_{nl}} \Big] \\ & \times q(\boldsymbol{x}_{nk}|\boldsymbol{r}_{nk}) q(\boldsymbol{r}_{nk}) q(z_n = k) \Bigg\}^{\delta_{z_n,k}}. \end{aligned} \tag{25}$$

Different from [3,14], we do not keep the posterior dependence of ϕ_{nl} on z_n and when $\phi_{nl} = 0$ the auxiliary posterior of u_{nl} is assumed to be independent of z_n, though the closed forms of the posteriors are available when retaining the dependencies. We found that the above specifications lead to more robust inference results. As in [29,33], we assume a full factorization for $q(\boldsymbol{r}_{nk})$, i.e.,

$$q(\boldsymbol{r}_{nk}) = \prod_{j=1}^{p_k} q(r_{nkj}) = \prod_{j=1}^{p_k} q(r_{nkj} = 1)^{r_{nkj}} q(r_{nkj} = 0)^{1-r_{nkj}}. \tag{26}$$

Additionally, the auxiliary posterior $q(\Theta)$ is assumed to be its full factorized form

$$\begin{aligned} q(\Theta) &= q(\boldsymbol{\pi}) q(\boldsymbol{\mu}) q(\boldsymbol{\sigma}) q(\boldsymbol{\beta}) q(\boldsymbol{w}) q(\boldsymbol{\rho}) \\ &= q(\boldsymbol{\pi}) \cdot \prod_{l=1}^{d} \left[q(\mu_{0l}) \prod_{k=1}^{K} q(\mu_{kl}) \right] \cdot \prod_{l=1}^{d} \left[q(\sigma_{0l}) \prod_{k=1}^{K} q(\sigma_{kl}) \right] \cdot \prod_{l=1}^{d} q(\beta_l) \cdot \prod_{k=1}^{K} \prod_{l=1}^{d} q(\boldsymbol{w}_{kl}) \cdot \prod_{k=1}^{K} \prod_{j=1}^{p_k} q(\rho_{kj}). \end{aligned} \tag{27}$$

For ease of exposition, we use n, l, j and k in the following to denote the index of the individual, the feature, the latent factor and the class, respectively. We omit the iteration indexes (t) and $(t+1)$ and without loss of generosity deliver the update during one iteration of the algorithm. We use $\langle \cdot \rangle$ to denote the expectation operation with respect to the current auxiliary posteriors.

4.3. Auxiliary Posteriors of the Latent Variables: VB-E Step

The VB-E step updates the auxiliary posterior $q(\boldsymbol{h}_n)$ of the latent variables following the factorizations of (25) and (26).

(i) $q(u_{nl}|\phi_{nl}, z_n)$: Through some mathematical manipulations (see the Supplementary Materials for the details), we obtain

$$\begin{aligned} q(u_{nl}|\phi_{nl} = 1, z_n = k) &= \mathcal{G}(u_{nl}|\hat{a}_{kl}, \hat{b}_{nl}^{k}), \\ q(u_{nl}|\phi_{nl} = 0) &= \mathcal{G}(u_{nl}|\hat{a}_{0l}, \hat{b}_{nl}^{0}), \end{aligned} \tag{28}$$

where

$$\begin{aligned} \hat{a}_{kl} &= \frac{v_{kl} + 1}{2}, \hat{b}_{nl}^{k} = \frac{v_{kl} + \langle \sigma_{kl} \rangle \langle (\tilde{y}_{nl}^{k} - \mu_{kl})^2 \rangle}{2}, \\ \hat{a}_{0l} &= \frac{v_{0l} + 1}{2}, \hat{b}_{nl}^{0} = \frac{v_{0l} + \langle \sigma_{0l} \rangle \sum_k \langle \delta_{z_n,k} \rangle \langle (\tilde{y}_{nl}^{k} - \mu_{0l})^2 \rangle}{2}, \end{aligned} \tag{29}$$

and $\tilde{y}_{nl}^{k} = y_{nl} - \boldsymbol{w}_{kl}^{T} \mathbf{R}_{nk} \boldsymbol{x}_{nk}$. Note that

$$
\begin{aligned}
\langle(\tilde{y}^k_{nl}-\mu_{kl})^2\rangle =& \big(y_{nl}-\langle\mu_{kl}\rangle\big)^2+\hat{\lambda}_{kl}^{-1}-2\big(y_{nl}-\langle\mu_{kl}\rangle\big)\langle\boldsymbol{w}_{kl}\rangle^T\langle\mathbf{R}_{nk}\boldsymbol{x}_{nk}\rangle\\
&+\mathrm{tr}\big(\langle\boldsymbol{w}_{kl}\boldsymbol{w}_{kl}{}^T\rangle\langle\boldsymbol{x}_{nk}\boldsymbol{x}_{nk}{}^T\odot\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T\rangle\big),\\
\langle(\tilde{y}^k_{nl}-\mu_{0l})^2\rangle =& \big(y_{nl}-\langle\mu_{0l}\rangle\big)^2+\hat{\lambda}_{0l}^{-1}-2\big(y_{nl}-\langle\mu_{0l}\rangle\big)\langle\boldsymbol{w}_{kl}\rangle^T\langle\mathbf{R}_{nk}\boldsymbol{x}_{nk}\rangle\\
&+\mathrm{tr}\big(\langle\boldsymbol{w}_{kl}\boldsymbol{w}_{kl}{}^T\rangle\langle\boldsymbol{x}_{nk}\boldsymbol{x}_{nk}{}^T\odot\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T\rangle\big),
\end{aligned}
\tag{30}
$$

where $\hat{\lambda}_{kl}$ is the precision of posterior $q(\mu_{kl})$ and $\hat{\lambda}_{0l}$ is the precision of $q(\mu_{0l})$. We denote the trace operator as $\mathrm{tr}(\cdot)$ and the Hadamard product operator between two matrices as $\odot$.

In the sequel, we use $\langle\cdot\rangle_k^1$ and $\langle\cdot\rangle^0$ to distinguish between the expectations regarding $q(u_{nl}|\phi_{nl}=1, z_n=k)$ and $q(u_{nl}|\phi_{nl}=0)$. As with the property of Gamma distribution, we obtain

$$
\begin{aligned}
\langle u_{nl}\rangle_k^1 &= \frac{\hat{a}_{kl}}{\hat{b}^k_{nl}}, \quad \langle\log u_{nl}\rangle_k^1=\psi(\hat{a}_{kl})-\log(\hat{b}^k_{nl}),\\
\langle u_{nl}\rangle^0 &= \frac{\hat{a}_{0l}}{\hat{b}^0_{nl}}, \quad \langle\log u_{nl}\rangle^0=\psi(\hat{a}_{0l})-\log(\hat{b}^0_{nl}),
\end{aligned}
\tag{31}
$$

where $\psi(\cdot)$ is the digamma function.

(ii) $q(\phi_{nl})$: Define

$$
\begin{aligned}
\overline{q}(\phi_{nl}=1)&=\exp\left\{\sum_k\langle\delta_{z_n,k}\rangle\left[\frac{1}{2}\langle\log\sigma_{kl}\rangle+\frac{v_{kl}}{2}\log\frac{v_{kl}}{2}-\log\Gamma\Big(\frac{v_{kl}}{2}\Big)-\hat{a}_{kl}\log\hat{b}^k_{nl}+\log\Gamma(\hat{a}_{kl})\right]+\langle\log\beta_l\rangle\right\},\\
\overline{q}(\phi_{nl}=0)&=\exp\left\{\frac{1}{2}\langle\log\sigma_{0l}\rangle+\frac{v_{0l}}{2}\log\frac{v_{0l}}{2}-\log\Gamma\Big(\frac{v_{0l}}{2}\Big)-\hat{a}_{0l}\log\hat{b}^0_{nl}+\log\Gamma(\hat{a}_{0l})+\langle\log(1-\beta_l)\rangle\right\}.
\end{aligned}
\tag{32}
$$

Then, $q(\phi_{nl})$ can be obtained by

$$
q(\phi_{nl}=1)=\frac{\overline{q}(\phi_{nl}=1)}{\overline{q}(\phi_{nl}=1)+\overline{q}(\phi_{nl}=0)},
\tag{33}
$$

and $q(\phi_{nl}=0)=1-q(\phi_{nl}=1)$. Denote $\langle\phi_{nl}\rangle=q(\phi_{nl}=1)$ and $\langle 1-\phi_{nl}\rangle=q(\phi_{nl}=0)$.

(iii) $q(\boldsymbol{x}_{nk}|\boldsymbol{r}_{nk})$: The posterior $q(\boldsymbol{x}_{nk}|\boldsymbol{r}_{nk})$ is multivariate Gaussian with precision matrix and mean vector as

$$
\hat{\mathbf{C}}^k_n(\boldsymbol{r}_{nk})=\mathbf{I}+\mathbf{A}_{nk}\odot\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T, \quad \hat{\boldsymbol{f}}^k_n(\boldsymbol{r}_{nk})=\hat{\mathbf{C}}^k_n(\boldsymbol{r}_{nk})^{-1}\mathbf{R}_{nk}\boldsymbol{t}_{nk},
\tag{34}
$$

where

$$
\begin{aligned}
\mathbf{A}_{nk}&=\sum_l\Big[\langle\phi_{nl}\rangle\langle\sigma_{kl}\rangle\langle u_{nl}\rangle_k^1+\langle 1-\phi_{nl}\rangle\langle\sigma_{0l}\rangle\langle u_{nl}\rangle^0\Big]\langle\boldsymbol{w}_{kl}\boldsymbol{w}_{kl}^T\rangle,\\
\boldsymbol{t}_{nk}&=\sum_l\Big[\langle\phi_{nl}\rangle\langle\sigma_{kl}\rangle\langle u_{nl}\rangle_k^1\big(y_{nl}-\langle\mu_{kl}\rangle\big)+\langle 1-\phi_{nl}\rangle\langle\sigma_{0l}\rangle\langle u_{nl}\rangle^0\big(y_{nl}-\langle\mu_{0l}\rangle\big)\Big]\langle\boldsymbol{w}_{kl}\rangle.
\end{aligned}
\tag{35}
$$

(iv) $q(\boldsymbol{r}_{nk})$: The posterior $q(r_{nkj})$ can be obtained by

$$
q(r_{nkj}=1)=\frac{\overline{q}(r_{nkj}=1)}{\overline{q}(r_{nkj}=1)+\overline{q}(r_{nkj}=0)},
\tag{36}
$$

and $q(r_{nkj}=0)=1-q(r_{nkj}=1)$, where

$$\begin{aligned}\bar{q}(r_{nkj}=c)=\exp\bigg[&-\frac{1}{2}\langle\log|\mathbf{I}+\mathbf{A}_{nk}\odot\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T|\rangle\\&+\frac{1}{2}\mathrm{tr}\langle(\mathbf{I}+\mathbf{A}_{nk}\odot\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T)^{-1}(\mathbf{R}_{nk}\boldsymbol{t}_{nk})(\mathbf{R}_{nk}\boldsymbol{t}_{nk})^T\rangle\\&+c\langle\log\rho_{kj}\rangle+(1-c)\langle\log(1-\rho_{kj})\rangle\bigg],\end{aligned}\tag{37}$$

with $c\in\{0,1\}$. The expectations in (37) are taken by fixing $r_{nkj}=c$. Denote $\langle r_{nkj}\rangle=q(r_{nkj}=1)$ and $\langle 1-r_{nkj}\rangle=q(r_{nkj}=0)$.

When posterior independence is assumed between $\boldsymbol{x}_{nk}$ and $\boldsymbol{r}_{nk}$ or $\boldsymbol{x}_{nk}$ is observable as in the regression models of [29,33], $q(r_{nkj})$ can be derived analytically and the expectations in (30) regarding $q(\boldsymbol{x}_{nk},\boldsymbol{r}_{nk})$ can be obtained in closed form using the results:

$$\begin{aligned}\langle\mathbf{R}_{nk}\rangle&=\mathrm{diag}(\langle\boldsymbol{r}_{nk}\rangle),\\\langle\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T\rangle&=\langle\mathbf{R}_{nk}\rangle\odot\langle\mathbf{I}-\mathbf{R}_{nk}\rangle+\langle\boldsymbol{r}_{nk}\rangle\langle\boldsymbol{r}_{nk}\rangle^T,\\\langle\boldsymbol{x}_{nk}\rangle&=\big(\mathbf{I}+\mathbf{A}_{nk}\odot\langle\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T\rangle\big)^{-1}\langle\mathbf{R}_{nk}\rangle\boldsymbol{t}_{nk},\\\langle\boldsymbol{x}_{nk}\boldsymbol{x}_{nk}^T\rangle&=\big(\mathbf{I}+\mathbf{A}_{nk}\odot\langle\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T\rangle\big)^{-1}+\langle\boldsymbol{x}_{nk}\rangle\langle\boldsymbol{x}_{nk}\rangle^T.\end{aligned}\tag{38}$$

However, the computation in high-dimensional cases is obstructed as it involves multiplication and inversion of large-scale matrices.

The sparse property of the indicator vector $\boldsymbol{r}_{nk}$ motivates us to resort to a "drop-out" sampling scheme [30], where the conditioning of $\boldsymbol{x}_{nk}$ on $\boldsymbol{r}_{nk}$ does not influence the efficiency of the algorithm. Specifically, we keep a random sample $\hat{\boldsymbol{r}}_{nk}$ from $q(\boldsymbol{r}_{nk})$ at each iteration of the algorithm and use it as an imputation for $\boldsymbol{r}_{nk}$ to update the remaining auxiliary posteriors. During this process, the connections to the latent factor with smaller $q(r_{nkj}=1)$ have higher chance of drop out. Simplification of the computation can be realized, for example, in Equation (30),

$$\begin{aligned}\langle\boldsymbol{w}_{kl}\rangle^T\langle\mathbf{R}_{nk}\boldsymbol{x}_{nk}\rangle&=\big(\hat{\mathbf{R}}_{nk}\langle\boldsymbol{w}_{kl}\rangle\big)^T\big(\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)^{-1}\hat{\mathbf{R}}_{nk}\boldsymbol{t}_{nk},\\\langle\boldsymbol{w}_{kl}\boldsymbol{w}_{kl}{}^T\rangle\langle\boldsymbol{x}_{nk}\boldsymbol{x}_{nk}^T\odot\boldsymbol{r}_{nk}\boldsymbol{r}_{nk}^T\rangle&=\big(\langle\boldsymbol{w}_{kl}\boldsymbol{w}_{kl}^T\rangle\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)\Big[\big(\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)^{-1}\\&\quad+\big(\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)^{-1}\big(\hat{\mathbf{R}}_{nk}\boldsymbol{t}_{nk}\big)\big(\hat{\mathbf{R}}_{nk}\boldsymbol{t}_{nk}\big)^T\big(\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)^{-1}\Big],\end{aligned}\tag{39}$$

where the latent dimensions to be tackled are reduced due to the sparse property of the random sample $\hat{\boldsymbol{r}}_{nk}$. For the multiplication and inversion computations, only the entries of vector or matrix corresponding to $\hat{r}_{nkj}\neq 0$ need to be involved.

To obtain a random sample from $q(\boldsymbol{r}_{nk})$, we update the entries of $\hat{\boldsymbol{r}}_{nk}$ one by one through a single turn of Gibbs sampling, where the sampling probability for $\hat{r}_{nkj}$ has the form in (37) but with the expectation replaced by the current imputation of $\hat{\boldsymbol{r}}_{nk,-j}$.

(v) $q(z_n)$: To update $q(z_n=k)$, we define the quantity

$$\begin{aligned}\bar{q}(z_n=k)=\exp\Bigg\{&\sum_l\langle\phi_{nl}\rangle\bigg[\frac{1}{2}\langle\log\sigma_{kl}\rangle+\frac{v_{kl}}{2}\log\frac{v_{kl}}{2}-\log\Gamma\Big(\frac{v_{kl}}{2}\Big)-\hat{a}_{kl}\log\hat{b}_{nl}^k+\log\Gamma(\hat{a}_{kl})\bigg]\\&-\frac{1}{2}\sum_l\langle 1-\phi_{nl}\rangle\langle\sigma_{0l}\rangle\langle u_{nl}\rangle^0\langle(\tilde{y}_{nl}^k-\mu_{0l})^2\rangle-\frac{1}{2}\log\big|\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big|\\&-\frac{1}{2}\mathrm{tr}\Big[\big(\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)^{-1}+\big(\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)^{-1}\big(\hat{\mathbf{R}}_{nk}\boldsymbol{t}_{nk}\big)\big(\hat{\mathbf{R}}_{nk}\boldsymbol{t}_{nk}\big)^T\big(\mathbf{I}+\mathbf{A}_{nk}\odot\hat{\boldsymbol{r}}_{nk}\hat{\boldsymbol{r}}_{nk}^T\big)^{-1}\Big]\\&+\sum_j\Big[\langle r_{nkj}\rangle\big(\langle\log\rho_{kj}\rangle-\log\langle r_{nkj}\rangle\big)+\langle 1-r_{nkj}\rangle\big(\langle\log(1-\rho_{kj})\rangle-\log\langle 1-r_{nkj}\rangle\big)\Big]+\langle\log\pi_k\rangle\Bigg\}.\end{aligned}\tag{40}$$

Then,

$$q(z_n = k) = \frac{\overline{q}(z_n = k)}{\sum_{k'} \overline{q}(z_n = k')}. \tag{41}$$

4.4. Auxiliary Posteriors of the Parameters: VB-M Step

The VB-M step updates the posterior $q(\Theta)$ for the parameters following the factorization in (27). Through mathematical manipulation (see the Supplementary Materials for the details), we have

$$\begin{aligned} q(\boldsymbol{\pi}) &= \mathcal{D}ir(\boldsymbol{\pi}|\hat{\boldsymbol{\alpha}}), \\ q(\beta_l) &= \mathcal{B}eta(\beta_l|\hat{\kappa}_{1l}, \hat{\kappa}_{2l}), \\ q(\rho_{kj}) &= \mathcal{B}eta(\rho_{kj}|\hat{\tau}_{1kj}, \hat{\tau}_{2kj}), \end{aligned} \tag{42}$$

where

$$\begin{aligned} &\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_K)^T, \quad \hat{\alpha}_k = \alpha_{0k} + \sum_n \langle \delta_{z_n,k} \rangle, \\ &\hat{\kappa}_{1l} = \kappa_1 + \sum_n \langle \phi_{nl} \rangle, \quad \hat{\kappa}_{2l} = \kappa_2 + \sum_n \langle 1 - \phi_{nl} \rangle, \\ &\hat{\tau}_{1kj} = \tau_1 + \sum_n \langle \delta_{z_n,k} \rangle \langle r_{nkj} \rangle, \quad \hat{\tau}_{2kj} = \tau_2 + \sum_n \langle \delta_{z_n,k} \rangle \langle 1 - r_{nkj} \rangle. \end{aligned} \tag{43}$$

The posterior $q(\boldsymbol{w}_{kl})$ is given by

$$q(\boldsymbol{w}_{kl}) = \mathcal{N}(\boldsymbol{w}_{kl}|\hat{\boldsymbol{m}}_{kl}, \hat{\mathbf{M}}_{kl}), \tag{44}$$

where

$$\begin{aligned} \hat{\mathbf{M}}_{kl} &= m_0 \mathbf{I} + \sum_n \langle \delta_{z_n,k} \rangle \Big(\langle \sigma_{kl} \rangle \langle \phi_{nl} \rangle \langle u_{nl} \rangle_k^1 + \langle \sigma_{0l} \rangle \langle 1 - \phi_{nl} \rangle \langle u_{nl} \rangle^0 \Big) \langle \boldsymbol{x}_{nk} \boldsymbol{x}_{nk}^T \odot \boldsymbol{r}_{nk} \boldsymbol{r}_{nk}^T \rangle, \\ \hat{\boldsymbol{m}}_{kl} &= \hat{\mathbf{M}}_{kl}^{-1} \sum_n \langle \delta_{z_n,k} \rangle \Big[\langle \sigma_{kl} \rangle \langle \phi_{nl} \rangle \langle u_{nl} \rangle_k^1 (y_{nl} - \langle \mu_{kl} \rangle) + \langle \sigma_{0l} \rangle \langle 1 - \phi_{nl} \rangle \langle u_{nl} \rangle^0 (y_{nl} - \langle \mu_{0l} \rangle) \Big] \langle \mathbf{R}_{nk} \boldsymbol{x}_{nk} \rangle. \end{aligned} \tag{45}$$

The posteriors $q(\mu_{0l})$ and $q(\mu_{kl})$ are given by

$$q(\mu_{0l}) = \mathcal{N}(\mu_{0l}|\hat{s}_{0l}, \hat{\lambda}_{0l}), \quad q(\mu_{kl}) = \mathcal{N}(\mu_{kl}|\hat{s}_{kl}, \hat{\lambda}_{kl}), \tag{46}$$

where

$$\begin{aligned} \hat{\lambda}_{0l} &= \lambda_0 + \langle \sigma_{0l} \rangle \sum_n \langle 1 - \phi_{nl} \rangle \langle u_{nl} \rangle^0, \\ \hat{s}_{0l} &= \hat{\lambda}_{0l}^{-1} \Big(\lambda_0 s_{0l} + \langle \sigma_{0l} \rangle \sum_{n,k} \langle \delta_{z_n,k} \rangle \langle 1 - \phi_{nl} \rangle \langle u_{nl} \rangle^0 \langle \tilde{y}_{nl}^k \rangle \Big), \\ \hat{\lambda}_{kl} &= \lambda_0 + \langle \sigma_{kl} \rangle \sum_n \langle \delta_{z_n,k} \rangle \langle \phi_{nl} \rangle \langle u_{nl} \rangle_k^1, \\ \hat{s}_{kl} &= \hat{\lambda}_{kl}^{-1} \Big(\lambda_0 s_{0l} + \langle \sigma_{kl} \rangle \sum_n \langle \delta_{z_n,k} \rangle \langle \phi_{nl} \rangle \langle u_{nl} \rangle_k^1 \langle \tilde{y}_{nl}^k \rangle \Big). \end{aligned} \tag{47}$$

In addition, the posteriors $q(\sigma_{0l})$ and $q(\sigma_{kl})$ are updated as

$$q(\sigma_{0l}) = \mathcal{G}(\sigma_{0l}|\frac{\hat{\eta}_{0l}}{2}, \frac{\hat{\xi}_{0l}}{2}), \quad q(\sigma_{kl}) = \mathcal{G}(\sigma_{kl}|\frac{\hat{\eta}_{kl}}{2}, \frac{\hat{\xi}_{kl}}{2}), \tag{48}$$

where

$$\begin{aligned}
\hat{\eta}_{0l} &= \eta_0 + \sum_n \langle 1 - \phi_{nl} \rangle, \\
\hat{\xi}_{0l} &= \xi_0 + \sum_{n,k} \langle \delta_{z_n,k} \rangle \langle 1 - \phi_{nl} \rangle \langle u_{nl} \rangle^0 \langle (\tilde{y}^k_{nl} - \mu_{0l})^2 \rangle, \\
\hat{\eta}_{kl} &= \eta_0 + \sum_n \langle \delta_{z_n,k} \rangle \langle \phi_{nl} \rangle, \\
\hat{\xi}_{kl} &= \xi_0 + \sum_n \langle \delta_{z_n,k} \rangle \langle \phi_{nl} \rangle \langle u_{nl} \rangle^1_k \langle (\tilde{y}^k_{nl} - \mu_{kl})^2 \rangle.
\end{aligned} \tag{49}$$

The degrees of freedom v_{0l} can be obtained by solving the nonlinear equation

$$\sum_n \langle 1 - \phi_{nl} \rangle \left[1 + \log\left(\frac{v_{0l}}{2}\right) - \psi\left(\frac{v_{0l}}{2}\right) + \langle \log u_{nl} \rangle^0 - \langle u_{nl} \rangle^0 \right] = 0. \tag{50}$$

Similarly, v_{kl} can be obtained by solving

$$\sum_n \langle \delta_{z_n,k} \rangle \langle \phi_{nl} \rangle \left[1 + \log\left(\frac{v_{kl}}{2}\right) - \psi\left(\frac{v_{kl}}{2}\right) + \langle \log u_{nl} \rangle^1_k - \langle u_{nl} \rangle^1_k \right] = 0. \tag{51}$$

4.5. Algorithm

The developed structured VB algorithm is summarized in Algorithm 1. The optimization process can be monitored via the ELBO (21). The computation of the ELBO is detailed in Appendix A.

Algorithm 1 Proposed Structured VB Algorithm for Robust Clustering and Model Selection

```
Require: training data $\boldsymbol{y}_n$, $1 \le n \le N$, the number of clusters $K$;
Ensure: the response probabilities, the centroids, the saliency of features, the factor loading
    matrices, the activity of factors;
while the evidence lower bound $\mathcal{L}$ increases more than $\epsilon$ and the number of iteration is
    less than $IterMax$ do
    VB-E step
    Update $q(u_{nl}|\phi_{nl}, z_n)$ according to (28) for $1 \le l \le d$ and $1 \le n \le N$;
    Update $q(\phi_{nl})$ according to (33) for $1 \le l \le d$ and $1 \le n \le N$;
    Update $q(\boldsymbol{x}_{nk}|\boldsymbol{r}_{nk})$ according to (34) for $1 \le k \le K$ and $1 \le n \le N$;
    Update $q(\boldsymbol{r}_{nk})$ according to (36) for $1 \le k \le K$ and $1 \le n \le N$: run a single turn of
    Gibbs sampling for a sample from $q(\boldsymbol{r}_{nk})$;
    Update $q(z_n)$ according to (41) for $1 \le n \le N$;
    VB-M step
    Update $q(\boldsymbol{\pi})$, $q(\beta_l)$ and $q(\rho_{kj})$ according to (42) for $1 \le l \le d$, $1 \le j \le p_k$ and
    $1 \le k \le K$;
    Update $q(\boldsymbol{w}_{kl})$ according to (44) for $1 \le l \le d$ and $1 \le k \le K$;
    Update $q(\mu_{0l})$ and $q(\mu_{kl})$ according to (46) for $1 \le l \le d$ and $1 \le k \le K$;
    Update $q(\sigma_{0l})$ and $q(\sigma_{kl})$ according to (48) for $1 \le l \le d$ and $1 \le k \le K$;
    Update $v_{0l}$ and $v_{kl}$ according to (50) and (51) for $1 \le l \le d$ and $1 \le k \le K$;
end while
```

We apply K-mean clustering for initialization of the VB algorithm and initialize a large p ($p < \min(d, N)$) for the latent dimensions of the K local factor models. At each iteration, we randomize the updating order of $\hat{r}_{nkj}$'s in the Gibbs sampling step to avoid co-adaptation. To further accelerate the algorithm, we make the number of factors adaptive. The empirical estimator of factor activity, i.e.,

$$\hat{\rho}_{kj} = \frac{\sum_n \langle \delta_{z_n,k} \rangle \langle r_{nkj} \rangle}{\sum_n \langle \delta_{z_n,k} \rangle}, \tag{52}$$

is computed at the end of each iteration. If $\hat{\rho}_{kj} = 0$, then we remove the jth latent factor from the kth local factor model. The pruning is carried out after a burn-in period of the algorithm.

4.6. Interpreting the Model

The expectation of feature saliency β_l can be used to show the informative degree of features after being adjusted by latent factors, which is given by

$$\langle \beta_l \rangle = \frac{\hat{\kappa}_{1l}}{\hat{\kappa}_{1l} + \hat{\kappa}_{2l}}. \tag{53}$$

In addition, the expectation of factor activity ρ_{kj} can be applied to evaluate the explanatory power of latent factors in each class, which can be obtained as

$$\langle \rho_{kj} \rangle = \frac{\hat{\tau}_{1kj}}{\hat{\tau}_{1kj} + \hat{\tau}_{2kj}}. \tag{54}$$

We also consider the reconstruction performance of the proposed algorithm. The centroid of each class is estimated by

$$\langle \tilde{\boldsymbol{\mu}}_k \rangle = \langle \mathbf{B} \rangle \langle \boldsymbol{\mu}_k \rangle + \langle \mathbf{I} - \mathbf{B} \rangle \langle \boldsymbol{\mu}_0 \rangle, \tag{55}$$

where $\langle \mathbf{B} \rangle = \mathrm{diag}(\langle \beta_1 \rangle, \langle \beta_2 \rangle, \ldots, \langle \beta_d \rangle)$, $\langle \boldsymbol{\mu}_k \rangle = (\hat{s}_{k1}, \hat{s}_{k2}, \ldots, \hat{s}_{kd})^T$ and $\langle \boldsymbol{\mu}_0 \rangle = (\hat{s}_{01}, \hat{s}_{02}, \ldots, \hat{s}_{0d})^T$. Then, reconstruction for the nth individual in class k can be computed as

$$\hat{\boldsymbol{y}}_n = \langle \mathbf{W}_k \rangle \langle \mathbf{R}_{nk} \boldsymbol{x}_{nk} \rangle + \langle \tilde{\boldsymbol{\mu}}_k \rangle, \tag{56}$$

where $\langle \mathbf{W}_k \rangle = [\hat{\boldsymbol{m}}_{k1}, \hat{\boldsymbol{m}}_{k2}, \ldots, \hat{\boldsymbol{m}}_{kd}]^T$ and $\langle \mathbf{R}_{nk} \boldsymbol{x}_{nk} \rangle$ is obtained from the VB-E step after the algorithm converges.

5. Experiment Study

5.1. Experiments on Synthetic Data

In this section, we justify the developed model and the structured VB algorithm using controlled experiments. We continue the experiments in [14] with the same synthetic data where the features were generated independently in each component. An additional set of data was generated where we imposed correlation between features within the mixture component. The proposed model and algorithm was compared with the semi-Bayesian clustering model and algorithm in [10], called varFnMS, in which a finite mixture of Gaussian is adopted and a mean-field VB is applied and compared with the full-Bayesian model and algorithm in [14], denoted varFnMS-T, which is based on the mixture of Student's t distribution and uses the structured VB algorithm.

The synthetic data in [14] contain 800 data points from four well-separated classes. The data are 10-dimensional with two influential features located around the class centers $(0,3)$, $(1,9)$, $(6,4)$ and $(7,10)$ with identity covariance matrices in each class. The remaining eight "noisy" features were sampled from $\mathcal{N}(0,1)$. We made randomly 1% of the data outliers by adding noises sampled uniformly from $[-10,10]^{10}$. The features are mutually independent in each class, which is consistent with the assumption underlying varFnMS and varFnMS-T. In the additional set of data, the local independence assumption is violated. We assigned a four-factor model for class 1, a two-factor model for class 2, a one-factor model for class 3 and no factor in class 4. The mean vector of each factor model remained the same as that in the "locally independent" data. The factor loading matrices were generated randomly with each entry from $\mathcal{N}(0,1)$. The noise term in each class was generated from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{10})$.

The proposed algorithm, denoted as varFnMS-TFA, the varFnMS and the varFnMS-T were carried out twenty times, separately. The number of clusters K was set as four. The K-mean clustering algorithm was used to initialize the posterior $q(z_n)$. The feature saliency and factor activity were both initialized as 0.5. The hyperparameters $\alpha_{0k}, \kappa_1, \kappa_2, \tau_1, \tau_2, \lambda_0, m_0,$

η_0 and ξ_0 were set to be 10^{-5} and $\mathbf{s}_0$ was set as the empirical mean of the feature data. We assumed a nine-factor model for each class at the beginning and initialized the posterior means of the latent factors by sampling from $\mathcal{N}(\mathbf{0}, \mathbf{I}_9)$. The algorithm terminates when the difference of the ELBO between two consecutive iterations is less than 10^{-7} or the maximum number of iterations ($IterMax = 500$) is reached. To avoid the "label switching" problems, we labeled the obtained clusters from the twenty repeated experiments by matching with the true classification of the data.

The ELBO reached and the classification error rate comparing the clustering with the original grouping of the data via the three algorithms with the two synthetic datasets are presented in Table 1. For the dataset where features are mutually independent within class, the classification accuracy of varFnMS-TFA is slightly higher than the other two algorithms, but the difference is not evident. For the dataset generated with correlated features, the proposed algorithm shows significantly higher accuracy than the other two algorithms under the local independence assumption. Moreover, the ELBO reached via varFnMS-TFA is the highest on average in both datasets and the discrepancy is enlarged where the features are locally correlated. As seen in Figure 2, it successfully captures the correlation across features through the latent factors.

Table 1. Evidence lower bound (ELBO) and classification error obtained via varFnMS, varFnMS-T and varFnMS-TFA for the two synthetic datasets where the features are locally independent and correlated, separately.

	Independent		**Correlated**	
Algorithm	**ELBO**	**Error**	**ELBO**	**Error**
varFnMS	−13,320.803 (136.822)	0.076 (0.118)	−16,868.400 (113.338)	0.351 (0.071)
varFnMS-T	−13,140.178 (763.027)	0.073 (0.111)	−16,468.433 (135.973)	0.302 (0.083)
varFnMS-TFA	−9082.876 (3320.315)	0.071 (0.109)	−8854.751 (2696.577)	0.051 (0.053)

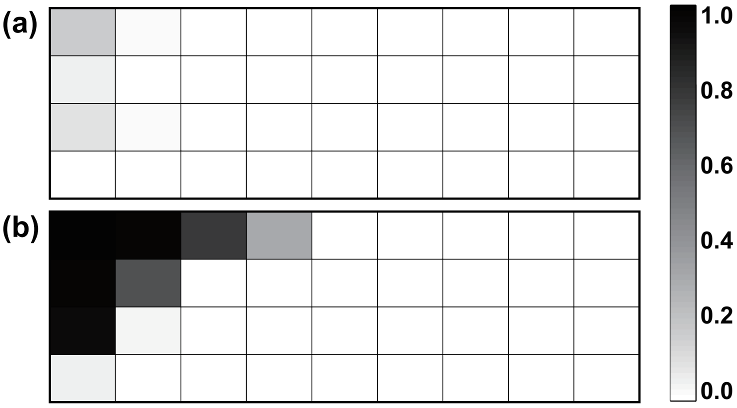

Figure 2. Factor activity estimated by the proposed algorithm for the two synthetic datasets where the features are locally (**a**) independent and (**b**) correlated, separately.

The estimated factor activity in each class by the proposed algorithm (averaged over the twenty repeats) for the two synthetic datasets is presented in Figure 2. Generally, the algorithm recovers the ground truth in both datasets. It can be seen in subplot (a) that there is no significantly active factor across the four classes for the "locally independent" data and the true pattern of factor activity in the "locally correlated" data is recovered as shown in subplot (b).

Figure 3 compares the estimated feature saliency for the two synthetic datasets. As shown in subplot (a), the three algorithms make the same good estimation on the feature saliency when the features are independent within class. But when the dependence relationship is imposed, the proposed algorithm shows apparently different behavior from the other two algorithms as shown in subplot (b). While varFnMS and varFnMS-T estimate the salience of a feature that could be confounded by the other features, varFnMS-TFA gives the factor-adjusted feature saliency, where the confounding effects are resolved by the latent factors.

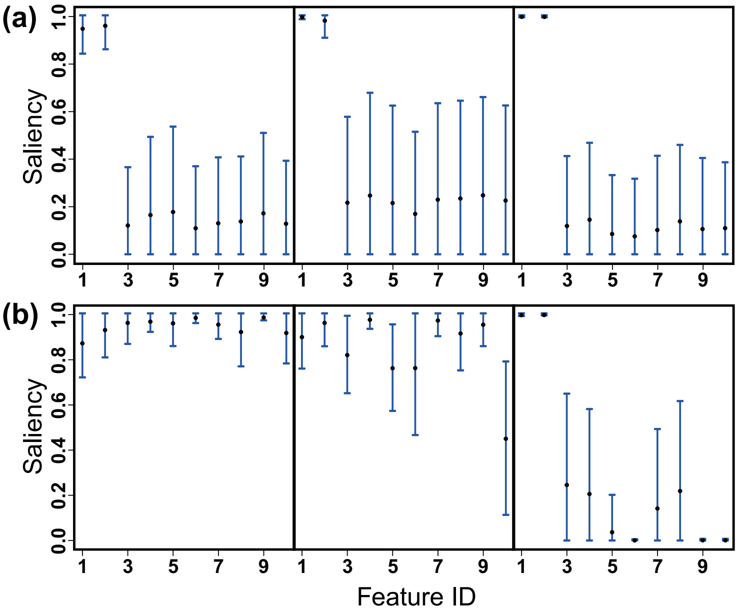

Figure 3. Feature saliency estimated via varFnMS (**left**), varFnMS-T (**middle**) and varFnMS-TFA (**right**) for the two synthetic datasets where the features are locally (**a**) independent and (**b**) correlated, separately.

In Table 2, the estimated class centroids in the two synthetic datasets (averaged over the twenty repeats) are present. When features are generated independently within class, the three algorithms exhibit comparable performance and recover the real centroids approximately. But with correlations imposed, the estimation accuracy by varFnMS or varFnMS-T is apparently degraded. In class 1, 2 and 3, they misestimate the means of the first two features that are salient and the variations of estimation are significantly enlarged compared with the results of varFnMS-TFA. In comparison, the varFnMS-TFA algorithm that considers the local dependence relationships gives more accurate and stable estimation results.

Table 2. The centroids estimated via varFnMS, varFnMS-T and varFnMS-TFA for the two synthetic datasets where the features are locally independent and correlated, separately.

		Independent			**Correlated**	
Class 1	**varFnMS**	**varFnMS-T**	**varFnMS-TFA**	**varFnMS**	**varFnMS-T**	**varFnMS-TFA**
$\mu_{11} = 0$	0.405 (0.791)	0.399 (0.910)	0.392 (0.906)	0.739 (1.056)	0.064 (0.569)	0.019 (0.140)
$\mu_{12} = 3$	3.289 (0.664)	3.178 (0.471)	3.223 (0.670)	3.922 (1.405)	1.771 (1.270)	3.093 (0.201)
$\mu_{13} = 0$	0.000 (0.037)	−0.107 (0.497)	−0.116 (0.490)	−0.314 (0.760)	−1.730 (0.952)	−0.001 (0.121)
$\mu_{14} = 0$	−0.007 (0.049)	0.041 (0.178)	−0.027 (0.096)	−0.009 (0.776)	−1.053 (0.814)	−0.025 (0.078)
$\mu_{15} = 0$	0.002 (0.043)	−0.025 (0.098)	0.000 (0.044)	−0.271 (0.915)	−1.748 (0.869)	−0.020 (0.041)
$\mu_{16} = 0$	0.005 (0.044)	−0.086 (0.393)	−0.031 (0.121)	0.011 (0.210)	−0.390 (0.428)	0.000 (0.000)
$\mu_{17} = 0$	−0.016 (0.043)	−0.073 (0.292)	−0.017 (0.046)	−0.185 (0.543)	0.060 (0.537)	−0.014 (0.046)
$\mu_{18} = 0$	−0.007 (0.039)	−0.051 (0.194)	−0.032 (0.119)	−0.004 (0.448)	0.340 (0.670)	−0.052 (0.180)
$\mu_{19} = 0$	0.034 (0.050)	0.075 (0.228)	0.015 (0.032)	0.019 (0.866)	1.410 (0.979)	−0.012 (0.028)
$\mu_{1,10} = 0$	−0.021 (0.040)	−0.168 (0.662)	−0.034 (0.125)	0.026 (0.422)	0.262 (0.237)	−0.004 (0.030)

Table 2. *Cont.*

	Independent			Correlated		
Class 2	**varFnMS**	**varFnMS-T**	**varFnMS-TFA**	**varFnMS**	**varFnMS-T**	**varFnMS-TFA**
$\mu_{21} = 1$	1.332 (0.650)	1.096 (0.447)	1.058 (0.282)	2.000 (0.939)	1.129 (0.678)	1.049 (0.151)
$\mu_{22} = 9$	8.798 (0.802)	8.875 (0.478)	8.793 (0.862)	6.712 (1.229)	6.762 (0.863)	9.012 (0.101)
$\mu_{23} = 0$	0.100 (0.472)	0.136 (0.622)	0.040 (0.209)	−0.024 (0.984)	0.340 (0.315)	0.008 (0.066)
$\mu_{24} = 0$	0.002 (0.057)	0.015 (0.047)	0.008 (0.035)	0.071 (0.738)	0.487 (0.792)	0.042 (0.091)
$\mu_{25} = 0$	−0.207 (0.926)	−0.228 (1.006)	−0.192 (0.865)	−0.028 (0.647)	0.394 (0.321)	−0.011 (0.028)
$\mu_{26} = 0$	−0.049 (0.138)	−0.049 (0.167)	−0.012 (0.037)	−0.107 (0.566)	0.124 (0.305)	0.004 (0.019)
$\mu_{27} = 0$	−0.034 (0.200)	−0.053 (0.231)	−0.001 (0.027)	−0.293 (0.578)	−0.468 (0.615)	−0.007 (0.031)
$\mu_{28} = 0$	−0.083 (0.367)	−0.094 (0.423)	−0.093 (0.417)	−0.052 (0.684)	−0.213 (0.314)	0.008 (0.067)
$\mu_{29} = 0$	0.036 (0.189)	0.039 (0.202)	0.004 (0.033)	−0.340 (0.611)	−0.702 (0.637)	−0.009 (0.027)
$\mu_{2,10} = 0$	0.009 (0.129)	0.018 (0.193)	0.040 (0.245)	−0.099 (0.550)	−0.147 (0.160)	0.002 (0.032)
Class 3						
$\mu_{31} = 6$	5.684 (1.136)	5.691 (0.998)	5.758 (0.767)	4.773 (1.507)	5.313 (1.603)	6.013 (0.158)
$\mu_{32} = 4$	4.392 (0.843)	4.514 (1.246)	4.553 (1.464)	3.962 (0.782)	3.644 (0.726)	4.174 (0.656)
$\mu_{33} = 0$	−0.082 (0.504)	0.056 (0.402)	0.064 (0.215)	0.206 (0.502)	0.127 (0.454)	0.018 (0.049)
$\mu_{34} = 0$	0.114 (0.570)	−0.015 (0.722)	−0.063 (0.631)	0.053 (0.466)	−0.204 (0.516)	0.006 (0.037)
$\mu_{35} = 0$	−0.025 (0.289)	−0.015 (0.330)	0.062 (0.305)	0.197 (0.517)	0.131 (0.428)	−0.011 (0.019)
$\mu_{36} = 0$	−0.181 (0.557)	−0.032 (0.632)	−0.087 (0.854)	0.076 (0.390)	0.064 (0.196)	0.002 (0.008)
$\mu_{37} = 0$	−0.137 (0.646)	−0.176 (0.761)	−0.213 (0.649)	0.269 (0.495)	0.703 (0.619)	−0.005 (0.051)
$\mu_{38} = 0$	−0.241 (0.744)	−0.332 (0.946)	−0.141 (0.465)	−0.170 (0.599)	0.216 (0.427)	0.010 (0.034)
$\mu_{39} = 0$	0.030 (0.270)	−0.030 (0.196)	−0.017 (0.158)	−0.023 (0.711)	0.468 (0.629)	−0.012 (0.033)
$\mu_{3,10} = 0$	−0.309 (0.771)	−0.304 (0.767)	−0.142 (0.416)	−0.017 (0.512)	0.068 (0.156)	−0.001 (0.030)
Class 4						
$\mu_{41} = 7$	6.792 (0.752)	6.966 (0.130)	6.960 (0.123)	5.443 (1.236)	6.329 (0.899)	6.709 (1.289)
$\mu_{42} = 10$	9.857 (0.521)	9.830 (0.689)	9.831 (0.688)	9.039 (1.183)	9.649 (0.623)	9.741 (0.995)
$\mu_{43} = 0$	0.002 (0.051)	−0.007 (0.060)	−0.005 (0.045)	0.075 (0.100)	0.034 (0.079)	−0.006 (0.046)
$\mu_{44} = 0$	0.000 (0.053)	0.005 (0.036)	0.000 (0.021)	−0.072 (0.111)	−0.061 (0.129)	0.380 (1.725)
$\mu_{45} = 0$	0.008 (0.043)	0.009 (0.041)	0.002 (0.032)	0.043 (0.175)	−0.013 (0.096)	−0.015 (0.027)
$\mu_{46} = 0$	−0.003 (0.041)	−0.003 (0.037)	−0.006 (0.025)	−0.017 (0.095)	−0.044 (0.084)	0.002 (0.008)
$\mu_{47} = 0$	−0.025 (0.063)	−0.011 (0.058)	0.002 (0.026)	0.111 (0.117)	0.079 (0.098)	−0.097 (0.425)
$\mu_{48} = 0$	−0.003 (0.051)	−0.005 (0.045)	−0.004 (0.031)	−0.053 (0.106)	−0.026 (0.077)	0.081 (0.434)
$\mu_{49} = 0$	0.029 (0.055)	0.021 (0.051)	0.016 (0.032)	0.071 (0.118)	0.050 (0.106)	−0.011 (0.035)
$\mu_{4,10} = 0$	−0.026 (0.048)	−0.027 (0.049)	−0.011 (0.036)	−0.001 (0.059)	0.015 (0.053)	−0.001 (0.031)

5.2. Experiments on Real Datasets

In this section, we apply the proposed model on the benchmark datasets: Iris, Olive, Wine and WDBC. The Iris dataset is obtained from the R package "datasets". Olive is the Italian olive oil dataset and Wine the Italian wine dataset. They are both obtained from the R package "pgmm". WDBC is the Wisconsin diagnostic breast cancer dataset downloaded from the UCI machine learning repository (https://doi.org/10.24432/C5DW2B; accessed on 26 May 2023). For each dataset, we repeated each algorithm ten times and retrieved the result with the highest value on ELBO. We set the initial dimensions of latent factors for each dataset as $d - 1$, where d is the number of features in the data. Table 3 presents the basic information for the four datasets and the classification error obtained. There is a significant decrease on the classification error for the Olive data and a slight improvement on the results for Iris and WDBC when using the proposed algorithm. The exception goes to the Wine data, where the proposed algorithm gives results slightly inferior to varFnMS-T.

Figures 4 and 5 show the factor activity and feature saliency for the four benchmark datasets. It can be seen from Figure 4 that strong factor activity is detected in Iris, Olive and WDBC data. As shown in Figure 5, the patterns of estimated feature saliency are noticeably changed when applying the proposed algorithm. The combined results indicate that the correlation between features could interfere with our decision about the features' relevance and the classification of data.

Table 3. Classification error obtained via varFnMS, varFnMS-T and varFnMFAS-T for the four benchmark datasets.

Dataset	N	d	K	varFnMS	varFnMS-T	varFnMS-TFA
Iris	150	4	3	0.093	0.093	0.020
Olive	572	8	3	0.203	0.199	0.042
Wine	178	27	3	0.079	0.062	0.073
WDBC	569	30	2	0.095	0.095	0.088

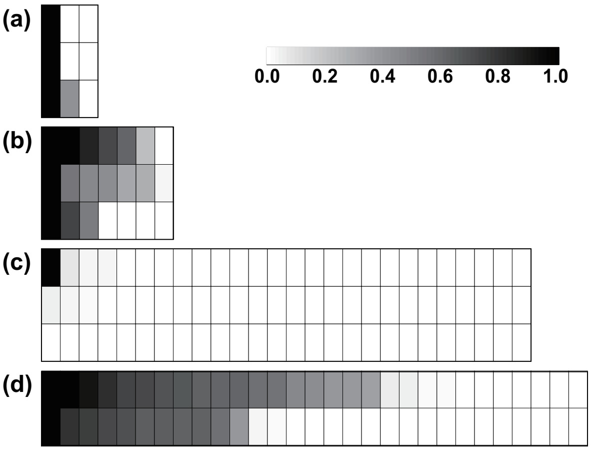

Figure 4. Factor activity estimated via the proposed algorithm for the four benchmark datasets: (**a**) Iris, (**b**) Olive, (**c**) Wine and (**d**) WDBC.

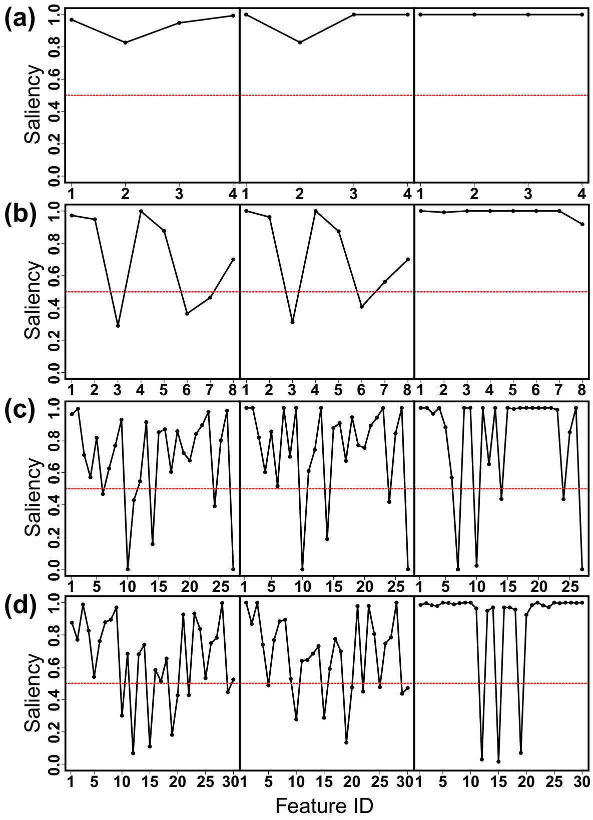

Figure 5. Feature saliency estimated via varFnMS (**left**), varFnMS-T (**middle**) and varFnMS-TFA (**right**) for the four benchmark datasets: (**a**) Iris, (**b**) Olive, (**c**) Wine and (**d**) WDBC. The saliency level at 0.5 is marked by the red dotted line.

5.3. Application on Handwritten Object Recognition

In this section, we apply the developed algorithm to the machine learning task of handwritten alphabet recognition. The handwritten alphabet dataset is obtained from the Kaggle webpage (https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format/data; accessed on 26 May 2023). It contains more than 370,000 images for the English alphabets (A–Z). The images are gray-scale in the size of 28 × 28 pixels. We focus on separation of the handwritten alphabets A, B and C and reserve randomly 200 images for each of the alphabets. As the variability of some pixels in the image of an alphabet is exactly zero, we may encounter the singularity problem during iterations of the clustering algorithms. Therefore, pre-processing was implemented on the data as detailed in Appendix B. In the proposed algorithm, the initial number of latent factors was set as fifty for each class. The three algorithms attain the same classification error rate as 0.16. The patterns of feature saliency estimated via varFnMS, varFnMS-T and the proposed algorithm are compared in Figure 6. The pixels are arranged along the x-axis column by column in the 28 × 28-pixel image. The saliences for the margin of the image with almost zero variability have been set as zero in the pre-processing stage. As can be seen from Figure 6, for the potentially discriminant part of the image, while the other two algorithms may have ambiguity concerning deciding the relevance of features, the evaluation based on the proposed algorithm is clearer which could be an improvement by extracting the confounding effects through latent factors.

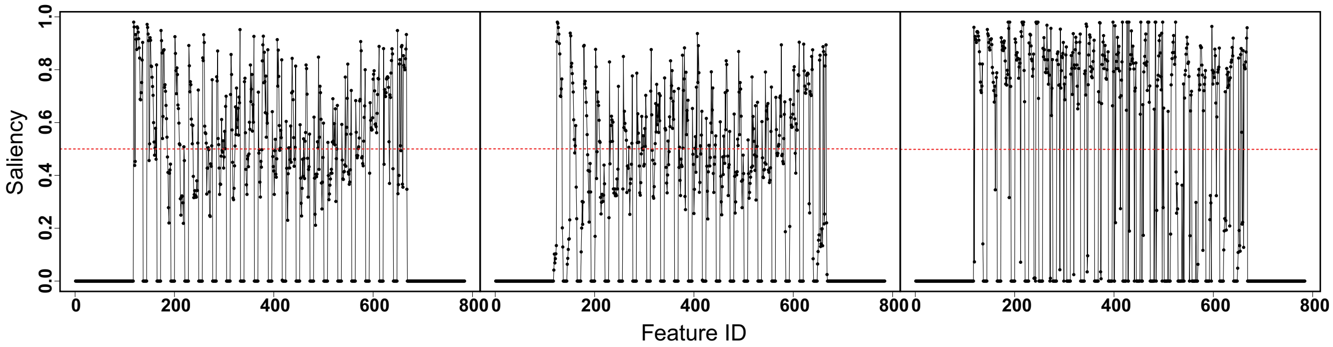

Figure 6. Feature saliency estimated via varFnMS (**left**), varFnMS-T (**middle**) and varFnMS-TFA (**right**) for the handwritten alphabet data. The saliency level at 0.5 is marked by the red dotted line.

The centroids estimated via the three algorithms are shown in Figure 7, where the reconstruction process of images via the proposed algorithm is also illustrated. The estimated centroid in each class can be calculated following (55), which is a mixing of the class-specific mean and the background. The calculation is outlined by the red box, where the centroid, the class-specific mean and the background are present from left to right, successively. It can be seen that all three algorithms exhibit good performance with respect to characterizing the alphabets. The estimated centroids can sketch the general appearance of the alphabets. But it is noticeable that varFnMS and varFnMS-T make some mistakes on estimation of the background. There should have been no handwritten stroke at the bottom of the background image, since this distinguishes the images of alphabet A. The proposed algorithm performs well with respect to reconstructing the images. An example of reconstruction is present in subplot (c). Additional examples are present in Appendix C. Generally, by adding the influence of latent factors, handwriting on the images becomes legible. The factor loadings on the two most active factors for each alphabet are shown at the right side of subplot (c). As can be seen, the information from latent factors plays an important role in refining the images.

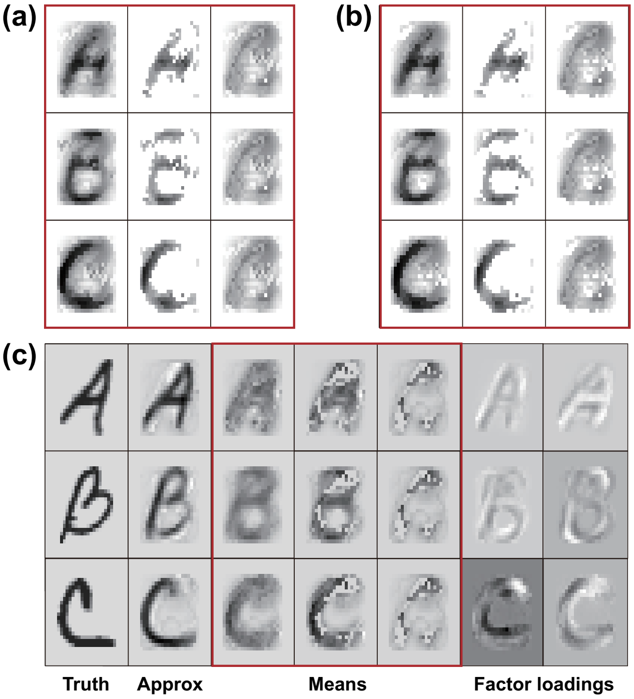

Figure 7. Reconstruction of the handwritten alphabet image via varFnMS (**a**), varFnMS-T (**b**) and varFnMS-TFA (**c**). The estimated centroids are outlined with a red box, where the centroid, the class-specific mean and the background are present from left to right, successively.

6. Conclusions

In this paper, we developed a hierarchical latent variable model for robust clustering and model selection. We considered the cases where features are correlated within mixture components in a Student's t mixture model. Factor-adjusted feature saliency was proposed to evaluate the relevance of features to data separation. Automatic latent dimension reduction was achieved by introducing the variables of factor activity. A full Bayesian treatment was adopted and a structured VB inference framework was developed that have enabled a tighter bond to the marginal likelihood and improved the inference accuracy. Controlled experiments on synthetic and real-world datasets showed that the proposed model is able to capture the correlation between features and shows better clustering performance than the models relying on the local independence assumption. Application of the developed algorithm on the high-dimensional handwritten alphabet data showed its applicability and usefulness for image recognition and reconstruction.

In the proposed model, we take the number of clusters (number of components in the mixture model) as fixed and given before inference. An ongoing work is to extend our model to realize automatic selection of the number of clusters. We imposed the Dirichlet prior on the mixing probabilities, which can act as a penalization to drive the mixing probabilities associated with unnecessary components towards extinction. We will also investigate the novel penalization methods proposed in [34] which result in continuous objective functions and can shrink the mixing weights to exactly zero. Other limitations include assuming that features are approximated Gaussian distributed in each component. This assumption can be violated when the features only take positive values or follow skewed distributions. Future work may consider extending the model to tackle these scenarios.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math12071091/s1, S1. Deriving the auxiliary posteriors of the latent variables; S2. Deriving the auxiliary posteriors of the parameters.

Author Contributions: Conceptualization, Y.N. and W.X.; methodology, S.F.; software, S.F.; validation, S.F.; formal analysis, S.F.; writing—original draft preparation, S.F.; writing—review and editing, W.X.; visualization, S.F.; supervision, Y.N.; funding acquisition, Y.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China grant number 2020YFA0713603 and the National Natural Science Foundation of China grant number 11971386.

Data Availability Statement: The five datasets used in this study are publicly available. The Iris data can be obtained from the R package "datasets". (The R software can be downloaded at http://www.r-project.org/; R version is 4.3.1. (accessed on 26 May 2023)) The Olive and Wine data can be obtained from the R package "pgmm". The Wisconsin diagnostic breast cancer (WDBC) data can be downloaded from the UCI machine learning repository at https://doi.org/10.24432/C5DW2B (accessed on 26 May 2023). The handwritten alphabets data are openly available in the Kaggle webpage: https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format/data (accessed on 26 May 2023). The R codes for the developed algorithm are available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The evidence lower bound monitoring the optimization process of the proposed algorithm can be evaluated as follows:

$$
\begin{aligned}
\mathcal{L} = \sum_{n,k,l} \langle \delta_{z_n,k} \rangle \Big[& \langle \phi_{nl} \rangle \langle \log p(y_{nl} | \boldsymbol{x}_{nk}, \boldsymbol{r}_{nk}, u_{nl}, \phi_{nl} = 1, z_n = k) \rangle + \langle 1 - \phi_{nl} \rangle \langle \log p(y_{nl} | \boldsymbol{x}_{nk}, \boldsymbol{r}_{nk}, u_{nl}, \phi_{nl} = 0, z_n = k) \rangle \\
& + \langle \phi_{nl} \rangle \langle \log p(u_{nl} | \phi_{nl} = 1, z_n = k) - \log q(u_{nl} | \phi_{nl} = 1, z_n = k) \rangle \Big] \\
& + \sum_{n,k} \langle \delta_{z_n,k} \rangle \Big[\langle \log p(\boldsymbol{x}_{nk}) - \log q(\boldsymbol{x}_{nk} | \boldsymbol{r}_{nk}) \rangle + \langle \log p(\boldsymbol{r}_{nk}) - \log q(\boldsymbol{r}_{nk}) \rangle + \langle \log p(z_n = k) \rangle - \log q(z_n = k) \Big] \\
& + \sum_{n,l} \Big[\langle 1 - \phi_{nl} \rangle \langle \log p(u_{nl} | \phi_{nl} = 0) - \log q(u_{nl} | \phi_{nl} = 0) \rangle + \langle \log p(\phi_{nl}) - \log q(\phi_{nl}) \rangle \Big] \\
& + \sum_{l} \langle \log p(\mu_{0l}) - \log q(\mu_{0l}) + \log p(\sigma_{0l}) - \log q(\sigma_{0l}) + \log p(\beta_l) - \log q(\beta_l) \rangle \\
& + \sum_{k,l} \langle \log p(\mu_{kl}) - \log q(\mu_{kl}) + \log p(\sigma_{kl}) - \log q(\sigma_{kl}) + \log p(\boldsymbol{w}_{kl}) - \log q(\boldsymbol{w}_{kl}) \rangle \\
& + \sum_{k,j} \langle \log p(\rho_{kj}) - \log q(\rho_{kj}) \rangle + \langle \log p(\boldsymbol{\pi}) - \log q(\boldsymbol{\pi}) \rangle.
\end{aligned}
\tag{A1}
$$

Table A1 lists the computation for the expectations in the evidence lower bound (A1).

Table A1. Evaluation of the evidence lower bound.

		Expectations of the Logarithm of Priors of the Latent Variables
$\langle \log p(y_{nl} \vert \boldsymbol{x}_{nk}, \boldsymbol{r}_{nk}, u_{nl}, \phi_{nl} = 1, z_n = k) \rangle$	$=$	$\frac{1}{2} \langle \log \sigma_{kl} \rangle + \frac{1}{2} \langle \log u_{nl} \rangle_k^1 - \frac{1}{2} \langle \sigma_{kl} \rangle \langle u_{nl} \rangle_k^1 \langle (\tilde{y}_{nl}^k - \mu_{kl})^2 \rangle + \text{const.}$
$\langle \log p(y_{nl} \vert \boldsymbol{x}_{nk}, \boldsymbol{r}_{nk}, u_{nl}, \phi_{nl} = 0, z_n = k) \rangle$	$=$	$\frac{1}{2} \langle \log \sigma_{0l} \rangle + \frac{1}{2} \langle \log u_{nl} \rangle^0 - \frac{1}{2} \langle \sigma_{0l} \rangle \langle u_{nl} \rangle^0 \langle (\tilde{y}_{nl}^k - \mu_{0l})^2 \rangle + \text{const.}$
$\langle \log p(u_{nl} \vert \phi_{nl} = 1, z_n = k) \rangle$	$=$	$\frac{v_{kl}}{2} \log \frac{v_{kl}}{2} - \log \Gamma\left(\frac{v_{kl}}{2}\right) + \left(\frac{v_{kl}}{2} - 1\right) \langle \log u_{nl} \rangle_k^1 - \frac{v_{kl}}{2} \langle u_{nl} \rangle_k^1$
$\langle \log p(u_{nl} \vert \phi_{nl} = 0) \rangle$	$=$	$\frac{v_{0l}}{2} \log \frac{v_{0l}}{2} - \log \Gamma\left(\frac{v_{0l}}{2}\right) + \left(\frac{v_{0l}}{2} - 1\right) \langle \log u_{nl} \rangle^0 - \frac{v_{0l}}{2} \langle u_{nl} \rangle^0$
$\langle \log p(\phi_{nl}) \rangle$	$=$	$\langle \phi_{nl} \rangle \langle \log \beta_l \rangle + \langle 1 - \phi_{nl} \rangle \langle \log(1 - \beta_l) \rangle$
$\langle \log p(\boldsymbol{x}_{nk}) \rangle$	$=$	$-\frac{p_k}{2} \log 2\pi - \frac{1}{2} \text{tr} \langle \hat{\mathbf{C}}_n^k(\boldsymbol{r}_{nk})^{-1} + \hat{\boldsymbol{f}}_n^k(\boldsymbol{r}_{nk}) \hat{\boldsymbol{f}}_n^k(\boldsymbol{r}_{nk})^T \rangle$
$\langle \log p(\boldsymbol{r}_{nk}) \rangle$	$=$	$\sum_j [\langle r_{nkj} \rangle \langle \log \rho_{kj} \rangle + \langle 1 - r_{nkj} \rangle \langle \log(1 - \rho_{kj}) \rangle]$
$\langle \log p(z_n = k) \rangle$	$=$	$\langle \log \pi_k \rangle$
$\langle \log p(\boldsymbol{\pi}) \rangle$	$=$	$\sum_k (\alpha_{0k} - 1) \langle \log \pi_k \rangle + \text{const.}$
$\langle \log p(\beta_l) \rangle$	$=$	$(\kappa_1 - 1) \langle \log \beta_l \rangle + (\kappa_2 - 1) \langle \log(1 - \beta_l) \rangle + \text{const.}$
$\langle \log p(\rho_{kj}) \rangle$	$=$	$(\tau_1 - 1) \langle \log \rho_{kj} \rangle + (\tau_2 - 1) \langle \log(1 - \rho_{kj}) \rangle + \text{const.}$
$\langle \log p(\mu_{kl}) \rangle$	$=$	$-\frac{1}{2} \lambda_0 \left[\hat{\lambda}_{kl}^{-1} + (\hat{s}_{kl} - s_{0l})^2 \right] + \text{const.}$

Table A1. *Cont.*

		Expectations of the Logarithm of Priors of the Latent Variables
$\langle \log p(\mu_{0l}) \rangle$	$=$	$-\frac{1}{2}\lambda_0 \left[\hat{\lambda}_{0l}^{-1} + (\hat{s}_{0l} - s_{0l})^2 \right] + \text{const.}$
$\langle \log p(\sigma_{kl}) \rangle$	$=$	$\left(\frac{\eta_0}{2} - 1 \right) \langle \log \sigma_{kl} \rangle - \frac{\xi_0}{2} \langle \sigma_{kl} \rangle + \text{const.}$
$\langle \log p(\sigma_{0l}) \rangle$	$=$	$\left(\frac{\eta_0}{2} - 1 \right) \langle \log \sigma_{0l} \rangle - \frac{\xi_0}{2} \langle \sigma_{0l} \rangle + \text{const.}$
$\langle \log p(\boldsymbol{w}_{kl}) \rangle$	$=$	$-\frac{p_k}{2} \log 2\pi - \frac{1}{2} m_0 \text{tr}\left(\hat{\mathbf{M}}_{kl}^{-1} + \hat{\boldsymbol{m}}_{kl} \hat{\boldsymbol{m}}_{kl}^T \right)$
		Expectations of the Logarithm of the Auxiliary Posteriors
$\langle \log q(u_{nl} \vert \phi_{nl} = 1, z_n = k) \rangle$	$=$	$\hat{a}_{kl} \log \hat{b}_{nl}^k - \log \Gamma(\hat{a}_{kl}) + (\hat{a}_{kl} - 1) \langle \log u_{nl} \rangle_k^1 - \hat{b}_{nl}^k \langle u_{nl} \rangle_k^1$
$\langle \log q(u_{nl} \vert \phi_{nl} = 0) \rangle$	$=$	$\hat{a}_{0l} \log \hat{b}_{nl}^0 - \log \Gamma(\hat{a}_{0l}) + (\hat{a}_{0l} - 1) \langle \log u_{nl} \rangle^0 - \hat{b}_{nl}^0 \langle u_{nl} \rangle^0$
$\langle \log q(\phi_{nl}) \rangle$	$=$	$\langle \phi_{nl} \rangle \log q(\phi_{nl} = 1) + \langle 1 - \phi_{nl} \rangle \log q(\phi_{nl} = 0)$
$\langle \log q(\boldsymbol{x}_{nk} \vert \boldsymbol{r}_{nk}) \rangle$	$=$	$-\frac{p_k}{2} \log 2\pi + \frac{1}{2} \log \vert \hat{\mathbf{C}}_n^k(\boldsymbol{r}_{nk}) \vert - \frac{1}{2} p_k$
$\langle \log q(\boldsymbol{r}_{nk}) \rangle$	$=$	$\sum_j [\langle r_{nkj} \rangle \log q(r_{nkj} = 1) + \langle 1 - r_{nkj} \rangle \log q(r_{nkj} = 0)]$
$\langle \log q(\boldsymbol{\pi}) \rangle$	$=$	$\log \Gamma(\sum_k \hat{\alpha}_k) - \sum_k \log \Gamma(\hat{\alpha}_k) + \sum_k (\hat{\alpha}_k - 1) \langle \log \pi_k \rangle$
$\langle \log q(\beta_l) \rangle$	$=$	$(\hat{\kappa}_{1l} - 1) \langle \log \beta_l \rangle + (\hat{\kappa}_{2l} - 1) \langle \log(1 - \beta_l) \rangle - \log \mathcal{B}(\hat{\kappa}_{1l}, \hat{\kappa}_{2l})$
$\langle \log q(\rho_{kj}) \rangle$	$=$	$\left(\hat{\tau}_{1kj} - 1 \right) \langle \log \rho_{kj} \rangle + \left(\hat{\tau}_{2kj} - 1 \right) \langle \log(1 - \rho_{kj}) \rangle - \log \mathcal{B}(\hat{\tau}_{1kj}, \hat{\tau}_{2kj})$
$\langle \log q(\mu_{kl}) \rangle$	$=$	$\frac{1}{2} \log \hat{\lambda}_{kl} + \text{const.}$
$\langle \log q(\mu_{0l}) \rangle$	$=$	$\frac{1}{2} \log \hat{\lambda}_{0l} + \text{const.}$
$\langle \log q(\sigma_{kl}) \rangle$	$=$	$\frac{\hat{\eta}_{kl}}{2} \log \frac{\hat{\xi}_{kl}}{2} - \log \Gamma\left(\frac{\hat{\eta}_{kl}}{2} \right) + \left(\frac{\hat{\eta}_{kl}}{2} - 1 \right) \langle \log \sigma_{kl} \rangle - \frac{\hat{\xi}_{kl}}{2} \langle \sigma_{kl} \rangle$
$\langle \log q(\sigma_{0l}) \rangle$	$=$	$\frac{\hat{\eta}_{0l}}{2} \log \frac{\hat{\xi}_{0l}}{2} - \log \Gamma\left(\frac{\hat{\eta}_{0l}}{2} \right) + \left(\frac{\hat{\eta}_{0l}}{2} - 1 \right) \langle \log \sigma_{0l} \rangle - \frac{\hat{\xi}_{0l}}{2} \langle \sigma_{0l} \rangle$
$\langle \log q(\boldsymbol{w}_{kl}) \rangle$	$=$	$-\frac{p_k}{2} \log 2\pi + \frac{1}{2} \log \vert \hat{\mathbf{M}}_{kl} \vert - \frac{1}{2} p_k$

Appendix B

Figure A1 shows the frequency histogram of the standard deviation for features in the handwritten alphabet image. As can be seen, the values of standard deviation range from 0 to 120 and a large proportion of the features have zero variance or comparably small variance. Most of them lie on the marginal area of the image. To improve the computational efficiency, we removed the features with standard deviation smaller than ten at the pre-processing stage which left 400 features for the clustering task.

In addition, we observed the singularity problem during iterations of the clustering algorithms as the variability of some pixels in the image of an alphabet is exactly zero. This happens when the clustering of the images gets close to their original grouping. To tackle the problem, we put a noise mask on the data. Each element of the noise mask was generated from $\mathcal{N}(0, 0.1)$.

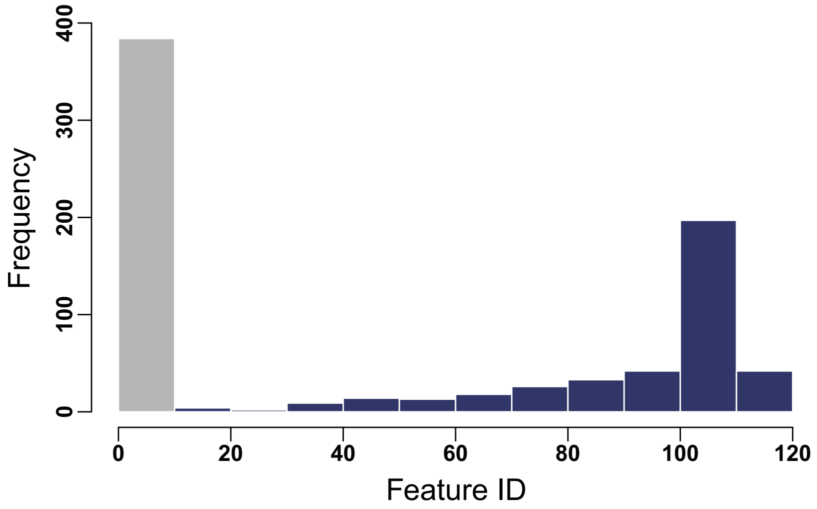

Figure A1. Frequency histogram of the standard deviation for the features in the handwritten alphabet data. The frequency bar corresponding to the standard deviation below 10 is marked in grey.

Appendix C

Figure A2 presents the reconstructed images for the alphabet A, B and C by the proposed algorithm.

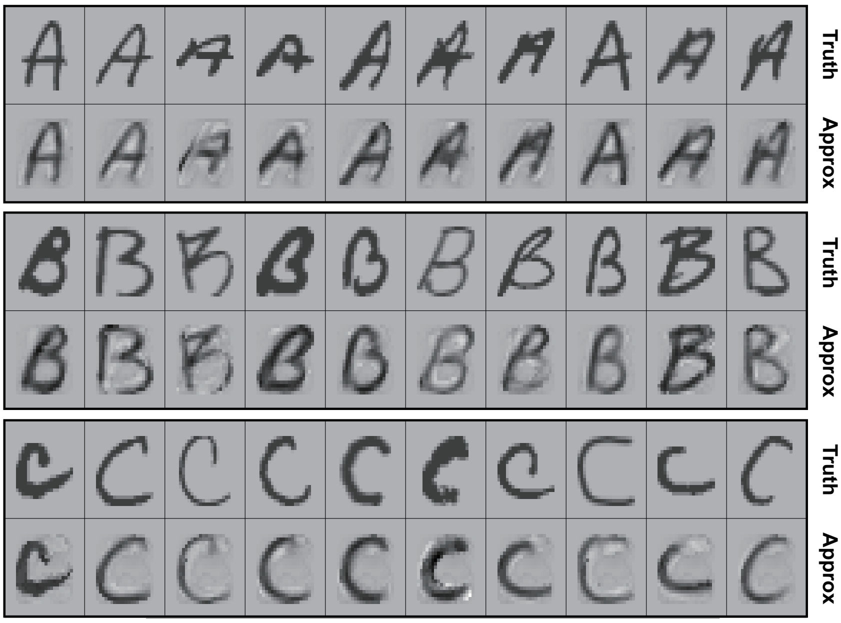

Figure A2. Reconstructed images for the handwritten alphabet A, B and C by the proposed algorithm.

References

1. Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; Zhou, H. Variational deep embedding: An unsupervised and generative approach to clustering. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 1965–1972.
2. Yang, L.; Cheung, N.M.; Li, J.; Fang, J. Deep clustering by Gaussian mixture variational autoencoders with graph embedding. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6440–6449.
3. Sun, J.; Zhou, A.; Keates, S.; Liao, S. Simultaneous Bayesian clustering and feature selection through student's t mixtures model. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1187–1199. [CrossRef] [PubMed]
4. Law, M.H.C.; Figueiredo, M.A.T.; Jain, A.K. Simultaneous feature selection and clustering using mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1154–1166. [CrossRef] [PubMed]
5. Bouveyron, C.; Brunet-Saumard, C. Model-based clustering of high-dimensional data: A review. *Comput. Stat. Data Anal.* **2014**, *71*, 52–78. [CrossRef]
6. Dash, M.; Liu, H. Feature selection for clustering. In Proceedings of the 4th International Conference on the Practical Application of Knowledge Discovery and Data Mining, Crowne Plaza Midland Hotel, Manchester, UK, 11–13 April 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 110–121.
7. Mitra, P.; Murthy, C.; Pal, S. Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 301–312. [CrossRef]
8. Pan, W.; Shen, X. Penalized model-based clustering with application to variable selection. *J. Mach. Learn. Res.* **2007**, *8*, 1145–1164.
9. Bhattacharya, S.; McNicholas, P.D. A LASSO-penalized BIC for mixture model selection. *Adv. Data Anal. Classif.* **2014**, *8*, 45–61. [CrossRef]
10. Constantinopoulos, C.; Titsias, M.K.; Likas, A. Bayesian feature and model selection for Gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1013–1018. [CrossRef] [PubMed]
11. Li, Y.; Dong, M.; Hua, J. Simultaneous localized feature selection and model detection for Gaussian mixtures. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 953–960.
12. Hong, X.; Li, H.; Miller, P.; Zhou, J.; Li, L.; Crookes, D.; Lu, Y.; Li, X.; Zhou, H. Component-based feature saliency for clustering. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 882–896. [CrossRef]
13. Zhang, H.; Wu, Q.M.J.; Nguyen, T.M. Variational Bayes and localized feature selection for student's t-mixture models. *Int. J. Pattern Recognit. Artif. Intell.* **2013**, *27*, 1350016. [CrossRef]
14. Sun, J.; Zhou, A. Unsupervised robust Bayesian feature selection. In Proceedings of the 2014 International Joint Conference on Neural Networks, Beijing, China, 6–11 July 2014; pp. 558–564.

15. Perthame, E.; Friguet, C.; Causeur, D. Stability of feature selection in classification issues for high-dimensional correlated data. *Stat. Comput.* **2016**, *26*, 783–796. [CrossRef]
16. Fan, J.; Ke, Y.; Wang, K. Factor-adjusted regularized model selection. *J. Econom.* **2020**, *216*, 71–85. [CrossRef] [PubMed]
17. Mai, Q.; Zou, H.; Yuan, M. A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika* **2012**, *99*, 29–42. [CrossRef]
18. Galimberti, G.; Soffritti, G. Using conditional independence for parsimonious model-based Gaussian clustering. *Stat. Comput.* **2013**, *23*, 625–638. [CrossRef]
19. Devijver, E.; Gallopin, M. Block-diagonal covariance selection for high-dimensional Gaussian graphical models. *J. Am. Stat. Assoc.* **2018**, *113*, 306–314. [CrossRef]
20. Ruan, L.; Yuan, M.; Zou, H. Regularized parameter estimation in high-dimensional Gaussian mixture models. *Neural Comput.* **2011**, *23*, 1605–1622. [CrossRef] [PubMed]
21. McLachlan, G.J.; Bean, R.W.; Jones, L.B.T. Extension of the mixture of factor analyzers model to incorporate the multivariate *t*-distribution. *Comput. Stat. Data Anal.* **2007**, *51*, 5327–5338. [CrossRef]
22. Archambeau, C.; Delannay, N.; Verleysen, M. Mixtures of robust probabilistic principal component analyzers. *Neurocomputing* **2008**, *71*, 1274–1282. [CrossRef]
23. McNicholas, P.D.; Murphy, T.B. Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics* **2010**, *26*, 2705–2712. [CrossRef]
24. Andrews, J.L.; McNicholas, P.D. Extending mixtures of multivariate *t*-factor analyzers. *Stat. Comput.* **2011**, *21*, 361–373. [CrossRef]
25. Wang, Z.; Lan, C. Towards a hierarchical Bayesian model of multi-view anomaly detection. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2020; pp. 2420–2426.
26. Mackay, D.J.C. Probable networks and plausible predictions—A review of practical Bayesian methods for supervised neural networks. *Netw. Comput. Neural Syst.* **1995**, *6*, 469–505. [CrossRef]
27. Bhattacharya, A.; Dunson, D.B. Sparse Bayesian infinite factor models. *Biometrika* **2011**, *98*, 291–306. [CrossRef] [PubMed]
28. Murphy, K.; Viroli, C.; Gormley, I.C. Infinite mixtures of infinite factor analysers. *Bayesian Anal.* **2020**, *15*, 937–963. [CrossRef]
29. Ormerod, J.T.; You, C.; Müller, S. A variational Bayes approach to variable selection. *Electron. J. Stat.* **2017**, *11*, 3549–3594. [CrossRef]
30. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1050–1059.
31. Beal, M.J.; Ghahramani, Z. The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures. In *Bayesian Statistic 7: Proceedings of the Seventh Valencia International Meeting*; Oxford University Press: Oxford, UK, 2003; pp. 453–463.
32. Teh, Y.W.; Newman, D.; Welling, M. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 19 Proceedings of the 2006 Conference*; MIT Press: Cambridge, MA, USA, 2006; pp. 1353–1360.
33. Zhang, C.X.; Xu, S.; Zhang, J.S. A novel variational Bayesian method for variable selection in logistic regression models. *Comput. Stat. Data Anal.* **2019**, *133*, 1–19. [CrossRef]
34. Huang, T.; Peng, H.; Zhang, K. Model selection for Gaussian mixture models. *Stat. Sin.* **2017**, *27*, 147–169. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.