Simultaneous Bayesian Clustering and Feature Selection Through Student's t Mixtures Model

Jianyong Sun, Aimin Zhou, *Member, IEEE*, Simeon Keates, and Shengbin Liao

***Abstract*— In this paper, we proposed a generative model for feature selection under the unsupervised learning context. The model assumes that data are independently and identically sampled from a finite mixture of Student's t distributions, which can reduce the sensitiveness to outliers. Latent random variables that represent the features' salience are included in the model for the indication of the relevance of features. As a result, the model is expected to simultaneously realize clustering, feature selection, and outlier detection. Inference is carried out by a tree-structured variational Bayes algorithm. Full Bayesian treatment is adopted in the model to realize automatic model selection. Controlled experimental studies showed that the developed model is capable of modeling the data set with outliers accurately. Furthermore, experiment results showed that the developed algorithm compares favorably against existing unsupervised probability model-based Bayesian feature selection algorithms on artificial and real data sets. Moreover, the application of the developed algorithm on real leukemia gene expression data indicated that it is able to identify the discriminating genes successfully.**

***Index Terms*— Bayesian inference, feature selection, robust clustering, tree-structured variational Bayes (VB).**

I. Introduction

COMPETITIVE performances of clustering algorithms cannot be expected on high-dimensional data sets due to the curse of dimensionality and the impact of redundancy and noise. Fortunately, the intrinsic dimensionality of a high-dimensional data set is usually much less than original feature space [1]–[3]. The performance of a learning algorithm could be improved significantly if a subset of features or a combination of features is properly selected [4]. Feature selection is to select a subset of most informative features (or attributes, variables) rather than selecting a combination of features (which is usually referred to as feature extraction), such as in principal component analysis and independent component analysis.

Existing feature selection algorithm can be categorized as supervised feature selection (on data with full class labels) [5]–[9], unsupervised feature selection (on data without class labels) [10]–[15], and semisupervised feature selection (on data with partial labels) [14], [16], [17]. Feature selection in unsupervised context is considered to be more difficult than the other two cases, since there is no target information available for training. The selected informative features must greatly preserve the distribution and the manifold structure of the data space. In this paper, we focused on unsupervised feature selection.

Various feature selection methods for unsupervised learning have been developed, which can be categorized according to different feature selection criteria. Criteria scores, such as Laplacian score [18], eigenvalue sensitive criteria [19], information entropy [20], and correlation [21], have been proposed. In [22], consistency-based feature selection methods were proposed and evaluated. To preserve pairwise similarity along data samples in the original data space, a similarity preserving feature selection framework is proposed in [11]. Local learning-based feature selection methods [13] have been extensively studied recently. For examples, in [23] and [24], subspace learning based on nonnegative matrix factorization is developed, where the loading matrix is penalized by L_2 and/or L_1 norms. Moreover, L_2, L_1, and $L_{2,1}$-norms have been widely applied in various feature selection methods, such as in [25]–[27]. In [17], a global and local structure preservation framework that integrates global pairwise sample similarity and local geometric data structure is proposed for feature selection. In [15] and [28]–[31], spectral learning aiming to preserve the underlying manifold structure is applied for selecting proper features. In [32], embedding learning and sparse regression are jointly applied to perform feature selection. A discrimination analysis based on a property of Fourier transform of the data density distribution is applied for feature selection via optic diffraction principle [10]. A theoretically optimal criterion, namely, the discriminative optimal criterion, has been developed for feature selection in [33].

Apart from these mentioned algorithms, clustering (which aims to discover data structure) can also be used as a criterion. Intuitively, informative feature subsets that greatly preserve the sample data distribution should vary at different clusters. In the wrapper method proposed in [4], a clustering algorithm is used to evaluate the candidate feature subsets. The performance of the wrapper method highly depends on the employed

The work of J. Sun was supported by the National Science Foundation of China under Grant 61573279, Grant 61573326, and Grant 11301494. The work of A. Zhou was supported by NSFC under Grant 61673180. The work of S.Liao was supported by the Key Science and Technology Project of Wuhan under Grant 2014010202010108. *(Corresponding authors: Jianyong Sun; Aimin Zhou.)*

J. Sun is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China, and also with the School of Computer Science, University of Essex, Colchester, CO4 3SQ, U.K. (e-mail: jysun@essex.ac.uk).

A. Zhou is with the Shanghai Key Laboratory of Multidimensional Information Processing, and also with the Department of Computer Science and Technology, East China Normal University, Shanghai 200062, China (e-mail: amzhou@cs.ecnu.edu.cn).

S. Keates is with the Faculty of Engineering and Science, University of Greenwich, Kent, ME4 4TB, U.K.

S. Liao is with the National Engineering Research Centre for E-Learning, Huazhong Normal University, Wuhan 430079, China.

clustering algorithms. Alternatively, clustering and feature selection are embedded together with a proper objective function. Subset features can be obtained by optimizing the objective function. It is well acknowledged that the choosing of feature subsets and the clustering estimation (including the cluster statistics and the optimal number of components) are highly dependent problem [34]. This clearly suggests that the two problems should be considered simultaneously.

Most of clustering-based feature selection methods were developed on finite Gaussian mixtures. Carbonetto *et al.* [35] proposed a Bayesian shrinkage approach where shrinkage hyperpriors are placed over the component means. The shrinkage hyperpriors can lead to automatic feature selection. Pan *et al.* [36] proposed a penalized likelihood approach where a L_1 penalty is imposed on the cluster means. The proposed approach can automatically realize feature selection through thresholding and model selection through the BIC criterion. Law *et al.* [37] defined the saliency of feature as a probability, which is to quantify whether the data distribution with respect to the saliency features can be sufficiently represented. They proposed to fit the Gaussian mixture model to the sample data distribution using the EM algorithm, while the MML criterion is employed for model selection. Moreover, a Bayesian treatment to the finite Gaussian mixture model that benefits from automatic model selection was proposed in [34]. Li *et al.* [38] improved their work by utilizing "localized" feature saliency to address the local intrinsic property of data.

Outliers or scattered objects exist elsewhere in real data sets. As well known, Gaussian mixture models are not able to deal with outliers properly. The outliers, if exist, should seriously deteriorate the performances of Gaussian-based clustering algorithms. Moreover, the presence of outliers could also lead to selecting a false model complexity, and make the optimal selection of a subset of informative features get much more difficult. Therefore, previous clustering-based feature selection methods cannot be expected to perform well on data with outliers. It is thus indispensable to propose a principled approach to realize the selection of the most informative features and the improvement on the clustering performance, while eliminating the bad effect of outlying data. This motives us to propose a finite mixture model that is able to deal with outliers; and to develop a Bayesian inference algorithm that can carry out unsupervised clustering, feature selection, and outlier detection simultaneously.

Specifically, in this paper, we propose a hierarchical latent variable model to address the three tasks. First of all, it has been a common practice to adopt heavy-tailed distributions for handling outlier data in the literature. The Student's t distribution is such a heavy-tail distribution, and has been widely used [39], [40]. In our model, we adopt a finite mixture of Student's t distributions as the backbone. Fig. 1 shows the difference between a Student's t distribution and a Gaussian distribution with the same mean and variance, but with different parameters (ν, also called the degree of freedom). There are other heavy-tail distribution is available, such as the Laplace distribution and the Pearson type-VII distribution [40], which can also be adopted for handling outliers. Note that the

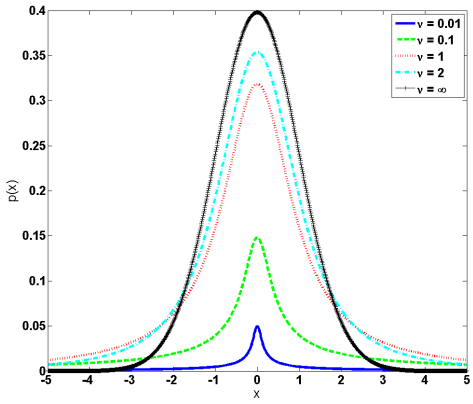


Fig. 1. Demonstration of the Student t-distribution with different parameters $\nu = 0.01, 0.1, 1, 2, \infty$. Note that the Student t-distribution becomes the Gaussian distribution in case $\nu = \infty$.

Student's t is a scalar mixture of Gaussians. This property makes the Student's t distribution convenient for inference, and hence popular for outlier detection.

Second, regarding feature selection, we propose to use a localized feature saliency similar to the approach developed in [38]. The feature saliency characterizes the importance of the feature and can be used as criterion for the selection of the most informative features. Localization of the feature saliency addresses the cluster effect on relevant feature subsets. Finally, to carry out model selection, we adopt a full Bayesian treatment to the model, where proper prior distributions are assumed for the parameters, including the number of clusters, the mixing proportions, and the parameters of the cluster components. To carry out inference, we resort to a tree-structured variational Bayesian (VB), since the likelihood function of the training data with respect to the proposed model is not tractable.

In the rest of this paper, Section II presents the proposed latent variable model. The inference is presented in Section III-A–III-F, in which the tree-structured VB algorithm is described. Moreover, the interpretation of the model is described in Section III-G. The experimental study is presented in Section IV. In this paper, controlled experiments were first carried out to justify the out performance of the developed models over the model using Gaussian distributions on synthetic data sets and another state-of-the-art feature selection algorithms. Then, the developed algorithm was compared with them on some real data sets. Section V concludes this paper.

II. Model

In this section, we present the proposed hierarchical latent variable model step by step starting from the introduction of saliency features to variables that are modeled to follow the mixture of Student's t. To make the description clear, Table I shows the notations used.

Suppose that a vector of random variable $Y = (Y_1, \ldots, Y_d) \in \mathbb{R}^d$, where d is the dimensionality of the input data, and denote Y_ℓ as the ℓth feature. In the sequel, we use $\mathbf{y}$ to represent the realization of Y. To represent if a feature is relevant or not, we use a vector of random binary variable $\Phi = (\phi_1, \ldots, \phi_d)$. That is, if $\phi_\ell = 1$, we say that the ℓth feature is relevant, and 0 otherwise.

TABLE I

NOTATIONS USED IN MODELING

notations	meaning
Y	d-dimensional random variable representing the data
$\mathbf{y}$	d-dimensional real variable representing the sample of Y
$\mathbf{y}_n$	the n-th observation
Φ	d-dimensional latent variable representing feature relevance
Φ_n	d-dimensional latent variable for the n-th data
Θ	model parameters
$\mathbf{z}_n$	the n-th latent variable for cluster assignment
$\mathbf{u}_n$	the n-th d-dimensional latent variable for relevant feature
$\mathbf{v}_n$	the n-th d-dimensional latent variable for irrelevant feature
$\beta_{j\ell}$	feature saliency for the ℓ-th feature at the j-th cluster
$\mu_{j\ell}$	mean of the ℓ-th relevant feature at the j-th cluster
$\chi_{j\ell}$	mean of the ℓ-th irrelevant feature at the j-th cluster
$\tau_{j\ell}$	variance of the ℓ-th irrelevant feature at the j-th cluster
π	parameter of the prior for $\mathbf{z}_n$
κ_1, κ_2	parameters of the prior $p(\beta_{j\ell})$ and $p(\tau_{j\ell})$
η_0, ω_0	parameters of the prior $p(\sigma_{j\ell})$
m_0, γ_0	parameters of the prior $p(\mu_{j\ell})$ and $p(\chi_{j\ell})$
α_0	parameters of the prior $p(\pi)$

To handle outliers, heavy-tailed probability distributions, such as Student's t distribution [41] or Pearson type-VII distribution [40] can be used. Taking the features' relevancy into consideration in the Student's t distribution, we result in the following model:

$$p(\mathbf{y}|\Phi;\Theta) = \prod_{\ell=1}^{d} [S_t(y_\ell|\theta_\ell)]^{\phi_\ell} [S_t(y_\ell|\gamma_\ell)]^{1-\phi_\ell} \tag{1}$$

where S_t represents the Student's t distribution.

To realize clustering, a finite mixture of $p(\mathbf{y}|\Phi;\Theta)$ can be applied. That is

$$p(\mathbf{y}|\Phi;\Theta) = \sum_{j=1}^{K} \pi_j \, p(\mathbf{y}|\Phi,\Theta_j)$$

where $\Theta = \{\Theta_j\}$ and $\Theta_j = \{\theta_{j\ell}, \gamma_{j\ell}, 1 \le \ell \le d\}$ are the parameters of the cluster components. To this end, we can introduce a discrete latent variable $\mathbf{z}$ to specify which cluster that the data belongs to, and a Bernoulli prior over Φ with parameter β to characterize the importance of features. To account for the case that in different clusters, features might have different relevance, we propose to impose that Φ depends on the latent variable $\mathbf{z}$. As a result, $\beta_{j\ell}, 1 \le j \le K, 1 \le \ell \le d$ are the parameters associated with the Bernoulli prior over Φ depending on $\mathbf{z}$, which are called feature saliency [37]. Mathematically, the model can be written hierarchically for a set of training data $\mathbf{y}_n, 1 \le n \le N$ as follows:

$$\begin{aligned}
&p(\mathbf{y}_n|\Phi_n, \mathbf{z}_n) \\
&\quad = \prod_{j=1}^{K} \left[\prod_{\ell=1}^{d} [S_t(y_{n\ell}|\theta_{j\ell})]^{\phi_{n\ell}} [S_t(y_{n\ell}|\gamma_{j\ell})]^{1-\phi_{n\ell}} \right]^{\delta_{\mathbf{z}_n,j}} \\
&p(\Phi_n|\mathbf{z}_n, \beta) \\
&\quad = \prod_{j=1}^{K} \left[\prod_{\ell=1}^{d} \beta_{j\ell}^{\phi_{n\ell}} (1-\beta_{j\ell})^{1-\phi_{n\ell}} \right]^{\delta_{\mathbf{z}_n,j}}
\end{aligned}$$

where Φ_n and $\mathbf{z}_n$ are latent variables associated with each data point $\mathbf{y}_n$, and $\delta_{\mathbf{z}_n,j}$ is the Kronecker delta function. Note that a similar idea has been implemented in [38] which is termed as "localized feature saliency." The difference between their work and our work is that we impose dependencies between Φ_n to $\mathbf{z}_n$ and $\mathbf{y}_n$, while in [38], the dependence is implemented by introducing different feature saliency variables in different classes (which results in $\phi_{nj\ell}$ for $1 \le j \le K$ rather than just $\phi_{n\ell}$ as in our implementation). Note that the Student's t distribution can be written as a convolution of a Gaussian and a gamma distribution as follows:

$$S_t(y|\theta) = \int \mathcal{N}(y|\mu, \sigma u) \, \mathcal{G}\left(u|\frac{\nu}{2}, \frac{\nu}{2}\right) du$$

where σ is the precision (inverse variance) and $\theta = (\mu, \sigma, \nu)$ is the parameters, and

$$\mathcal{G}(x|a,b) = b^a x^{a-1} \frac{\exp(-bx)}{\Gamma(a)}.$$

If we introduce $\mathbf{u}_n = (u_{n1}, \ldots, u_{nd})$ and $\mathbf{v}_n = (v_{n1}, \ldots, v_{nd})$ as latent variables for the Student's t components with and without relevant features, respectively, we can obtain a distribution of $p(\mathbf{y}_n|\Phi_n, \mathbf{u}_n, \mathbf{v}_n, \mathbf{z}_n)$ as follows:

$$\begin{aligned}
&p(\mathbf{y}_n|\Phi_n, \mathbf{u}_n, \mathbf{v}_n, \mathbf{z}_n) \\
&\quad = \prod_{j=1}^{K} \left[\prod_{\ell=1}^{d} \mathcal{N}(y_{n\ell}|\mu_{j\ell}, u_{n\ell}\sigma_{j\ell})^{\phi_{n\ell}} \right. \\
&\qquad \left. \times \, \mathcal{N}(y_{n\ell}|\chi_{j\ell}, v_{n\ell}\tau_{j\ell})^{1-\phi_{n\ell}} \right]^{\delta_{\mathbf{z}_n,j}}.
\end{aligned}$$

The hierarchical latent variable model is completed by introducing conjugate prior over $\mathbf{z}_n$, $\mathbf{u}_n$ and $\mathbf{v}_n$ as follows:

$$\begin{aligned}
p(\mathbf{u}_n|\mathbf{z}_n) &= \prod_{j=1}^{K} \left[\prod_{\ell=1}^{d} \mathcal{G}\left(u_{n\ell} \middle| \frac{\nu_{j\ell}}{2}, \frac{\nu_{j\ell}}{2}\right) \right]^{\delta_{\mathbf{z}_n,j}} \\
p(\mathbf{v}_n|\mathbf{z}_n) &= \prod_{j=1}^{K} \left[\prod_{\ell=1}^{d} \mathcal{G}\left(v_{n\ell} \middle| \frac{\gamma_{j\ell}}{2}, \frac{\gamma_{j\ell}}{2}\right) \right]^{\delta_{\mathbf{z}_n,j}} \\
p(\mathbf{z}_n) &= \prod_{j=1}^{K} \pi_j^{\delta_{\mathbf{z}_n,j}}.
\end{aligned}$$

To realize model selection, i.e., selecting the optimal number of components, we adopt the full Bayesian treatment, which means that we need to specify conjugate priors for the parameters (i.e., Θ). The conjugate priors associated with the model parameters are as follows:

$$\begin{aligned}
p(\beta) &= \prod_{j=1}^{K} \prod_{\ell=1}^{d} \mathcal{B}(\beta_{j\ell}|\kappa_1, \kappa_2) \\
p(\sigma) &= \prod_j \prod_\ell p(\sigma_{j\ell}) = \prod_j \prod_\ell \mathcal{G}\left(\sigma_{j\ell} \middle| \frac{\eta_0}{2}, \frac{\omega_0}{2}\right) \\
p(\mu) &= \prod_j \prod_\ell p(\mu_{j\ell}) = \prod_j \prod_\ell \mathcal{N}(\mu_{j\ell}|m_0, \lambda_0) \\
p(\chi) &= \prod_j \prod_\ell p(\chi_{j\ell}) = \prod_j \prod_\ell \mathcal{N}(\chi_{j\ell}|m_0, \lambda_0) \\
p(\tau) &= \prod_j \prod_\ell p(\tau_{j\ell}) = \prod_j \prod_\ell \mathcal{G}\left(\tau_{j\ell} \middle| \frac{\eta_0}{2}, \frac{\omega_0}{2}\right) \\
p(\pi) &= \mathcal{D}(\pi|\alpha_0)
\end{aligned} \tag{2}$$

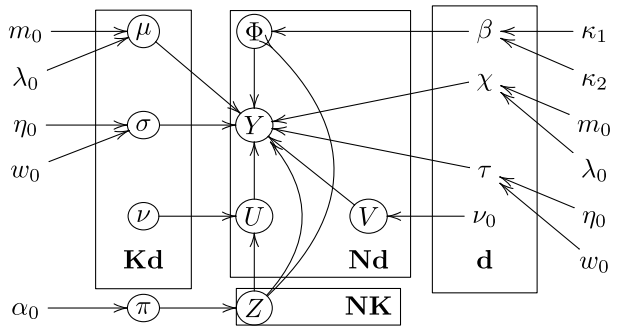


Fig. 2. Plate diagram of the proposed hierarchical graphical model.

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

$$\mathcal{B}(x|a,b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)}$$

and $B(a,b)$ is the beta function, $\mathcal{G}(x|a,b)$ is the gamma distribution, and

$$\mathcal{D}(\pi|\alpha_0) = \frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k^0\right)}{\prod_{k=1}^{K}\Gamma\left(\alpha_k^0\right)}\prod_{k=1}^{K}\pi_k^{\alpha_k^0-1}$$

is the Dirichlet distribution. The parameters in the priors, including $\kappa_1, \kappa_2, \eta_0, \omega_0, m_0, \lambda_0$, and α_0 are considered as hyperparameters. Note that in the priors, we assume the same hyperparameters for $\sigma_{j\ell}$ and $\tau_{j\ell}$ and for $\mu_{j\ell}$ and $\chi_{j\ell}$, respectively. The resultant model can be depicted using the plate diagram shown in Fig. 2. In a rectangle of the plate diagram, the bold typeset indicates the dimensions of the circled variables. For example, **Kd** means that there are $K \times d$ variables of $\nu_{j\ell}, 1 \le j \le K, 1 \le \ell \le N$. The arrows in the diagram indicate the variable dependencies, e.g., the arrow pointing to U from Z means that U depends on Z.

In the following, we use n, ℓ, and j to denote the index of the data point, the features, and the mixing component. We omit the typeset of parameters in the formula.

In the proposed model, bear in mind that the joint probability distribution is written as

$$p(\mathbf{y}_n, \mathbf{u}_n, \mathbf{v}_n, \Phi_n, \mathbf{z}_n|\Theta)$$

where $\Theta = \{\mu, \sigma, \chi, \tau, \pi, \beta, \nu, \gamma\}$ and it can be factorized as

$$p(\mathbf{y}_n|\Phi_n, \mathbf{u}_n, \mathbf{v}_n, \mathbf{z}_n)p(\Phi_n|\mathbf{z}_n)p(\mathbf{u}_n|\mathbf{z}_n)p(\mathbf{v}_n|\mathbf{z}_n)p(\mathbf{z}_n)$$

and are fully factorized over the dimensions. In the sequel, we denote the latent variables as $\mathbf{h}_n = \{\mathbf{u}_n, \mathbf{v}_n, \mathbf{z}_n, \Phi_n, 1 \le n \le N\}$. According to the model, the complete likelihood of a data $\mathbf{y}_n$ can be written as follows:

$$L_{\mathcal{C}}(\mathbf{y}_n, \mathbf{h}_n, \Theta) = p(\mathbf{y}_n, \mathbf{h}_n|\Theta)p(\Theta) \tag{3}$$

where $p(\Theta) = p(\mu)p(\sigma)p(\beta)p(\pi)p(\chi)p(\tau)$. Note that we assume the same hyperparameters of the prior distributions corresponding to the parameters with respect to all the components. We do not assume any priors for ν and γ, since there are no conjugate priors.

TABLE II
NOTATIONS USED IN THE INFERENCE

notations	meaning
$\langle\cdot\rangle_q$	the expectation with respect to q
$\mathcal{F}$	the log-likelihood bound (i.e. the free energy)
$L_{\mathcal{C}}$	the complete log-likelihood
$\psi(x)$	the digamma function (i.e. $\frac{d}{dx}\ln\Gamma(x)$)
$\langle\phi_{n\ell}\rangle_j^0$	$q(\phi_{n\ell}=0\|j)$
$\langle\phi_{n\ell}\rangle_j^1$	$q(\phi_{n\ell}=1\|j)$
$\langle u_{n\ell}\rangle_j$	the expectation of $q(u_{n\ell}\|j)$
$\langle v_{n\ell}\rangle_j$	the expectation of $q(v_{n\ell}\|j)$
$\langle\mathbf{z}_n\rangle_j$	$q(\mathbf{z}_n=j)$
$\bar{a}_{nj\ell}, \bar{b}_{nj\ell}$	parameters of $q(u_{n\ell}\|j)$
$\bar{s}_{nj\ell}, \bar{t}_{nj\ell}$	parameters of $q(v_{n\ell}\|j)$
$\hat{\alpha}$	parameter of $q(\pi)$
$\bar{\kappa}_{1j\ell}, \bar{\kappa}_{2j\ell}$	parameters of $q(\beta_{j\ell})$
$\bar{\eta}_{j\ell}, \bar{\omega}_{j\ell}$	parameters of $q(\sigma_{j\ell})$
$\psi_{j\ell}, \xi_{j\ell}$	parameters of $q(\tau_{j\ell})$
$\bar{\sigma}_{j\ell}, \bar{\mu}_{j\ell}$	parameters of $q(\mu_{j\ell})$
$\bar{\varrho}_{j\ell}, \bar{\varsigma}_{j\ell}$	parameters of $q(\chi_{j\ell})$

III. INFERENCE

In this section, we first define some notations as listed in Table II. These notations will be used in the inference. A brief introduction to the VB method is given, while the detailed inference follows. The algorithm is then summarized and interpreted.

A. Brief Introduction to VB

The integration of $p(\mathbf{y}_n, \mathbf{u}_n, \mathbf{v}_n, \Phi_n, \mathbf{z}_n|\Theta)p(\Theta)$ over the latent variables and the parameters is not tractable. Therefore, exact inference is impossible. We adopt the VB algorithm for model inference [42]. To apply the VB algorithm, the evidence, obtained by integrating out the latent variables (denoted by $\mathcal{H}$) and the parameters (denoted by Θ) given a model structure $\mathcal{M}$, is approximated by introducing an auxiliary distribution q. The lower bound to the evidence is as follows:

$$\begin{aligned}\log p(\mathbf{X}|\mathcal{M}) &\ge \int_{\Theta}\int_{\mathcal{H}} q(\mathcal{H},\Theta)\log\frac{p(\mathbf{Y},\mathcal{H},\Theta|\mathcal{M})}{q(\mathcal{H},\Theta)}d\mathcal{H}d\Theta \\ &= \langle\log p(\mathbf{Y},\mathcal{H},\Theta|\mathcal{M})\rangle_q - \langle\log q(\mathcal{H},\Theta)\rangle_q \\ &\triangleq \mathcal{F}(q(\mathcal{H}), q(\Theta), \mathbf{Y}, \mathcal{M})\end{aligned} \tag{4}$$

where $p(\mathbf{Y},\mathcal{H},\Theta|\mathcal{M})$ is the complete data likelihood and $q(\mathcal{H},\Theta) = q(\mathcal{H})q(\Theta)$ is the auxiliary posterior distribution, $\mathcal{F}(q(\mathcal{H}), q(\Theta), \mathbf{Y}, \mathcal{M})$ is called the free energy. In the equations, we use $\langle\cdot\rangle_q$ to denote the expectation with respect to q. It is obvious that maximizing the evidence is equivalent to maximizing the free energy $\mathcal{F}$.

To maximize the free energy, we apply coordinate ascent search as adopted in [43]. Applying the coordinate ascent search, the auxiliary distributions of latent variable $\mathcal{H}$ and parameters Θ are optimized alternatively as follows:

$$\begin{aligned}q^{(t+1)}(\mathcal{H}) &= \arg\max_{q(\mathcal{H})}\mathcal{F}(q(\mathcal{H}), q^t(\Theta), \mathbf{Y}, \mathcal{M}) \\ q^{(t+1)}(\Theta) &= \arg\max_{q(\Theta)}\mathcal{F}(q^{t+1}(\mathcal{H}), q(\Theta), \mathbf{Y}, \mathcal{M}).\end{aligned}$$

B. Tree-Like Factorization of the Random Variables

We propose a tree-like factorization over the latent variables for the auxiliary posteriors (i.e., $q(\mathcal{H})$). Tree-like structural factorization in VB has been shown to be superior over the full factorization scheme [44], [45]. The factorization can be summarized as follows:

$$\begin{aligned} &q(\mathbf{h}_n, \pi, \{\beta_{j\ell}\}, \{\mu_{j\ell}, \sigma_{j\ell}\}, \{\chi_{j\ell}, \tau_{j\ell}\}) \\ &\quad = \underbrace{q(\mathbf{u}_n|\mathbf{z}_n) q(\Phi_n|\mathbf{z}_n) q(\mathbf{v}_n|z_n) q(\mathbf{z}_n)} \\ &\qquad \times q(\pi) q(\{\beta_{j\ell}\}) q\left(\{\mu_{j\ell}, \sigma_{j\ell}\}\right) q\left(\{\chi_{j\ell}, \tau_{j\ell}\}\right). \end{aligned}$$

The tree-like factorization is reflected on the dependences between $\mathbf{u}_n, \mathbf{v}_n, \Phi_n$, and $\mathbf{z}_n$. Specifically, due to the full factorization over the features and the conjugate prior we used, it can be seen that

$$\begin{aligned} q(\mathbf{h}_n, \Theta) = q(\mathbf{z}_n) &\prod_n \prod_\ell q(v_{n\ell}|\mathbf{z}_n) q(u_{n\ell}|\mathbf{z}_n) q(\phi_{n\ell}|\mathbf{z}_n) \\ &\times q(\pi) \prod_j \prod_\ell q(\chi_{j\ell}) q(\tau_{j\ell}) q(\beta_{j\ell}) q(\mu_{j\ell}) q(\sigma_{j\ell}). \end{aligned}$$

The auxiliary posteriors of the latent variables and the parameters can be obtained by maximizing the free energy associated with the proposed model

$$\mathcal{F} = \langle \log L_C(\mathbf{y}_n, \mathbf{h}_n, \Theta) \rangle_q - \langle \log q(\mathbf{h}_n) \rangle_q \tag{5}$$

where $\langle \cdot \rangle_q$ is the expectation with respect to the auxiliary posterior q.

C. Auxiliary Posteriors of the Latent Variables

The free energy associated with the auxiliary posterior $q(\mathbf{u}_n|\mathbf{z}_n)$ can be read as follows:

$$\mathcal{F} = \langle \log[p(\mathbf{y}_n, h_n)] - \log q(\mathbf{u}_n|\mathbf{z}_n) \rangle_q.$$

According to the KKT condition, and using the Lagrange multiplier, we obtain (see the Appendix for details)

$$q(\mathbf{u}_n \mathbf{z}_n) \propto \prod_{\ell=1}^{d} \exp\langle \log[p(y_{n\ell}|u_{n\ell}, \mathbf{z}_n) p(u_{n\ell}|\mathbf{z}_n)] \rangle_q.$$

This shows that $q(\mathbf{u}_n|\mathbf{z}_n) = \prod_{\ell=1}^{d} q(u_{n\ell}|\mathbf{z}_n)$. Through mathematical manipulation, we can obtain

$$q(u_{n\ell}|\mathbf{z}_n = j) = \mathcal{G}(u_{n\ell}|\bar{a}_{nj\ell}, \bar{b}_{nj\ell}) \tag{6}$$

where

$$\bar{a}_{nj\ell} = \frac{v_{j\ell} + 1}{2}; \quad \bar{b}_{nj\ell} = \frac{v_{j\ell} + \langle (y_{n\ell} - \mu_{j\ell})^2 \sigma_{j\ell} \rangle}{2}.$$

Similarly to the above calculation, the other posteriors can be computed. We find that the posterior of the latent variable $v_{n\ell}$, i.e., $q(v_{n\ell}|\mathbf{z}_n)$, is of the following form:

$$q(v_{n\ell}) = \mathcal{G}(v_{n\ell}|\bar{s}_{nj\ell}, \bar{t}_{nj\ell}) \tag{7}$$

where

$$\bar{s}_{nj\ell} = \frac{\gamma_{j\ell} + 1}{2}; \quad \bar{t}_{n\ell} = \frac{\gamma_{j\ell} + \langle (y_{n\ell} - \chi_{j\ell})^2 \tau_{j\ell} \rangle}{2}.$$

Note that $\langle (y_{n\ell} - \mu_{j\ell})^2 \rangle = (y_{n\ell} - \langle \mu_{j\ell} \rangle)^2 + \bar{\sigma}_{j\ell}$ and $\langle (y_{n\ell} - \chi_{j\ell})^2 \rangle = (y_{n\ell} - \langle \chi_{j\ell} \rangle)^2 + \bar{\varsigma}_{j\ell}$, where $\bar{\sigma}_{j\ell}$ and $\bar{\varsigma}_{j\ell}$ are the standard deviations of the posterior $q(\mu_{j\ell})$ and $q(\chi_{j\ell})$, respectively. If we let

$$\begin{aligned} A = &[\langle \log p(y_{n\ell}|u_{n\ell}, j) \rangle + \langle \log p(u_{n\ell}|j) \rangle] \\ &+ \langle \log \beta_{j\ell} \rangle - \langle \log q(u_{n\ell}|j) \rangle \end{aligned}$$

and

$$\begin{aligned} B = &[\langle \log p(y_{n\ell}|v_{n\ell}, j) \rangle + \langle \log p(v_{n\ell}|j) \rangle] \\ &+ \langle \log(1 - \beta_{j\ell}) \rangle - \langle \log q(v_{n\ell}|j) \rangle \end{aligned}$$

then $q(\phi_{n\ell} = 1|j)$ can be written as

$$q(\phi_{n\ell} = 1|j) = \frac{\exp\{A\}}{\exp\{A\} + \exp\{B\}} \tag{8}$$

and $q(\phi_{n\ell} = 0|j) = 1 - q(\phi_{n\ell} = 1|j)$.

If we define the quantity

$$\begin{aligned} R_{n,j} = &\sum_\ell \left(\langle \phi_{n\ell} \rangle_j^1 \langle \log p(y_{n\ell}|u_{n\ell}, j) \rangle \right) \\ &+ \sum_\ell \left(\langle \phi_{n\ell} \rangle_j^0 \langle \log p(y_{n\ell}|v_{n\ell}, j) \rangle \right) + \langle \log \pi_j \rangle \\ &+ \sum_\ell \left(\langle \phi_{n\ell} \rangle_j^1 \log p(u_{n\ell}|j) \rangle + \langle \phi_{n\ell} \rangle_j^0 \log p(v_{n\ell}|j) \right) \\ &+ \sum_\ell \left(\langle \phi_{n\ell} \rangle_j^1 \langle \log \beta_{j\ell} \rangle + \langle \phi_{n\ell} \rangle_j^0 \langle \log(1 - \beta_{j\ell}) \rangle \right) \\ &- \sum_\ell \left(\langle \phi_{n\ell} \rangle_j^1 \langle \log q(u_{n\ell}|j) \rangle + \langle \phi_{n\ell} \rangle_j^0 \langle \log q(v_{n\ell}|j) \rangle \right). \end{aligned}$$

Then, the responsibility $q(\mathbf{z}_n = j)$ can be calculated as follows:

$$q(\mathbf{z}_n = j) = \frac{\exp\{R_{n,j}\}}{\sum_k \exp\{R_{n,k}\}}. \tag{9}$$

In the sequel, we use $\langle \mathbf{z}_n \rangle_j$ to denote $q(\mathbf{z}_n = j)$.

D. Auxiliary Posteriors of the Parameters

The posterior of the mixing proportion π is

$$q(\pi) = \mathcal{D}(\pi|\hat{\alpha}) \tag{10}$$

where $\hat{\alpha}_j = \sum_n q(\mathbf{z}_n = j) + \alpha_0$ and $\hat{\alpha}_0 = \sum_j \hat{\alpha}_j$ and

$$\langle \log \pi_j \rangle = \Psi(\hat{\alpha}_j) - \Psi(\hat{\alpha}_0).$$

The posterior of the feature saliency β is

$$q(\beta) = \prod_j \prod_\ell q(\beta_\ell) = \prod_j \prod_\ell \mathcal{B}(\beta_{j\ell}|\bar{\kappa}_{1j\ell}, \bar{\kappa}_{2j\ell}) \tag{11}$$

where $\bar{\kappa}_{1j\ell} = \kappa_1 + \sum_n \langle \phi_{n\ell} \rangle_j^1 \langle \mathbf{z}_n \rangle_j$ and $\bar{\kappa}_{2j\ell} = \kappa_2 + \sum_n \langle \phi_{n\ell} \rangle_j^0 \langle \mathbf{z}_n \rangle_j$. The expectation $\langle \log \beta_{j\ell} \rangle$ and $\langle \log(1 - \beta_{j\ell}) \rangle$ as used in the calculation of $q(\Phi_n|j)$ can be obtained as

$$\begin{aligned} \langle \log \beta_{j\ell} \rangle &= \psi(\bar{\kappa}_{1j\ell}) - \psi(\bar{\kappa}_{1j\ell} + \bar{\kappa}_{2j\ell}) \\ \langle \log(1 - \beta_{j\ell}) \rangle &= \psi(\bar{\kappa}_{2j\ell}) - \psi(\bar{\kappa}_{1j\ell} + \bar{\kappa}_{2j\ell}). \end{aligned}$$

The posterior of variance σ_j is

$$q(\sigma_j) = \prod_\ell q(\sigma_{j\ell}) = \prod_\ell \mathcal{G}(\sigma_{j\ell}|\bar{\eta}_{j\ell}, \bar{\omega}_{j\ell}) \tag{12}$$

where

$$\bar{\eta}_{j\ell} = \frac{\eta_0 + \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^1}{2}$$
$$\bar{\omega}_{j\ell} = \frac{\omega_0 + \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^1 \langle (y_{n\ell} - \mu_{j\ell})^2 \rangle \langle u_{n\ell} \rangle_j}{2}.$$

The posterior of variance of the common distribution τ is

$$q(\tau) = \prod_j \prod_\ell q(\tau_{j\ell}) = \prod_\ell \mathcal{G}(\tau_{j\ell} | \bar{\psi}_{j\ell}, \bar{\xi}_{j\ell}) \tag{13}$$

where

$$\bar{\psi}_{j\ell} = \frac{\eta_0 + \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^0}{2}$$
$$\bar{\xi}_{j\ell} = \frac{\omega_0 + \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^0 \langle (y_{n\ell} - \chi_{j\ell})^2 \rangle \langle v_{n\ell} \rangle_j}{2}.$$

The posterior of μ_j is

$$q(\mu_j) = \prod_\ell q(\mu_{j\ell}) = \prod_\ell \mathcal{N}(\mu_{j\ell} | \bar{\mu}_{j\ell}, \bar{\sigma}_{j\ell}) \tag{14}$$

where

$$\bar{\sigma}_{j\ell} = \langle \sigma_{j\ell} \rangle \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^1 \langle u_{n\ell} \rangle_j + \lambda_0$$
$$\bar{\mu}_{j\ell} = \bar{\sigma}_{j\ell}^{-1} \left(\langle \sigma_{j\ell} \rangle \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^1 \langle u_{n\ell} \rangle_j y_{n\ell} + \lambda_0 \mu_0 \right).$$

The posterior of χ is

$$q(\chi) = \prod_\ell \prod_j q(\chi_{j\ell}) = \prod_\ell \prod_j \mathcal{N}(\chi_{j\ell} | \bar{\varrho}_{j\ell}, \bar{\varsigma}_{j\ell}) \tag{15}$$

where

$$\bar{\varsigma}_{j\ell} = \langle \tau_{j\ell} \rangle \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^0 \langle v_{n\ell} \rangle_j + \lambda_0$$
$$\bar{\varrho}_{j\ell} = \bar{\varsigma}_{j\ell}^{-1} \left(\langle \tau_{j\ell} \rangle \sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^0 \langle v_{n\ell} \rangle_j y_{n\ell} + \lambda_0 \mu_0 \right).$$

The degree of freedom $v_{j\ell}$, $1 \le j \le d$, $\gamma_{j\ell}$, $1 \le \ell \le d$ can be obtained by solving the following nonlinear equations, where $\langle \log v_{n\ell} \rangle_j$ and $\langle \log u_{n\ell} \rangle_j$ denote the expectations of $\log q(v_{n\ell}|j)$ and $\log q(u_{n\ell}|j)$, respectively:

$$\sum_n \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^1 \times \left[1 + \log \frac{v_{j\ell}}{2} + \langle \log u_{n\ell} \rangle_j - \langle u_{n\ell} \rangle_j - \psi \left(\frac{v_{j\ell}}{2} \right) \right] = 0$$
$$\sum_{n,j} \langle \mathbf{z}_n \rangle_j \langle \phi_{n\ell} \rangle_j^0 \times \left[\langle \log v_{n\ell} - v_{n\ell} \rangle_j + 1 + \log \frac{\gamma_{j\ell}}{2} - \psi \left(\frac{\gamma_{j\ell}}{2} \right) \right] = 0$$

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

Algorithm 1 Proposed Tree-Like VB Algorithm for Clustering, Feature Selection, and Outlier Detection

Require: training data $\mathbf{y}_n$, $1 \le n \le N$, a cluster number K;
Ensure: the centroids, the saliency of the features and the outlier criteria;

```
while the free energy F increases less than ε do
    VB E-step
    Update q(u_n|z_n) according to (6)
    Update q(v_n) according to (7)
    Update q(Φ_n|z_n) according to (8)
    Update q(z_n) according to (9)
    VB M-Step
    Update q(π) according to (10)
    Update q(β) according to (11)
    Update q(σ_j), 1 ≤ j ≤ K according to (12)
    Update q(τ) according to (13)
    Update q(μ_j), 1 ≤ j ≤ K according to (14)
    Update q(ξ) according to (15)
    Calculate the log-likelihood bound using (16)
end while
```

E. Log-Likelihood Bound

The optimization process can be monitored by the log-likelihood bound as shown in (5), which can be evaluated in the following. The evaluation of the expectations of the log-likelihood bound (i.e., the free energy) is summarized in the Appendix

$$\begin{aligned}
\mathcal{F} = & \sum_{n,j} \langle \mathbf{z}_n \rangle_j \sum_\ell \langle \phi_{n\ell} \rangle_j^1 \langle \log [p(y_{n\ell}|u_{n\ell}, j) p(u_{n\ell}|j)] \rangle \\
& + \sum_{n,j} \langle \mathbf{z}_n \rangle_j \sum_\ell \langle \phi_{n\ell} \rangle_j^0 \langle \log [p(y_{n\ell}|v_{n\ell}, j) p(v_{n\ell}|j)] \rangle \\
& + \sum_{n,j} \langle \mathbf{z}_n \rangle_j \sum_\ell \langle \log p(\phi_{n\ell}|\beta_{j\ell}) \rangle_j + \sum_{n,j} \langle \mathbf{z}_n \rangle_j \langle \log \pi_j \rangle \\
& + \sum_j \langle \log p(\mu_j) + \log p(\sigma_j) - \log q(\mu_j) - \log q(\sigma_j) \rangle \\
& + \langle \log p(\chi) + \log p(\tau) - \log q(\chi) - \log q(\tau) \rangle \\
& + \langle \log p(\pi) - \log q(\pi) \rangle + \langle \log p(\beta) - \log q(\beta) \rangle \\
& - \sum_{n,j} \langle \mathbf{z}_n \rangle_j \sum_\ell \langle \log q(u_{n\ell}|j) \rangle_j - \sum_{n\ell} \langle \log q(v_{n\ell}|j) \rangle \\
& - \sum_{n,j} \langle \mathbf{z}_n \rangle_j \sum_\ell \langle \log q(\phi_{n\ell}|j) \rangle_j - \sum_{nj} \langle \mathbf{z_n} \rangle_j \log \langle \mathbf{z_n} \rangle_j .
\end{aligned} \tag{16}$$

F. Algorithm

The developed VB algorithm can be summarized in Algorithm 1. To start the run, in the beginning, a large number of clusters K are given. The K-mean clustering is carried out, while the resulting centroids are used as the initial value for $q(\mu)$. Note also that the adopted Bayesian framework allows us to realize model selection, i.e., to find the optimal number of clusters. Initializing a large K cluster number, some clusters that do not have enough evidence will be pruned during the optimization process. The automatic pruning can be observed in the demo, as shown in Fig. 3.

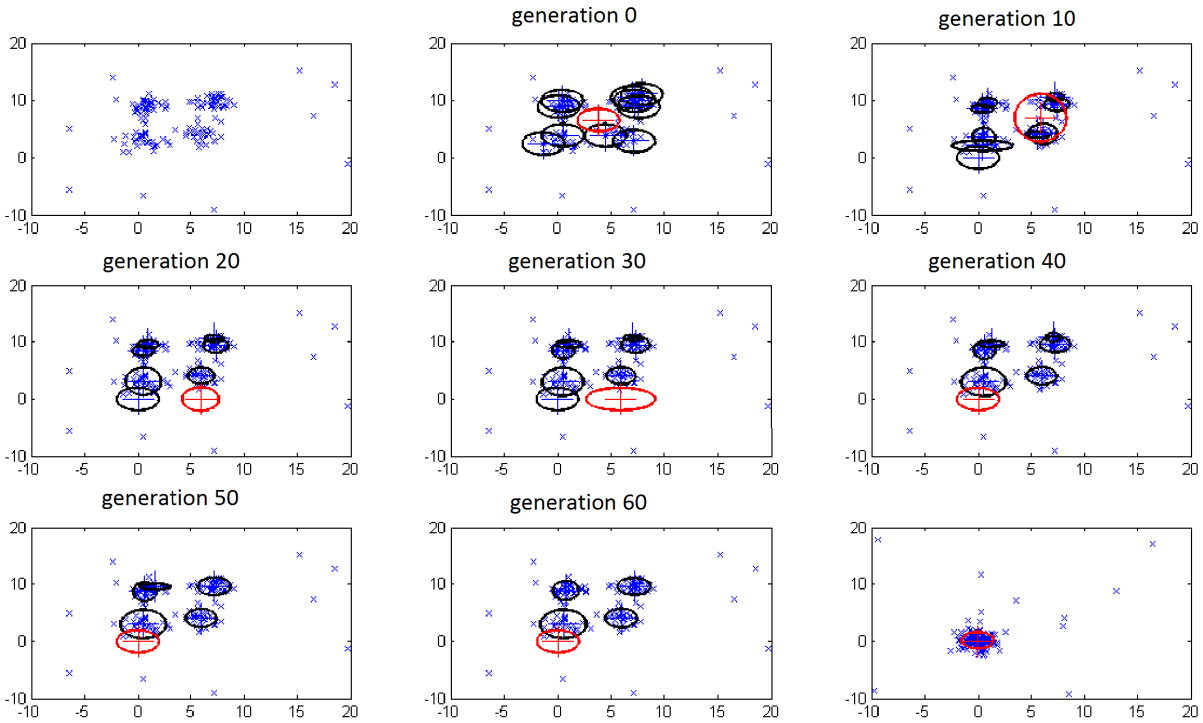


Fig. 3. Typical run of the developed algorithm on the example data set, while the black circles represent $q(\mu_1|k)$, and the red circles denote $q(\mu_0)$. The first plot shows the data set on the first two dimensions, while the last plot shows the estimation of the third and fourth dimensions.

Since the VB algorithm is proven to be monotonically increasing, it is thus able to terminate the algorithm if there is a small difference (ϵ in line 1) between consecutive iterations. In our implementation, we set $\epsilon = 1.0^{-7}$.

G. Interpreting the Model

Considering the time complexity of the algorithm, per iteration, computing the parameters of the posteriors of $\mathbf{u}_n$, $\mathbf{z}_n$, and Φ_n are $\mathcal{O}(NdK)$, while for $q(\mathbf{v}_n|\mathbf{z}_n)$, the time complexity is $\mathcal{O}(Nd)$. Therefore, the total time complexity is of $\mathcal{O}(NKd)$.

As claimed, the proposed model is supposed to deal with outliers, and to find most informative features. To detect outliers, the weighted expectation of the posteriors of $\mathbf{u_n}$ and $\mathbf{v}_n$ can be used as the outlier criterion. That is, if we define

$$c_n = \sum_j \langle \mathbf{z}_n \rangle_j \sum_\ell \left[\langle \phi_{n\ell} \rangle_j^1 \frac{\bar{a}_{njl}}{\bar{b}_{njl}} + \langle \phi_{n\ell} \rangle_j^0 \frac{\bar{s}_{njl}}{\bar{t}_{njl}} \right]$$

then the smaller the value of c_n with respect to $\mathbf{y}_n$, the higher chance that the datum is an outlier.

As stated in the model, the expectation of the feature saliency variable $\beta_{j\ell}, 1 \leq \ell \leq d$ can be applied to show the informative degree of the features for each cluster, which can be obtained as follows:

$$\langle \beta_{j\ell} \rangle = \frac{\bar{\kappa}_{1j\ell}}{\bar{\kappa}_{1j\ell} + \bar{\kappa}_{2j\ell}}.$$

The higher the $\langle \beta_{j\ell} \rangle$ value, the more important of feature ℓ in class j.

For the overall feature saliency, we can use the following quantity to specify:

$$\varsigma_\ell = \sum_j \langle \pi_j \rangle_q \langle \beta_{j\ell} \rangle = \sum_j \frac{\hat{\alpha}_j}{\sum_k \hat{\alpha}_k} \langle \beta_{j\ell} \rangle$$

which is a weighted average over the feature saliency for each cluster. The higher the ς_ℓ value, the more relevant the feature.

IV. Experiments

A. Synthetic Data

In this section, we justify the developed model and the tree-like VB algorithm using controlled experiments. Synthetic data sets are generated that are able to accommodate the data characteristics for the justification. The proposed model and the algorithm were compared with the semi-Bayesian feature selection model and algorithm, called varFnMS [34], in which a finite mixture of Gaussian is adopted and a full-factorized VB is applied.

Synthetic data are generated by first sampling a set of data points from four well-separated bivariate clusters. The centers and the variance–covariance matrices are $[0\ 3]^\intercal$, $[1\ 9]^\intercal$, $[6\ 4]^\intercal$, $[7\ 10]^\intercal$, and an identity matrix. Eight "noisy" features [sampled from $\mathcal{N}(0, 1)$] are then appended to this data, resulting in a 10-D patterns. 800 data points are generated, and a set of outliers uniformly sampled from $[-10\ 30]^{10}$ are added to the data set. Various percentages of outliers are added to the main data sets to test the performance of the algorithm on outlier detection.

The proposed algorithm was carried out for ten times with an initial cluster number $K = 10$. The K-means clustering algorithm is used to initialize the mean of the posterior $q(\mu_j)$,

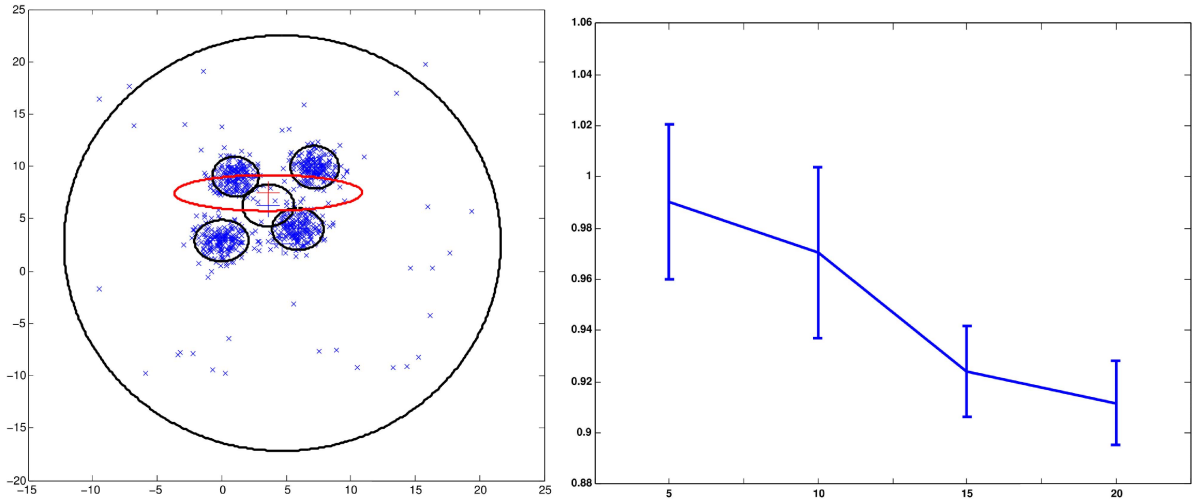


Fig. 4. Left: typical run of the semi-Bayesian feature selection algorithm, and first and second features are shown. Right: AUC values obtained by the developed algorithm and varFnMS for different percentages of outliers with standard deviations shown.

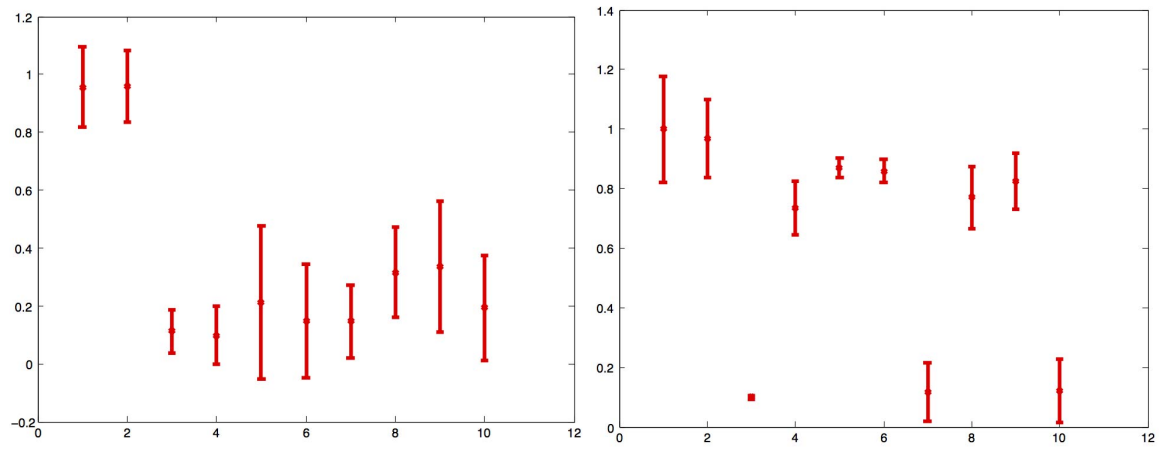


Fig. 5. Feature saliencies for the synthetic data with 5% percentage of outliers by the proposed algorithm (on the left) and the semi-Bayesian algorithm (on the right). The standard deviations of the ten runs were also shown in the plots.

and the feature saliency variable is initialized to be 0.5. The hyperparameters κ_1, κ_2, λ_0, and α_0 are set to be 10^{-5}, and m_0 is set to be the mean of all data. The algorithm terminates when the difference of log-likelihood bound is less than 10^{-7}.

Fig. 3 shows a typical run of the developed algorithm, while the estimated mean and covariance of $q(\mu)$ in the first break two-dimension is shown at certain iterations. From the figure, we can see that the developed algorithm groups the data accurately. Moreover, it can be seen that unnecessary components are pruned automatically during the optimization process. The last plot shows the data in the third and fourth variables. The red circle demonstrates the contour of the posterior $q(\chi)$ at the third and fourth features. Fig. 4(a) shows the results obtained by varFnMS at the first and second features. From the figure, it can be seen that varFnMS is not able to eliminate the effects of the outliers, and the number of clusters has not been estimated accurately.

To test the outlier detection performance, the area under curve (AUC) values obtained through the ROC analysis can be used. The higher the AUC values, the better the performance of outlier detection. Fig. 4(b) shows the obtained AUC values with standard deviation by the developed algorithm for different percentages of outliers in ten runs. Unfortunately, no statistics can be derived by varFnMS for the purpose of outlier detection. From the figure, it can be observed that the developed algorithm is able to pick outliers successfully. This shows that the proposed algorithm is able to simultaneously pickup outliers from the data, and discover clusters accurately.

Fig. 5 shows the feature saliency retrieved by the proposed algorithm and varFnMS. From the figure, we can see that the saliency of the noisy variables $(Y_3 - Y_8)$ obtained by the proposed algorithm is closer to the ground truth than that of the semi-Bayesian algorithm.

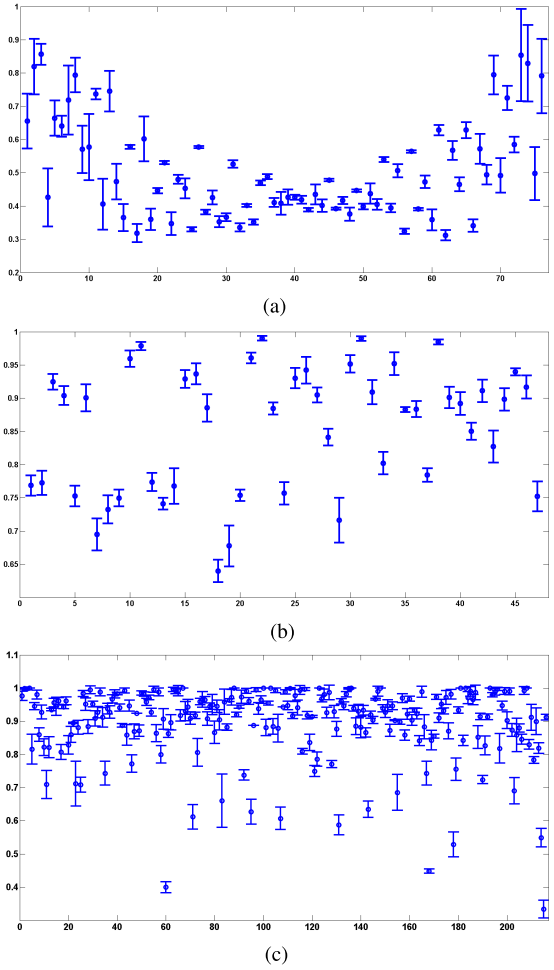
B. Experiments on Real Data Sets

In this section, we used the "multiple feature database" [34], [46], which consists of features of handwritten numerals ("0"–"9") extracted from a collection of Dutch utility maps. There are a total of 2000 images with 200 for each numerals. Numerals are represented in different feature sets. We used the same three data sets as in [34], that is, the Zernike moments (47 features), the Fourier coefficients (76 features),

TABLE III

AVERAGED CLASSIFICATION ERROR AND THE NUMBER OF COMPONENTS OBTAINED BY varFnMS AND THE PROPOSED ALGORITHM USING 30 AND 50 INITIAL COMPONENTS

Alg.		$K = 30$			$K = 50$		
		Zernike	Fourier	Profile	Zernike	Fourier	Profile
FnMS	error	0.53(0.02)	0.50(0.07)	0.77(0.04)	0.53(0.01)	0.52(0.03)	0.76(0.04)
	comp.	10.7(1.2)	6.1(0.9)	2.3(0.7)	10.8(1.1)	5.7(0.7)	2.3(0.4)
varFnMS	error	0.39(0.07)	0.35(0.06)	0.13(0.01)	0.37(0.03)	0.32(0.02)	0.12(0.01)
	comp.	26.8(6.4)	24.6(7.2)	26.3(3.7)	28.1(2.3)	25.0(1.6)	29.6(2.4)
the proposed	error	0.30(0.10)	0.25(0.05)	0.10(0.01)	0.29(0.12	0.230.03)	0.09(0.01)
algorithm	comp.	16.3(5.2)	15.0(4.7)	16.9(4.3)	19.2(3.1)	14.8(2.0)	20.1(3.4)



(a) (b) (c)

Fig. 6. Saliencies of different feature sets. (a) Fourier coefficients, (b) Zernike moments, and (c) profile correlations using the developed algorithm for models initialized with 30 components.

and profile correlations (216 features). The classification error is used to measure the performance. For each data point, it is assigned to the class with the largest responsibility. The proposed algorithm was run 20 times, where the data set is split into half to create the training and test data set. The estimated classification error and the number of components are summarized in Table III, where the components were initialized to be 30 and 50.

From Table III, we can see that our algorithm outperforms varFnMS and FnMS [46] in terms of classification error. It can be seen that the developed algorithm uses less number of components than that of varFnMS, which is closer to the true number of components. This suggests that the developed algorithm performs better in terms of recovering the true parameters. Fig. 6 shows the error bar plots of the saliencies obtained for the proposed model initialized with 30 components. As an comparison, Fig. 7 showed the saliencies by using varFnMS with the same experimental settings as the developed algorithm. From the figure, we can see that on the Fourier coefficient data set and the profile correlation data set, the feature saliency obtained by the developed algorithm has similar trends as that of varFnMS, but of smaller variances. On the Zernike moments data set, we can see that the variances of the feature saliency revealed by the developed algorithm are much less than those obtained by varFnMS. This shows that the developed algorithm is more robust than that of varFnMS.

TABLE IV

CONFUSION MATRIX OBTAINED BY THE DEVELOPED ALGORITHM ON THE LEUKEMIA DATA

	Predicated class					
BCR-ABL	12	0	1	0	1	0
E2A-PBX1	0	27	0	0	0	0
Hyperdiploid > 50	0	0	53	1	10	0
MLL	0	6	0	11	2	1
T-ALL	2	0	3	0	38	0
TEL-AML1	5	0	3	0	0	71

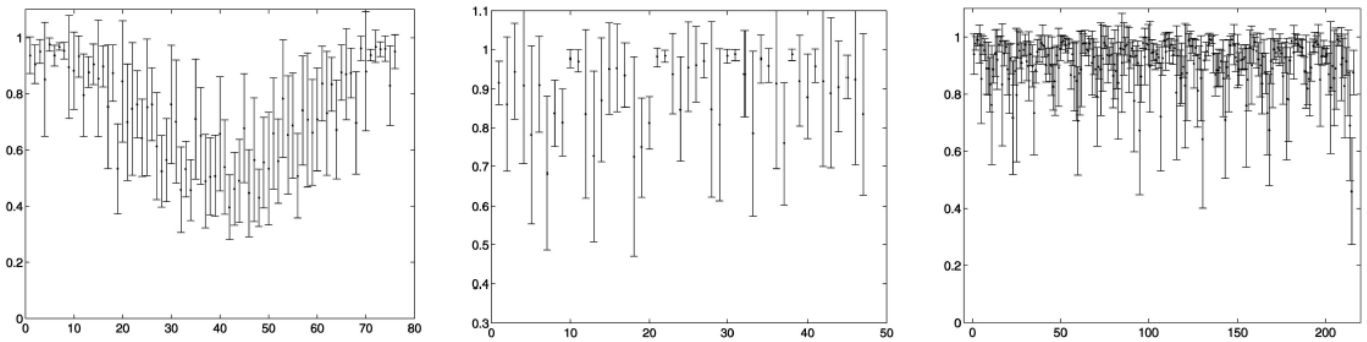
TABLE V

CORRELATION BETWEEN THE STATISTICS OBTAINED IN [47] AND THE FEATURE SALIENCIES WITH RESPECT TO THE LEUKEMIA SUBTYPES

subtypes	corr. coef. (Chi-square χ^2)	corr. coef. (t-statistics)
BCR-ABL	0.613	0.629
E2A-PBX1	0.709	0.812
Hyperdiploid > 50	0.851	0.800
MLL	0.650	0.698
T-ALL	0.713	0.720
TEL-AML1	0.752	0.768

C. Application on High-Dimensional Gene Expression Data

In this section, we apply the developed algorithm to a large-scale gene expression data set on leukemia [47]. The data were obtained through the diagnostic of bone marrow samples from pediatric acute leukemia (ALL) patients corresponding to six prognostically important leukemia subtypes, including 43 T-lineage ALL, 27 E2A-PBX1, 15 BCR-ABL, 79 TEL-AML1, and 20 MLL rearrangements and 64 "hyperdiploid $>$ 50" chromosomes, and containing more than 12 600 probe sets. The resultant data set contains 248 samples, and 12 625 gene expressions. Note that in [47],



(a) (b) (c)

Fig. 7. Saliencies of different feature sets. (a) Fourier coefficients, (b) Zernike moments, and (c) profile correlations using varFnMS for models initialized with 30 components; reproduced from [34].

TABLE VI

EVALUATION OF THE LOG-LIKELIHOOD BOUND

		Expectations of the logarithm of the priors of the latent variables
$\langle \log p(y_{n\ell}\|u_{n\ell}, j)\rangle$	=	$\frac{1}{2}\langle u_{n\ell}\rangle_j \langle \sigma_{n\ell}\rangle_j - \frac{1}{2}\left((y_{n\ell} - \langle \mu_{j\ell}\rangle)^2 + \bar{\sigma}_{j\ell}\right)\langle u_{j\ell}\rangle \langle \sigma_{j\ell}\rangle + const.$
$\langle \log p(u_{n\ell}\|j)\rangle$	=	$\frac{\nu_{j\ell}}{2}\log\frac{\nu_{j\ell}}{2} + \left(\frac{\nu_{j\ell}}{2} - 1\right)\langle \log u_{n\ell}\rangle_j - \frac{\nu_{j\ell}}{2}\langle u_{n\ell}\rangle_j - \log\Gamma\left(\frac{\nu_{j\ell}}{2}\right)$
$\langle \log p(y_{n\ell}\|v_{n\ell}, j)\rangle$	=	$\frac{1}{2}\langle v_{n\ell}\rangle_j \langle \sigma_{n\ell}\rangle - \frac{1}{2}((y_{n\ell} - \langle \chi_{j\ell}\rangle)^2 + \bar{\varsigma}_{j\ell})\langle v_{n\ell}\rangle_j \langle \tau_{j\ell}\rangle + const.$
$\langle \log p(v_{n\ell}\|j)\rangle$	=	$\frac{\gamma_{j\ell}}{2}\log\frac{\gamma_{j\ell}}{2} + \left(\frac{\gamma_{j\ell}}{2} - 1\right)\langle \log v_{n\ell}\rangle_j - \frac{\gamma_{j\ell}}{2}\langle v_{n\ell}\rangle_j - \log\Gamma\left(\frac{\gamma_{j\ell}}{2}\right)$
$\langle \log p(\phi_{n\ell}\|j, \beta_{j\ell})\rangle$	=	$\langle \phi_{n\ell}\rangle_j^1 \langle \log\beta_{j\ell}\rangle + \langle \phi_{n\ell}\rangle_j^0 \langle \log(1-\beta_{j\ell})\rangle$
$\langle \log p(\mathbf{z}_n = j)\rangle$	=	$\langle \log \pi_j\rangle$
		Expectations of the logarithm of the priors of the parameter variables
$\langle \log p(\beta_{j\ell})\rangle$	=	$(\kappa_1 - 1)\langle \log\beta_{j\ell}\rangle + (\kappa_2 - 1)\langle \log(1-\beta_{j\ell})\rangle + const.$
$\langle \log p(\pi_j)\rangle$	=	$\left(\alpha_j^0 - 1\right)\langle \log \pi_j\rangle + const.$
$\langle \log p(\sigma_{j\ell})\rangle$	=	$\frac{\eta_0}{2}\log\frac{\omega_0}{2} + \left(\frac{\eta_0}{2} - 1\right)\langle \log\sigma_{j\ell}\rangle - \frac{\omega_0}{2}\langle \sigma_{j\ell}\rangle - \log\Gamma\left(\frac{\omega_0}{2}\right)$
$\langle \log p(\mu_{j\ell})\rangle$	=	$\frac{1}{2}\log(\lambda_0) - \frac{1}{2}\left((\langle \mu_{j\ell}\rangle - m_0)^2 + \bar{\sigma}_{j\ell}\right)\lambda_0 + const$
$\langle \log p(\chi_{j\ell})\rangle$	=	$\frac{1}{2}\log(\lambda_0) - \frac{1}{2}\left((\langle \chi_{j\ell}\rangle - m_0)^2 + \bar{\varsigma}_{j\ell}\right)\lambda_0 + const$
$\langle \log p(\tau_{j\ell})\rangle$	=	$\frac{\eta_0}{2}\log\frac{\omega_0}{2} + \left(\frac{\eta_0}{2} - 1\right)\langle \log\tau_{j\ell}\rangle - \frac{\omega_0}{2}\langle \tau_{j\ell}\rangle - \log\Gamma\left(\frac{\omega_0}{2}\right)$
		Expectations of the logarithm of the auxiliary posteriors
$\langle \log q(u_{n\ell}\|j)\rangle$	=	$\bar{a}_{nj\ell}\log\bar{b}_{nj\ell} + (\bar{a}_{nj\ell} - 1)\langle \log u_{n\ell}\rangle_j - \bar{b}_{nj\ell}\langle u_{n\ell}\rangle_j - \log\Gamma(\bar{a}_{nj\ell})$
$\langle \log q(v_{n\ell}\|j)\rangle$	=	$\bar{s}_{nj\ell}\log\bar{t}_{nj\ell} + (\bar{s}_{nj\ell} - 1)\langle \log v_{n\ell}\rangle_j - \bar{t}_{nj\ell}\langle v_{n\ell}\rangle_j - \log\Gamma(\bar{s}_{nj\ell})$
$\langle \log q(\pi)\rangle$	=	$\log\Gamma(\sum_j \hat{\alpha}_j) - \sum_j \log\Gamma(\hat{\alpha}_j) + \sum_j (\bar{\alpha}_j - 1)\langle \log\pi_j\rangle$
$\langle \log q(\beta_{j\ell})\rangle$	=	$(\bar{\kappa}_{1\ell} - 1)\langle \log\beta_{j\ell}\rangle + (\bar{\kappa}_{2\ell} - 1)\langle \log(1-\beta_{j\ell})\rangle - \log B(\bar{\kappa}_{1\ell}, \bar{\kappa}_{1\ell})$
$\langle \log q(\sigma_{j\ell})\rangle$	=	$\bar{\eta}_{j\ell}\log\bar{\omega}_{j\ell} + (\bar{\eta}_{j\ell} - 1)\langle \log\sigma_{j\ell}\rangle - \bar{\omega}_{j\ell}\langle \sigma_{j\ell}\rangle - \log\Gamma(\bar{\eta}_{j\ell})$
$\langle \log q(\tau_{j\ell})\rangle$	=	$\bar{\psi}_{j\ell}\log\bar{\xi}_{j\ell} + (\bar{\psi}_{j\ell} - 1)\langle \log\tau_{j\ell}\rangle - \bar{\xi}_{j\ell}\langle \tau_{j\ell}\rangle - \log\Gamma(\bar{\psi}_{j\ell})$
$\langle \log q(\mu_{j\ell})\rangle$	=	$\frac{1}{2}\log\bar{\sigma}_{j\ell} - \frac{1}{2}$
$\langle \log q(\chi_{j\ell})\rangle$	=	$\frac{1}{2}\log\bar{\varsigma}_{j\ell} - \frac{1}{2}$

a 2-D hierarchical clustering algorithm is first performed. The six subtypes are then recognized through the clustering results. A variety of statistical metrics (including χ^2 and t-statistics) are used to select discriminating genes for the subtypes.

We apply the developed algorithm to the data set to test if the developed algorithm is able to cluster the data accurately, and to find out the discriminating genes in the subtypes. Table IV shows the confusion matrix obtained by the developed algorithm given $K = 6$. From Table IV, we can see that the developed algorithm agrees with the clustering results in [47] quite accurately.

On the other hand, we want to justify whether the feature saliency criterion $\langle \beta_{j\ell}\rangle$, $1 \le \ell \le d$ can be used to discriminate the genes in each class j.[1] Since these values are defined to show the relevance of the features, or in the leukemia clustering context, these values indicate the relevance of the genes to describe the clusters. Thus, it is expected that the feature saliency values obtained by the developed algorithm can also be used to discriminate genes for the cancer subtypes. Here, we use the correlation between the feature saliency and the statistics to evaluate the usefulness of the feature saliency values on discriminate the clusters, which will imply the performance of the developed algorithm.

To measure the correlation between the feature saliency values and the statistics, we use the Pearson correlation coefficients. Table V lists the average coefficients obtained by running the developed algorithm for ten times. From the table, we can see that the absolute values of these statistics and the feature saliency obtained by the developed algorithm for the genes have fairly strong correlation; the average of the coefficients is as high as 0.851, and no less than 0.613. This implies a coherence between the developed method and the method in [47] in terms of selecting discriminating genes.

[1]Note that in our method, we use a localized feature saliency rather than a global feature saliency as developed in [34] and [37]. This enables us to discriminate genes at different clusters.

V. Conclusion

In this paper, we developed a hierarchical latent variable model for feature selection and robust clustering. A full Bayesian treatment was adopted for model selection. A VB framework was used for inference. To make the inference much efficient, a tree-structured factorization of the auxiliary posteriors for the latent variables was adopted which has been shown better than the widely used full factorization approach. Quantities are proposed to detect outliers and estimate feature saliency. Controlled experiments on synthetic and real data showed that the proposed model is able to realize outlier detection and feature selection more robustly than a semi-Bayesian mixture of Gaussians model. The application of the developed algorithm to real high-dimensional data shows its applicability. In the future, unsupervised feature selection analysis on data with "big dimensionality" (i.e., the feature size is normally far beyond 10k as reviewed in [3]) is our primary research avenue. The development and the application of feature selection algorithms in broadcasting [48], cloud computing [49], image processing [50], and other areas are another avenue.

Appendix A

In this section, we present the derivation of the posterior distribution with respect to u_n, $1 \leq n \leq N$. The derivation for the other latent variables and parameters is similar.

To derive $q(\mathbf{u}_n|\mathbf{z}_n = k)$ (or briefly $q(\mathbf{u}_n|k)$), we need to maximize the free energy with respect to $q(\mathbf{u}_n|k)$, subject to the constraint $\int q(\mathbf{u}_n|k)d\mathbf{u}_n = 1$. The free energy associated with the auxiliary posterior $q(\mathbf{u}_n|k)$ can be written as follows:

$$\mathcal{F}_{q(\mathbf{u}_n|k)} = \langle \log p(\mathbf{y}_n, h_n)\rangle_q - \langle \log q(\mathbf{u}_n|\mathbf{z}_n)\rangle_q.$$

Discarding terms that are independent of $\mathbf{u}_n$, and using the Lagrange multiplier method, the functional to be maximized is the following:

$$\begin{aligned}\mathcal{F}_{\mathbf{u}_n|k} &= q(\mathbf{z}_n = k)\langle \log p(\mathbf{y}_n|\mathbf{u}_n, k)p(\mathbf{u}_n|k)\rangle_q \\ &\quad - q(\mathbf{z}_n = k)\langle \log q(\mathbf{u}_n|k)\rangle_q + \lambda\left(\int q(\mathbf{u}_n|k)d\mathbf{u}_n - 1\right).\end{aligned}$$

Note here the expectation is computed with respect to the probability density functions of the parameters

Taking derivatives of $\mathcal{F}_{\mathbf{u}_n|k}$ with respect to $q(\mathbf{u}_n|k)$ and λ, we have

$$\begin{aligned}\frac{\partial \mathcal{F}_{\mathbf{u}_n|k}}{\partial q(\mathbf{u}_n|k)} &= -q(\mathbf{z}_n = k)\left[1 + \log q(\mathbf{u}_n|k)\right] \\ &\quad + q(\mathbf{z}_n = k)\log[p(\mathbf{y}_n|\mathbf{u}_n, k)p(\mathbf{u}_n|k)] + \lambda \\ \frac{\partial \mathcal{F}_{\mathbf{u}_n|k}}{\partial \lambda} &= \int q(\mathbf{u}_n|k)d\mathbf{u}_n - 1.\end{aligned}$$

If we let

$$\begin{aligned}E(\mathbf{u}_n, k) &= \langle \log p(\mathbf{y}_n|\mathbf{u}_n, k)p(\mathbf{u}_n|k)\rangle \\ &= \sum_\ell \langle \log p(y_{n\ell}|u_{n\ell}, k)p(u_{n\ell}|k)\rangle \qquad (17)\end{aligned}$$

and equating these to zero, then according to the Karush–Kuhn–Tucker conditions, we have

$$q(\mathbf{u}_n|k) = \frac{\exp(E(\mathbf{u}_n, k))}{\exp\left\{1 - \frac{\lambda}{q(\mathbf{z}_n=k)}\right\}} \qquad (18)$$

then taking integral with respect to $q(\mathbf{u}_n|k)$ on both sides, we have

$$\lambda = q(\mathbf{z}_n = k)\left(1 - \log\left[\int \exp(E(\mathbf{u}_n, k))\right]\right). \qquad (19)$$

Finally, replacing (19) into (18), we obtain

$$q(\mathbf{u}_n|k) = \frac{\exp(E(\mathbf{u}_n, k))}{\int \exp(E(\mathbf{u}_n, k))d\mathbf{u}_n}.$$

Note that the dimensions of the latent variable $\mathbf{u}_n$ are independent, we can then obtain

$$q(\mathbf{u}_n|k) = \prod_{\ell=1}^{d} \frac{\exp\langle \log p(y_{n\ell}|u_{n\ell}, k)p(u_{n\ell}|k)\rangle}{\int \exp\langle \log p(y_{n\ell}|u_{n\ell}, k)p(u_{n\ell}|k)\rangle du_{n\ell}}$$

which leads to the posterior presented in the main context.

Appendix B

The evaluation of the free energy is presented here. Note that the main evaluation is on the computation of the expectations of the logarithms of the prior distributions for the latent variables [including $p(\mathbf{y}_n|\mathbf{u}_n, j)$, $p(\mathbf{u}_n|j)$, $p(\mathbf{y}_n|\mathbf{v}_n)$, $p(\mathbf{v}_n)$, $p(\Phi_n|\beta)$]; the parameters [including $p(\beta)$, $p(\pi)$, $p(\sigma)$, $p(\mu)$, $p(\xi)$, $p(\tau)$]; the posteriors [including $q(\mathbf{u}_n|\mathbf{z}_n)$, $q(\mathbf{v}_n)$, $q(\mathbf{z}_n)$, $q(\pi)$, $q(\beta)$, $q(\tau)$, $q(\mu)$, $q(\sigma)$, and $q(\chi)$]. The evaluation of these expectations is summarized in Table VI.

Acknowledgment

The authors would like to thank anonymous reviewers for their constructive and helpful comments.

References

[1] X. Lu, Y. Wang, and Y. Yuan, "Sparse coding from a Bayesian perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 929–939, Jun. 2013.

[2] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic progamming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[3] Y. Zhai, Y.-S. Ong, and I. Tsang, "The emerging 'big dimensionality,"' *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, Aug. 2014.

[4] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Aug. 2004.

[5] L. Laporte, R. Flamary, S. Canu, S. Déjean, and J. Mothe, "Nonconvex regularizations for feature selection in ranking with sparse SVM," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1118–1130, Jun. 2014.

[6] R. Chakraborty and N. R. Pal, "Feature selection using a neural framework with controlled redundancy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 35–49, Jan. 2015.

[7] Y. Li, J. Si, G. Zhou, S. Huang, and S. Chen, "FREL: A stable feature selection algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1388–1402, Jul. 2015.

[8] T. Naghibi, S. Hoffmann, and B. Pfister, "A semidefinite programming based search strategy for feature selection with mutual information measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1529–1541, Aug. 2016.

[9] H. Tao, C. Hou, F. Nie, Y. Jiao, and D. Yi, "Effective discriminative feature selection with nontrivial solution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 796–808, Apr. 2016.

[10] P. Padungweang, C. Lursinsap, and K. Sunat, "A discrimination analysis for unsupervised feature selection via optic diffraction principle," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1587–1600, Oct. 2012.

[11] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.

[12] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu, "Unsupervised feature selection via maximum projection and minimum redundancy," *Knowl.-Based Syst.*, vol. 75, pp. 19–29, Feb. 2015.

[13] J. Yao, Q. Mao, S. Goodison, V. Mai, and Y. Sun, "Feature selection for unsupervised learning through local learning," *Pattern Recognit. Lett.*, vol. 53, pp. 100–107, Feb. 2015.

[14] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang, "Graph regularized feature selection with data reconstruction," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 689–700, Mar. 2016.

[15] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2521602.

[16] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.

[17] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2014.

[18] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. NIPS*, 2005, pp. 80–87.

[19] Y. Jiang and J. Ren, "Eigenvalue sensitive feature selection," in *Proc. 28th ICML*, 2011, pp. 89–96.

[20] M. Sebban and R. Nock, "A hybrid filter/wrapper approach of feature selection using information theory," *Pattern Recognit.*, vol. 35, no. 4, pp. 835–846, 2002.

[21] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 856–863.

[22] A. Arauzo-Azofra, J. M. Benitez, and J. L. Castro, "Consistency measures for feature selection," *J. Intell. Inf. Syst.*, vol. 30, no. 3, pp. 273–292, 2007.

[23] Q. Gu and J. Zhou, "Local learning regularized nonnegative matrix factorization," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1–6.

[24] H. Zeng and Y.-M. Cheung, "Feature selection and kernel learning for local learning-based clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1532–1547, Aug. 2011.

[25] A. Ng, "Feature selection, L_1 vs. L_2 regularization, and rotational invariance," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 78.

[26] Y. Yang, H. Shen, Z. Ma, and Z. Huang, "$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1–6.

[27] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23. 2010, pp. 1813–1821.

[28] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1–7.

[29] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proc. 7th SIAM Int. Conf. Data Mining*, 2007.

[30] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. 24th ICML*, 2007, pp. 1151–1157.

[31] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2138–2150, Sep. 2014.

[32] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[33] S. H. Yang and B. G. Hu, "Discriminative feature selection by non-parametric Bayes error minimization," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1422–1434, Aug. 2012.

[34] C. Constantinopoulos, M. K. Titsia, and A. Likas, "Bayesian feature and model selection for Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 1013–1018, Jun. 2006.

[35] P. Carbonetto, N. de Freitas, P. Gustafson, and N. Thompson, "Bayesian feature weighting for unsupervised learning, with application to object recognition," in *Proc. 9th Int. Conf. Artif. Intell. Statist.*, 2003.

[36] W. Pan and X. Shen, "Penalized model-based clustering with application to variable selection," *J. Mach. Learn. Res.*, vol. 8, pp. 1145–1164, May 2007.

[37] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1154–1166, Sep. 2004.

[38] Y. Li, M. Dong, and J. Hua, "Simultaneous localized feature selection and model detection of Gaussian mixtures," *IEEE Trans. Pattern Recognit. Mach. Intell.*, vol. 31, no. 5, pp. 953–960, May 2009.

[39] J. Sun, A. Kabán, and S. Raychaudhury, "Robust mixtures in the presence of measurement errors," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, USA, 2007, pp. 847–854.

[40] J. Sun, A. Kaban, and J. Garibaldi, "Robust mixture modeling using the Pearson type VII distribution," *Pattern Recognit. Lett.*, vol. 31, no. 16, pp. 2447–2454, 2010.

[41] G. McLachlan and D. Peel, *Finite Mixture Models*. Hoboken, NJ, USA: Wiley, 2000.

[42] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[43] D. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Int. Soc. Bayesian Anal.*, Durham, USA, vol. 1, no. 1, pp. 121–144, 2006.

[44] J. Sun and A. Kaban, "A fast algorithm for robust mixtures in the presence of measurement errors," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1206–1220, Aug. 2010.

[45] J. Sun, J. Garibaldi, and K. Kenobi, "Robust Bayesian clustering for datasets with repeated measures," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 9, no. 5, pp. 1504–1514, Sep. 2012.

[46] A. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Learn.*, vol. 22, no. 1, pp. 4–38, Jan. 2000.

[47] E.-J. Yeoh *et al.*, "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling," *Cancer Cell*, vol. 1, no. 2, pp. 133–143, 2002.

[48] Z. Pan, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast motion estimation based on content property for low-complexity h.265/hevc encoder," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 675–684, Sep. 2016.

[49] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. E98-B, no. 1, pp. 190–200, 2015.

[50] Y. Zheng, J. Byeungwoo, D. Xu, Q. M. J. Wu, and H. Zhang, "Image segmentation by generalized hierarchical fuzzy C-means algorithm," *J. Intell. Fuzzy Syst.*, vol. 28, no. 2, pp. 961–973, 2015.

Jianyong Sun received the B.Sc. degree in computational mathematics from Xi'an Jiaotong University, Xi'an, China, in 1997, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K., in 2005.

He was a Lecturer with the University of Essex and a Senior Lecturer with the University of Greenwich, London, U.K. He is currently a Professor with Xi'an Jiaotong University. He has authored over 40 peer-reviewed research papers. His current research interests include theoretical and practical aspects of artificial intelligence, mainly on evolutionary computation, statistical machine learning and their applications in bioinformatics, and astro-informatics.

Aimin Zhou (S'08–M'10) received the B.Sc. and M.Sc. degrees from Wuhan University, Wuhan, China, in 2001 and 2003, respectively, and the Ph.D. degree from the University of Essex, Colchester, U.K., in 2009, all in computer science.

He is currently an Associate Professor with the Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, Shanghai, China. He has authored over 40 peer-reviewed papers. His current research interests include evolutionary computation and optimisation, machine learning, image processing, and their applications.

Dr. Zhou is an Associate Editor of Swarm and Evolutionary Computation and an Editorial Board Member of Complex and Intelligent Systems.

Simeon Keates received the M.A. and Ph.D. degrees in engineering from the Department of Engineering, University of Cambridge, Cambridge, U.K.

He was the Head of the School of Engineering, Computing and Applied Mathematics, University of Abertay Dundee, Dundee, Scotland, and an Associate Professor with the IT University of Copenhagen, Copenhagen, Denmark, where he lectured in the Design and Digital Communication study line. He was an Industrial Research Fellow with the Engineering Design Centre, University of Cambridge, supported by Royal Mail. He joined the Accessibility Research Group, IBM TJ Watson Research Center. He was with ITA Software as a Usability Lead designing interfaces for Air Canada. He is currently the Deputy Pro Vice Chancellor with the Faculty of Engineering and Science, University of Greenwich, London, U.K., where he is responsible for strategic management of Engineering and also for industrial engagement. He also has an extensive history of consultancy, with clients including Royal Mail, the US Social Security Administration, the U.K. Department of Trade and Industry, Danish Broadcasting Corporation (Danske Radio) and Lockheed Martin.

Dr. Keates was the Chair of HCI.

Shengbin Liao received the Ph.D. degree from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2008.

He is currently an Associate Professor with the National Engineering Research Center for E-Learning, Huazhong Normal University, Wuhan. His current research interests include machine learning and big data, distributed optimization and control, and automated problem solving based on deep learning.