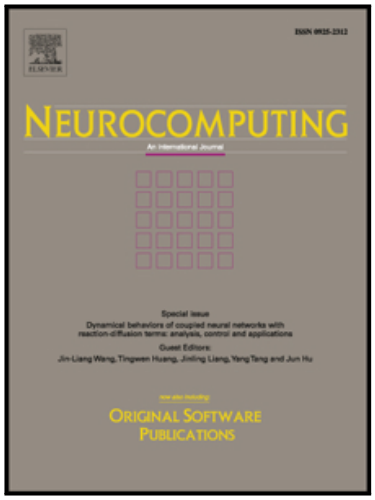
Communicated by Shiliang Sun

Accepted Manuscript



A Novel Statistical Approach for Clustering Positive Data Based on Finite Inverted Beta-Liouville Mixture Models

Can Hu, Wentao Fan, Ji-Xiang Du, Nizar Bouguila

PII: S0925-2312(18)31528-5
DOI: https://doi.org/10.1016/j.neucom.2018.12.066
Reference: NEUCOM 20288

To appear in: *Neurocomputing*

Received date: 7 February 2018
Revised date: 22 September 2018
Accepted date: 26 December 2018

Please cite this article as: Can Hu, Wentao Fan, Ji-Xiang Du, Nizar Bouguila, A Novel Statistical Approach for Clustering Positive Data Based on Finite Inverted Beta-Liouville Mixture Models, *Neurocomputing* (2018), doi: https://doi.org/10.1016/j.neucom.2018.12.066

A Novel Statistical Approach for Clustering Positive Data Based on Finite Inverted Beta-Liouville Mixture Models

Can Hu[a], Wentao Fan[a], Ji-Xiang Du[a], Nizar Bouguila[b]

[a] *Department of Computer Science and Technology, Huaqiao University, Xiamen, China*
[b] *Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada*

Abstract

Nowadays, a great number of positive data has been occurred naturally in many applications, however, it was not adequately analyzed. In this article, we propose a novel statistical approach for clustering multivariate positive data. Our approach is based on a finite mixture model of inverted Beta-Liouville (IBL) distributions, which is proper choice for modeling and analysis of positive vector data. We develop two different approaches to learn the proposed mixture model. Firstly, the maximum likelihood (ML) is utilized to estimate parameters of the finite inverted Beta-Liouville mixture model in which the right number of mixture components is determined according to the minimum message length (MML) criterion. Secondly, the variational Bayes (VB) is adopted to learn our model where the parameters and the number of mixture components can be determined simultaneously in a unified framework, without the requirement of using information criteria. We investigate the effectiveness of our model by conducting a series of experiments on both synthetic and real data sets.

Keywords: Clustering, mixture models, variational Bayes, maximum likelihood, minimum message length, inverted Beta-Liouville

Email addresses: canhu@hqu.edu.cn (Can Hu), fwt@hqu.edu.cn (Wentao Fan), jxdu@hqu.edu.cn (Ji-Xiang Du), nizar.bouguila@concordia.ca (Nizar Bouguila)

1. Introduction

Data clustering is a common unsupervised learning technology for data analysis via discovering similar statistical characters in a data set. It has been widely applied in many fields such as image processing[1], remote sensing [2], data mining [3]. Thus, there is an urgent need for effective technologies to model and analyze complicated data. Among the existing proposed technologies, finite mixture models have been successfully used and showed excellent performance of clustering [4, 5, 6]. The finite mixture model is motivated as a linear superposition of statistical distributions with varying proportions and shows the simplicity and flexibility for clustering. Most existing related works, however, have not taken into account the characteristics of data set. Indeed, most of finite mixture models mainly consider Gaussian as their basic distributions[7]. Nevertheless, it is obvious not an appropriate choice to model non-Gaussian data. For example, Dirichlet or generalized Dirichlet mixture models [8, 9] can often outperform the Gaussian mixture model for modeling proportional data in many applications such as image categorization, human action video recognition, etc.

In recent years, several works have been proposed to model positive data based on inverted Dirichlet mixture models [10, 11]. However, the inverted Dirichlet distribution has a very restrictive covariance structure that considerably limited its flexibility. In our work, we propose to model positive data based on a finite mixture model with inverted Beta-Liouville (IBL) distributions [12]. We are mainly motivated by the fact that the IBL distribution contains inverted Dirichlet distribution as a special case and therefore can provide more flexibility. Also, compared with Gaussian which can only approximate symmetric distributions, IBL allows both symmetric and asymmetric distributions.

A classic approach to learn finite mixture models is through maximum likelihood (ML) [13] and is usually carried out based on expectation maximization (EM) [14]. However, one problem of using ML in mixture modeling is that it lacks the ability to determine model complexity (i.e., the number of mixture components). A common solution is to add a determination step based on some

typical information criteria such as Akaike information criterion (AIC) [15], Bayes information criterion (BIC) [16], minimum description length (MDL) [17]. Based on the work of [15], all above criteria can be seen as an approximation to a particular criterion namely the minimum message length (MML) [18, 19]. The effectiveness of using EM algorithm together with MML to learn finite mixture models have been demonstrated through several works that have been proposed during the last decade [20, 21]. Thus, the first approach that we develop to learn finite IBL mixture models is based on a framework that using EM algorithm to estimate parameters and MML criterion to inference the number of mixture components. Even though ML is an effective approach to learn finite mixture models, it may suffer if the initialization was poorly chosen and would result in over-fitting. To tackle this problem, we may consider an alternative approach to learn IBL mixture models based on a Bayesian framework known as variational Bayes (VB) [22, 23, 24]. The VB algorithm provides a tractable lower bound for marginal distribution to approximate the real posterior distribution, where closed-form solutions are obtained without additional iterative numerical calculation. In contrast with the ML algorithm, the VB algorithm can estimate model parameters and select the optimal number of clusters simultaneously.

The major contributions of this work are illustrated as follows: 1) We propose a new statistical model-based approach for clustering positive data based on finite IBL mixture models. 2) We develop two different approaches to learn the proposed IBL mixture models. The first learning approach is based on the EM algorithm and uses MML criterion to determine the number of mixture components. The second learning approach is built by exploiting a VB inference framework, such that the parameters of our mixture model and the number of mixture components can be evaluated simultaneously in a unified framework. 3) The effectiveness of our approaches for learning the finite IBL mixture model and the clustering applications of the finite IBL mixture model are shown through extensive experiments.

The rest of this paper is organized as follows. In section 2, we present the finite IBL mixture model and the ML estimation based on MML is also given.

In section 3, the VB algorithm for learning the finite IBL mixture model is presented. The experiments based on synthetic data and real applications are conducted in section 4 and the conclusion follows in section 5.

2. Finite Inverted Beta-Liouville Mixture Model And Maximum Likelihood

2.1. Finite Inverted Beta-Liouville Mixture Model

If a D-dimension vector $\vec{X} = \{X_1, \cdots, X_D\}$ is drawn from a inverted Beta-Liouville (IBL) distribution [12], then we have

$$p(\vec{X}|\alpha_1, \ldots, \alpha_d, \alpha, \beta, \lambda) = \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\prod_{d=1}^{D}\frac{X_d^{\alpha_d-1}}{\Gamma(\alpha_d)} \times\lambda^{\beta}\Big(\sum_{d=1}^{D}X_d\Big)^{\alpha-\sum_{d=1}^{D}\alpha_d}\Big(\lambda+\sum_{d=1}^{D}X_d\Big)^{-(\alpha+\beta)}, \quad (1)$$

where $X_d > 0$ for $d = 1, \cdots, D$, $\alpha > 0$, $\beta > 0$ and $\lambda > 0$. Actually, the IBL distribution can be viewed as a generalized form of inverted Dirichlet distribution that may contain multiple symmetric and asymmetric modes. More details about IBL distribution can be found from [12].

The mean, variance and covariance of the IBL distribution are given by

$$E(X_d) = \frac{\lambda\alpha}{\beta-1}\frac{\alpha_d}{\sum_{d=1}^{D}\alpha_d}, \quad (2)$$

$$Var(X_d) = \frac{\lambda^2\alpha(\alpha+1)}{(\beta-1)(\beta-2)}\frac{\alpha_d(\alpha+1)}{\sum_{d=1}^{D}\alpha_d(\sum_{d=1}^{D}\alpha_d+1)} - \frac{\lambda^2\alpha^2}{(\beta-1)^2}\frac{\alpha_d^4}{(\sum_{d=1}^{D}\alpha_d)^4}, \quad (3)$$

$$Cov(X_m, X_n) = \frac{\alpha_m\alpha_n}{\sum_{d=1}^{D}\alpha_d}\Bigg[\frac{\lambda^2\alpha(\alpha+1)}{(\beta-1)(\beta-2)(\sum_{d=1}^{D}\alpha_d+1)} - \frac{\lambda^2\alpha^2}{(\beta-1)^2(\sum_{d=1}^{D}\alpha_d)}\Bigg]. \quad (4)$$

Given a set of data that contains N vectors: $\mathcal{X} = \{\vec{X}_1, \cdots, \vec{X}_N\}$, where each $\vec{X}_i = \{X_{i1}, \cdots, X_{iD}\}$ is drawn from the finite IBL mixture model with M components and is defined as follow

$$p(\vec{X}_i|\vec{\pi}, \Theta) = \sum_{j=1}^{M} \pi_j p(\vec{X}_i|\theta_j). \tag{5}$$

where $\Theta = (\theta_1, \cdots, \theta_M)$, $p(\vec{X}_i|\theta_j)$ denotes the IBL distribution in Eq. (1) associated with the jth component with parameters $\theta_j = (\alpha_{j1}, \ldots, \alpha_{jD}, \alpha_j, \beta_j, \lambda_j)$, and $\vec{\pi} = (\pi_1, \ldots, \pi_M)$ represent the mixing coefficients where $0 \leq \pi_j \leq 1$ and $\sum_{j=1}^{M} = 1$.

2.2. Maximum Likelihood Estimation

An important step for learning finite mixture models is to estimate involved parameters. In this part, we develop a learning approach based on maximum likelihood (ML) to learn our finite IBL mixture model. Specifically, the values of parameters are obtained by maximizing the log-likelihood function as

$$\tilde{\Theta} = \underset{\Theta}{argmax} \log p(\mathcal{X}|\vec{\pi}, \Theta), \tag{6}$$

where the log-likelihood function is generally given by

$$L(\mathcal{X}|\vec{\pi}, \Theta) = \log\, p(\mathcal{X}|\vec{\pi}, \Theta) = \log \prod_{i=1}^{N} p(\vec{X}_i|\vec{\pi}, \Theta) = \sum_{i=1}^{N} \log\Big(\sum_{j=1}^{M} \pi_j p(\vec{X}_i|\theta_j) \Big) \tag{7}$$

Now we define latent variables as indicator variables for a set of data that is observed. Let $\mathcal{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$, each $\vec{Z}_i = (Z_{i1}, \cdots, Z_{iM})$ corresponds to an observed data $\vec{X}_i$, where $Z_{ij} \in \{0, 1\}$ and $\sum_{j=1}^{M} Z_{ij} = 1$, and $Z_{ij} = 1$ if $\vec{X}_i$ belongs to component j, and 0, otherwise. Then, the log-likelihood function of the complete data set $\{\mathcal{X}, \mathcal{Z}\}$ takes the form

$$\Phi(\mathcal{X}, \mathcal{Z}|\vec{\pi}, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{ij} \{\log \pi_j + \log\, p(\vec{X}_i|\theta_j)\}. \tag{8}$$

Next, the conditional expectation of the complete-data log-likelihood function is maximized in the M-step of EM algorithm which is given by

$$\Omega(\mathcal{X}|\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} \langle Z_{ij} \rangle \{\log \pi_j + \log\, p(\vec{X}_i|\theta_j)\}, \tag{9}$$

where $\langle Z_{ij} \rangle$ (i.e., the posterior probability) denotes the expected value of the indicator variable and is given by

$$\langle Z_{ij} \rangle = \frac{\pi_j p(\vec{X}_i|\theta_j)}{\sum_{k=1}^{M} \pi_k p(\vec{X}_i|\theta_k)}. \tag{10}$$

Then, we can maximize $\Omega(\mathcal{X}|\Theta)$ as described in Eq. (9) by computing the first derivatives with respect to all parameters as follows

$$\begin{aligned}\frac{\partial \Omega(\mathcal{X}|\Theta)}{\partial \alpha_j} = &\sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\log \sum_{d=1}^{D} X_{id} - \log(\lambda_j + \sum_{d=1}^{D} X_{id}) \right] \\ &+ [\Psi(\alpha_j + \beta_j) - \Psi(\alpha_j)] \sum_{i=1}^{N} \langle Z_{ij} \rangle,\end{aligned} \tag{11}$$

$$\begin{aligned}\frac{\partial \Omega(\mathcal{X}|\Theta)}{\partial \beta_j} = &\sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\log \lambda_j - \log(\lambda_j + \sum_{d=1}^{D} X_{id}) \right] \\ &+ [\Psi(\alpha_j + \beta_j) - \Psi(\beta_j)] \sum_{i=1}^{N} \langle Z_{ij} \rangle,\end{aligned} \tag{12}$$

$$\begin{aligned}\frac{\partial \Omega(\mathcal{X}|\Theta)}{\partial \alpha_{jd}} = &\sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\log X_{id} - \log \sum_{d=1}^{D} X_{id} \right] \\ &+ \left[\Psi(\sum_{d=1}^{D} \alpha_{jd}) - \Psi(\alpha_{jd}) \right] \sum_{i=1}^{N} \langle Z_{ij} \rangle,\end{aligned} \tag{13}$$

$$\frac{\partial \Omega(\mathcal{X}|\Theta)}{\partial \lambda_j} = \sum_{i=1}^{N} \langle Z_{ij} \rangle \left[\frac{\beta_j}{\lambda_j} - \frac{\alpha_j + \beta_j}{\lambda_j + \sum_{d=1}^{D} X_{id}} \right], \tag{14}$$

where $\Psi(\cdot)$ denotes digamma function. From Eq. (11) to Eq. (14), it is clear that a closed-form solution for θ_j does not exist. To estimate these unknown parameters, the Newton-Raphson method is utilized

$$\theta_j^{(t+1)} = \theta_j^{(t)} - H(\theta_j^{(t)})^{-1} \frac{\partial \Omega(\mathcal{X}|\vec{\pi}^{(t)}, \Theta^{(t)})}{\partial \theta_j^{(t)}}, \tag{15}$$

where $H(\theta_j^{(t)})^{-1}$ denotes the inverse Hessian matrix for parameter θ_j and is described in details in Appendix A. It is worth noting that the closed-form solution for mixing coefficients π_j exists and is given by

$$\pi_j = \frac{1}{N} \sum_{i=1}^{N} \langle Z_{ij} \rangle. \tag{16}$$

2.3. MML Criteria For Estimating Parameters

One fundamental issue in mixture modeling is how to correctly and automatically select the optimal number of mixture components. There have been some criteria applied in dealing with this problem by mainly evaluating two parts including data part of maximizing likelihood and a penalty part of the complexity of statistical models [25]. Among these existing criteria, the minimum message length (MML) criterion has been widely used and shown excellent performance in many applications [20, 26]. The MML criterion for finite mixture models is generally defined by

$$MML \simeq -\log(h(\Theta,\vec{\pi})) - \log(p(\mathcal{X}|\vec{\pi},\Theta)) + \frac{1}{2}\log(|F(\Theta,\vec{\pi})|) + \frac{N_p}{2}(1-\log(12)), \tag{17}$$

where $h(\Theta)$ denotes the prior probability, $\log(p(\mathcal{X}|\vec{\pi},\Theta))$ is the likelihood which can be obtained from Eq. (7), $F(\Theta)$ represents the expected Fisher information matrix [19], $|\cdot|$ denotes determinant, and N_p denotes the number of the free estimated parameters (i.e., $N_p = M(D+4)-1$). The $\log(|F(\Theta)|)$ of the finite IBL mixture model can be approximated as (please see Appendix B for detail)

$$\begin{aligned}\log(|F(\Theta)|) \simeq\ & (M-1)\log(N) + \sum_{j=1}^{M}\log(|\tilde{H}(\alpha_j,\beta_j,\lambda_j)|) \\ & + \sum_{j=1}^{M}\log\left(\left|1-\Psi'(\sum_{d=1}^{D})\sum_{d=1}^{D}\frac{1}{\Psi'(\alpha_{jd})}\right|\right) - \sum_{j=1}^{M}\log(\pi_j) \\ & + D\sum_{j=1}^{M}\log(n_j) + \sum_{j=1}^{M}\sum_{d=1}^{D}\log(\Psi'(\alpha_{jd})).\end{aligned} \tag{18}$$

The prior $h(\Theta)$ is defined as (details can be viewed in Appendix C)

$$h(\Theta) = (M-1)!\prod_{j=1}^{M}\left[f(\alpha_j,\beta_j,\lambda_j)^{-3}\prod_{d=1}^{D}f(\alpha_{jd})\right], \tag{19}$$

where

$$f(\alpha_j,\beta_j,\lambda_j) = \left[\frac{e^{18}(\hat{\alpha}_j+\hat{\beta}_j+\hat{\lambda}_j)^3}{\hat{\alpha}_j\hat{\beta}_j\hat{\lambda}_j}\right]^{-1}, \tag{20}$$

and with

$$f(\alpha_{jd}) = e^6\frac{\sum_{d=1}^{D}\hat{\alpha}_{jd}}{\hat{\alpha}_{jd}}, \tag{21}$$

where the parameters with the hat notation are the estimated parameters.

Based on [27], MML is based on evaluating statistical models and is able to compress information from data. In addition, based on information-theory, the optimal number of mixture components occurs when the minimum of information is obtained. Thus, MML has the ability to select the optimal clusters to describe data set. The complete algorithm based on EM algorithm and MML criterion is summarized as follows

Algorithm 1

1: Initialization.

2: *E-step*: calculate the posterior probability $\langle Z_{ij} \rangle$ according to Eq. (10)

3: *M-step*: update parameters θ_j and π_j using Eq. (15) and Eq. (16), respectively.

4: calculate the MML criterion using Eq. (17).

5: select the optimal component M^* such that:

$$M^* = \underset{M}{argmin}\, MML(M).$$

3. Variational Learning For Estimating Parameters

Since EM algorithm may result in over-fitting due to poor initialization, we provide an alternative learning approach, in this section, that can estimate parameters and select the right number of components of the finite IBL mixture model. Our approach is based on a Bayesian framework known as variational Bayes (VB), which has shown promising results in learning mixture models [28, 8].

3.1. Latent Variables and Prior Distributions

We define latent variables $\mathcal{Z} = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$ as indicator variables for an observed data set. Each $\vec{Z}_i = (Z_{i1}, \cdots, Z_{iM})$ corresponds to a data point $\vec{X}_i$, where $Z_{ij} \in \{0, 1\}$, $\sum_{j=1}^{M} Z_{ij} = 1$, and $Z_{ij} = 1$ if $\vec{X}_i$ belongs to component j,

and 0, otherwise. The conditional distribution of $\mathcal{Z}$ given the mixing coefficients $\vec{\pi}$ takes the form

$$p(\mathcal{Z}|\vec{\pi}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\pi_j^{Z_{ij}}. \tag{22}$$

Then, the likelihood function of data set $\mathcal{X}$ with latent variables $\mathcal{Z}$ and related parameters Θ is given by

$$p(\mathcal{X}|\mathcal{Z},\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{M}p(\vec{X}_i|\theta_j)^{Z_{ij}}. \tag{23}$$

Next, we place priors over parameters $\Theta = (\boldsymbol{\alpha}, \vec{\alpha}, \vec{\beta}, \vec{\lambda})$. Since $\boldsymbol{\alpha}, \vec{\alpha}, \vec{\beta}, \vec{\lambda}$, Θ are positive, Gamma distribution $G(\cdot)$ is adopted as their priors

$$p(\boldsymbol{\alpha}) = G(\boldsymbol{\alpha}|\vec{u},\vec{v}) = \prod_{j=1}^{M}\prod_{d=1}^{D}\frac{\alpha_{jd}^{u_{jd}-1}e^{-v_{jd}\alpha_{jd}}}{\Gamma(u_{jd})}, \tag{24}$$

$$p(\vec{\alpha}) = G(\vec{\alpha}|\vec{g},\vec{h}) = \prod_{j=1}^{M}\frac{\alpha_j^{g_j-1}e^{-h_j\alpha_j}}{\Gamma(g_j)}, \tag{25}$$

$$p(\vec{\beta}) = G(\vec{\beta}|\vec{s},\vec{t}) = \prod_{j=1}^{M}\frac{\beta_j^{s_j-1}e^{-t_j\beta_j}}{\Gamma(s_j)}, \tag{26}$$

$$p(\vec{\lambda}) = G(\vec{\lambda}|\vec{c},\vec{f}) = \prod_{j=1}^{M}\frac{\lambda_j^{c_j-1}e^{-f_j\lambda_j}}{\Gamma(c_j)}. \tag{27}$$

Then, for the finite IBL mixture model, the joint distribution of all random variables and latent variables given mixing coefficients $\vec{\pi}$ is defined by

$$p(\mathcal{X},\mathcal{Z},\Theta|\vec{\pi}) = p(\mathcal{X}|\mathcal{Z},\Theta)p(\mathcal{Z}|\vec{\pi})p(\boldsymbol{\alpha})p(\vec{\alpha})p(\vec{\beta})p(\vec{\lambda}). \tag{28}$$

The graphical model of finite IBL mixture model is shown in Fig. 1.

3.2. Model Learning via VB Inference

For the finite IBL mixture model, the goal of VB is to find a lower bound on $p(\mathcal{X}|\vec{\pi})$ via Jensen's inequality. Here, we define $\Lambda = \{\mathcal{Z}, \Theta\}$, and the lower bound $\mathcal{L}(q)$ can then be obtained by

$$\begin{aligned}\log p(\mathcal{X}|\vec{\pi}) &= \log\int p(\mathcal{X},\Lambda|\vec{\pi})d\Lambda = \log\int q(\Lambda)\frac{p(\mathcal{X},\Lambda|\vec{\pi})}{q(\Lambda)}d\Lambda \\ &\geq \int q(\Lambda)\log\frac{p(\mathcal{X},\Lambda)|\vec{\pi}}{q(\Lambda)}d\Lambda = \mathcal{L}(q),\end{aligned} \tag{29}$$

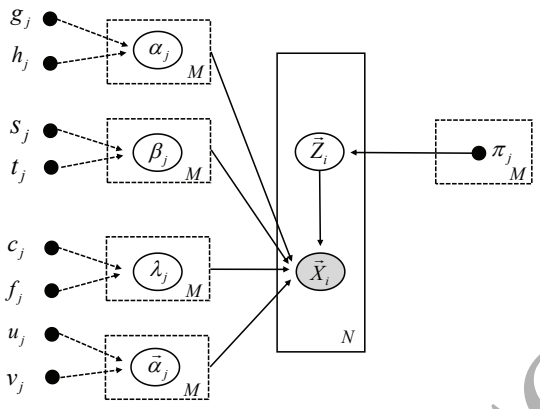


Figure 1: Graphical model of finite IBL mixture model. Symbols in circle denote variables and black points are parameters. Arcs represent the conditional dependence between two variables, and plates denote the replication (shown in the lower right with the number of replications).

where $q(\Lambda)$ is an approximation for the posterior distribution $p(\Lambda|\mathcal{X}, \vec{\pi})$. Then, we can decompose the log marginal probability as

$$\log\, p(\mathcal{X}|\vec{\pi}) = \mathcal{L}(q) + \mathrm{KL}(q\|p), \tag{30}$$

where the Kullback-Leibler divergence $\mathrm{KL}(q\|p)$ is defined by

$$\mathrm{KL}(q\|p) = -\int q(\Lambda)\log\frac{p(\Lambda|\mathcal{X}, \vec{\pi})}{q(\Lambda)}d\Lambda. \tag{31}$$

In our work, we adopt the *mean field* assumption [29, 30, 31] to restrict the family of distribution. Thus, the posterior distribution $q(\Lambda)$ can be factorized into different factors as

$$q(\Lambda) = q(\mathcal{Z})q(\Theta) = q(\mathcal{Z})q(\boldsymbol{\alpha})q(\vec{\alpha})q(\vec{\beta})q(\vec{\lambda}). \tag{32}$$

Then, we need to find proper individual factors to maximize the lower bound $\mathcal{L}(q)$ via variational optimization with respect to each factor in turn. The optimal solution to the factor $q(\mathcal{Z})$ is given by

$$q(\mathcal{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{M} r_{ij}^{Z_{ij}}, \tag{33}$$

where

$$r_{ij} = \frac{r^*_{ij}}{\sum_{k=1}^{M} r^*_{ik}}, \tag{34}$$

with

$$r^*_{ij} = \exp\Big[\log\pi_j + S_j + T_j + (\bar{\alpha}_j - \sum_{d=1}^{D}\bar{\alpha}_{jd})\log(\sum_{d=1}^{D} X_{id}) + \bar{\beta}_j\langle\log\lambda_j\rangle + \sum_{d=1}^{D}(\bar{\alpha}_{jd}-1)\log X_{id}) - (\bar{\alpha}_j + \bar{\beta}_j)H_{ij}\Big], \tag{35}$$

with expected values that are defined as follows

$$\bar{\alpha}_j = \frac{g^*_j}{h^*_j}, \qquad \bar{\beta}_j = \frac{s^*_j}{t^*_j}, \qquad \bar{\alpha}_{jd} = \frac{u^*_{jd}}{v^*_{jd}}, \quad \bar{\lambda}_j = \frac{c^*_{jd}}{f^*_{jd}},$$

$$H_{ij} = \Big\langle \log(\lambda_j + \sum_{d=1}^{D} X_{id}) \Big\rangle, \quad \langle\log\lambda_j\rangle = \Psi(c^*_j) - \log(f^*_j),$$

$$S_j = \Big\langle \log\frac{\Gamma(\sum_{d=1}^{D}\alpha_{jd})}{\prod_{d=1}^{D}\Gamma(\alpha_{jd})} \Big\rangle, \quad T_j = \Big\langle \log\frac{\Gamma(\alpha_j+\beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \Big\rangle. \tag{36}$$

Since H_{ij}, S_j and T_j are intractable, we use second order Taylor series expansion to calculate their lower bounds.

Similarly, the optimal solution to the factor $q(\boldsymbol{\alpha})$ is given by

$$q(\boldsymbol{\alpha}) = \prod_{j=1}^{M}\prod_{d=1}^{D} G(\alpha_{jd}|u^*_{jd}, v^*_{jd}), \tag{37}$$

where we have

$$u^*_{jd} = u_{jd} + \sum_{i=1}^{N}\langle Z_{ij}\rangle\bar{\alpha}_{jd}\Big[\Psi(\sum_{d=1}^{D}\bar{\alpha}_{jd}) - \Psi(\bar{\alpha}_{jd}) + \Psi'(\sum_{d=1}^{D}\bar{\alpha}_{jd})\sum_{l\neq d}^{D}(\langle\log\alpha_{jl}\rangle - \log\bar{\alpha}_{jl})\bar{\alpha}_{jl}\Big], \tag{38}$$

$$v^*_{jd} = v_{jd} - \sum_{i=1}^{N}\langle Z_{ij}\rangle\Big[\log X_{id} - \log(\sum_{d=1}^{D} X_{id})\Big], \tag{39}$$

and with expected values

$$\langle Z_{ij}\rangle = r_{ij}, \qquad \langle\log\alpha_{jl}\rangle = \Psi(u^*_{jl}) - \Psi(v^*_{jl}). \tag{40}$$

Next, the optimal solution to the factor $q(\vec{\alpha})$ can be calculated by

$$q(\vec{\alpha}) = \prod_{j=1}^{M} G(\alpha_j | g_j^*, h_j^*), \tag{41}$$

where g_j^* and h_j^* are given by

$$g_j^* = g_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \Big[\Psi(\bar{\alpha}_j + \bar{\beta}_j) - \Psi(\bar{\alpha}_j) + \bar{\beta}_j \Psi'(\bar{\alpha}_j + \bar{\beta}_j)(\langle \log \beta_j \rangle - \log \bar{\beta}_j) \Big] \bar{\alpha}_j, \tag{42}$$

$$h_j^* = h_j - \sum_{i=1}^{N} \langle Z_{ij} \rangle \log \Big(\sum_{d=1}^{D} X_{id} \Big) + \sum_{i=1}^{N} \langle Z_{ij} \rangle H_{ij}, \tag{43}$$

and we have

$$\langle \log \beta_j \rangle = \Psi(s_j^*) - \log(t_j^*). \tag{44}$$

Then, the variational optimal solution $q(\vec{\beta})$ can be updated by

$$q(\vec{\beta}) = \prod_{j=1}^{M} G(\beta_j | s_j^*, t_j^*), \tag{45}$$

where we have

$$s_j^* = s_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \Big[\Psi(\bar{\alpha}_j + \bar{\beta}_j) - \Psi(\bar{\beta}_j) + \bar{\alpha}_j \Psi'(\bar{\alpha}_j + \bar{\beta}_j)(\langle \log \alpha_j \rangle - \log \bar{\alpha}_j) \Big] \bar{\beta}_j \tag{46}$$

$$t_j^* = t_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \Big[H_{ij} - \langle \log \lambda_j \rangle \Big], \tag{47}$$

with

$$\langle \log \lambda_j \rangle = \Psi(g_j^*) - \log(h_j^*). \tag{48}$$

Finally, the variational optimal solution to $q(\vec{\lambda})$ can be updated by

$$q(\vec{\lambda}) = \prod_{j=1}^{M} G(c_j | f_j^*, t_j^*), \tag{49}$$

where

$$c_j^* = c_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \bar{\beta}_j, \tag{50}$$

$$f_j^* = f_j + \sum_{i=1}^{N} \langle Z_{ij} \rangle \frac{\bar{\alpha}_j + \bar{\beta}_j}{\bar{\lambda}_j + \sum_{d=1}^{D} X_{id}}. \tag{51}$$

In our case, the lower bound $\mathcal{L}(q)$ can be calculated by

$$\begin{aligned}\mathcal{L}(q) &= \sum_{\mathcal{Z}} \int q(\mathcal{Z},\Theta)\log\left\{\frac{p(\mathcal{X},\mathcal{Z},\Theta|\vec{\pi})}{q(\mathcal{Z},\Theta)}\right\}d\Theta \\ &= \langle \log\, p(\mathcal{X}|\mathcal{Z},\Theta)\rangle + \langle \log\, p(\mathcal{Z}|\vec{\pi})\rangle + \langle \log\, p(\Theta)\rangle \\ &- \langle \log q(\mathcal{Z})\rangle - \langle \log\, q(\Theta)\rangle. \end{aligned} \tag{52}$$

To determine the optimal number of mixture components M, we treat the mixing coefficients $\vec{\pi}$ as parameters and estimate values for M by maximizing the lower bound $\mathcal{L}(q)$ with respect to $\vec{\pi}$. Then, we can calculate the optimal values for π_j as

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N} r_{ij}. \tag{53}$$

By deleting the components with mixing coefficients that are close to 0, appropriate number of mixture components will be acquired.

3.3. The Complete Learning Algorithm

The complete algorithm for learning IBL mixture model with VB inference can be summarized as follows In our VB algorithm, it is useful to monitor

Algorithm 2

1: Initialize the number of components M.
2: Initialize values of hyper-parameters u_{jd}, v_{jd}, g_j, h_j, s_j, t_j, c_j, f_j.
3: Initialize the values of r_{ij} by using K-means.
4: **repeat**
5: Variational E-step:
Update the variational factors $q(\mathcal{Z})$, $q(\boldsymbol{\alpha})$, $q(\vec{\alpha})$, $q(\vec{\beta})$ and $q(\vec{\lambda})$.
6: Variational M-step:
Maximize lower bound $\mathcal{L}(q)$ with respect to $\vec{\pi}$ by using Eq. (52).
7: **until** convergence is reached
8: Select the optimal value of M by removing the components with small mixing coefficients (less than 10^{-5}).

the variational lower bound $\mathcal{L}(q)$ (Eq. (52)) during the re-estimation, which

contributes to testing for convergence. That is, we can evaluate the lower bound $L(q)$ at each iteration, and terminate the learning process if $L(q)$ does not increase significantly.

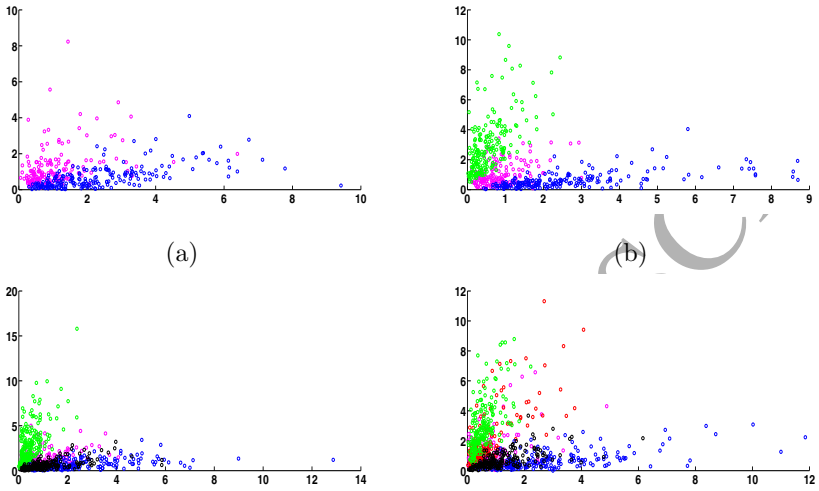


Figure 2: The two-dimensional synthetic data sets. (a) Data set 1 (D1); (b) Data set 2 (D2); (c) Data set 3 (D3); (d) Data set 4 (D4).

4. Experiment

In this section, we test the effectiveness of our two proposed methods including the MML-based finite IBL Mixture Model(MML-IBLMM) and the variational finite IBL Mixture Model (Var-IBLMM), through synthetic data sets and real-world applications. In the experiments of synthetic data, we compare the accuracy on learning IBL mixture model in terms of estimating model parameters and selecting the right number of components using MML-IBLMM and var-IBLMM, respectively. In the experiments regarding real-world applications, we demonstrate the merits of MML-IBLMM and Var-IBLMM on clustering by comparing them with several other well-defined mixture models, such

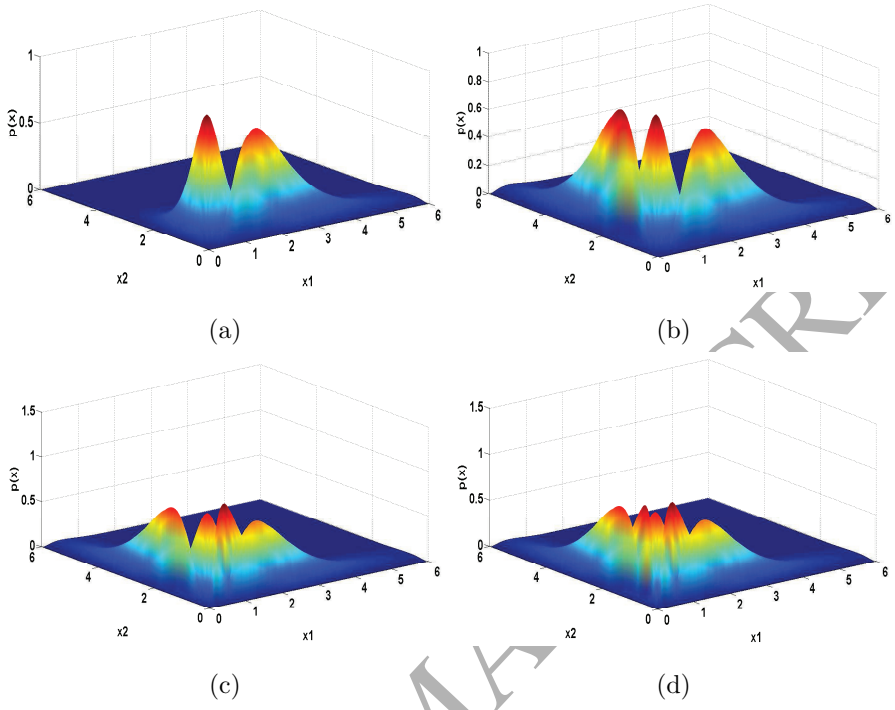


Figure 3: Probability densities for the two-dimensional synthetic data sets. (a) Data set 1 (D1), (b) Data set 2 (D2), (c) Data set 3 (D3), (d) Data set 4 (D4).

as MML-based finite Gaussian mixture model(MML-GMM) [32], variational finite Gaussian mixture model (Var-GMM) [33], MML-based finite Beta-Liouville Mixture Model (MML-BLMM)[34] and variational finite Beta-Liouville Mixture Model (Var-BLMM)[35]. In our experiments, we initialize the number of mixture components to 15 ($M = 15$), and set other hyperparameters as $(u_{jd}, v_{jd}, g_j, h_j, s_j, t_j, c_j, f_j) = (1, 0.1, 1, 0.1, 1, 0.1, 1, 0.1, 1, 0.1)$.

4.1. Synthetic Data

In this part, we provide the performance of both MML-IBLMM and Var-IBLMM by testing them on four different 2-dimensional synthetic data sets which are obtained by using Gibbs sampler. Specifically, we evaluate the effectiveness for estimating parameters and selecting the right components of

Table 1: Parameters for generating the synthetic data sets D1 $\sim$ D4. N denotes the total number of data points, N_j denotes the number of elements in cluster j.

	N_j	j	α_{j1}	α_{j2}	α_j	β_j	λ_j	π_j
D1	100	1	3	4	10	6	1	0.25
	300	2	15	3	7	7.5	3	0.75
D2	150	1	3	4	10	6	1	0.25
	150	2	15	3	7	7.5	3	0.25
	300	3	3	20	6	10	5	0.50
D3	160	1	3	4	10	6	1	0.20
	160	2	15	3	7	7.5	3	0.20
	200	3	3	20	6	10	5	0.25
	280	4	8	4	4	8	3	0.35
D4	200	1	3	4	10	6	1	0.20
	200	2	15	3	7	7.5	3	0.20
	200	3	3	20	6	10	5	0.20
	200	4	3	9	5	3.5	1.5	0.20
	200	5	8	4	4	8	3	0.20

mixture model between MML-IBLMM and Var-IBLMM algorithms. Table 1 presents the real parameters for generating the four synthetic data sets. The synthetic data sets and their corresponding probability densities can be viewed in Fig. 2 and Fig. 3, respectively.

The estimated parameters obtained by MML-IBLMM and Var-IBLMM based on 20 runs are shown in Table 2 and Table 3, respectively. As we can observe from this table, our proposed two algorithms can accurately estimate parameters of these four synthetic data sets.

Fig. 4 shows the message length values of different number of mixture components for each synthetic data set obtained by using MML-IBLMM. We can observe that the correct number of mixture components is obtained with the maximum of message length values (i.e., $M = 2$ in Data set 1, $M = 3$ in Data set

Table 2: The estimated parameters obtained by MML-IBLMM.

	j	α_{j1}^*	α_{j2}^*	α_j^*	β_j^*	λ_j^*	π_j^*
D1	1	2.91	3.87	9.70	5.82	1.00	0.250
	2	14.4	3.09	6.81	7.26	2.91	0.750
D2	1	2.88	3.85	10.40	5.80	0.97	0.243
	2	14.55	2.89	7.31	7.20	2.92	0.253
	3	2.89	20.8	6.25	9.68	4.825	0.504
D3	1	2.89	3.84	10.35	5.82	1.04	0.208
	2	15.46	2.88	6.78	7.25	3.08	0.205
	3	2.89	20.82	6.21	9.65	5.175	0.240
	4	8.33	4.14	4.13	7.67	2.89	0.347
D4	1	2.90	3.86	10.33	5.83	0.96	0.205
	2	15.38	2.89	6.75	6.75	3.09	0.195
	3	3.112	20.81	6.51	10.38	5.19	0.192
	4	2.94	9.27	4.45	3.395	1.455	0.203
	5	8.32	3.86	4.08	7.87	2.92	0.205

2, $M = 4$ in Data set 3, $M = 5$ in Data set 4). Therefore, we have proved here that MML criterion is a useful tool to discover the correct number of mixture components in mixture modeling.

In our VB learning algorithm, the number of mixture components is obtained by removing the components with the estimated mixing coefficients that are close to 0. We can verify this result according to the variational likelihood bound calculated by Eq. (52). The idea is that the variational likelihood bound should be maximum at the correct number of components. Fig. 5 shows the results of variational likelihood bounds for different data sets obtained by Var-IBLMM, by varying the number of mixture components from 1 to 10. As we can see from this figure, for each data set, we have received the correct number of mixture components at the maximum value of the variational likelihood bound.

It is convenient for Var-IBLMM to determine its convergence by inspecting

Table 3: The estimated parameters obtained by Var-IBLMM.

	j	α_{j1}^{**}	α_{j2}^{**}	α_{j}^{**}	β_{j}^{**}	λ_{j}^{**}	π_{j}^{**}
D1	1	2.93	3.92	9.78	5.88	1.00	0.254
	2	15.21	3.11	6.84	7.35	3.09	0.746
D2	1	2.92	3.84	9.80	5.83	0.98	0.245
	2	14.65	2.90	7.29	7.18	2.89	0.253
	3	3.10	20.65	6.18	9.75	4.83	0.502
D3	1	3.08	4.15	10.21	5.88	0.97	0.204
	2	14.68	2.92	6.88	7.31	3.11	0.196
	3	3.11	20.75	6.20	9.73	5.17	0.245
	4	8.23	3.88	4.14	7.77	2.88	0.355
D4	1	3.09	4.15	9.79	6.11	0.98	0.197
	2	15.28	2.93	7.12	7.21	3.12	0.198
	3	3.08	20.47	5.86	10.28	5.16	0.199
	4	2.93	9.75	4.85	3.509	1.502	0.202
	5	8.15	3.91	4.16	7.89	2.97	0.204

the variational lower bound. It converges to local optimal solution by employing the VB algorithm. Var-IBLMM can directly calculate the closed form solution by using approximate calculation and variational methods in comparison with MML-IBLMM which uses traditional EM and MML criteria.

4.2. Text Categorization

In this section, We test our two algorithms on a challenging real application namely text categorization. The main purpose of text categorization is to automatically assign documents into semantic clusters. Thus, it is vital to choose a reasonable model which can correctly describe the statistical characteristics of text data. In our experiment, to better describe and cluster the documents, we employ the bag-of-words (BOW) model to convert documents into feature vectors via calculating their (term frequencyinverse document frequency) (TFIDF) scores. In our BOW model, we have the assumption that documents can be

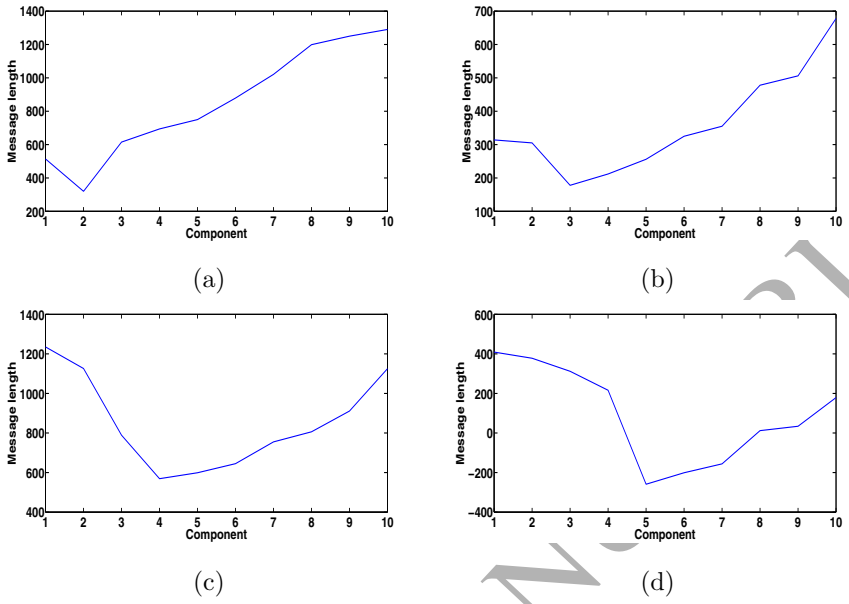


Figure 4: The values of message length of different number of mixture components for each synthetic data set obtained by MML-IBLMM. (a) D1, (b) D2, (c) D3, (d) D4.

composed into a series of words regardless of grammar and words that are independent between each other.

We conduct the experiment on text classification using the "ModApte" data set, which is a subset of Reuters-21578 data set [1]. In our case, the "ModApt" data set that contains 12,902 documents which are grouped into 135 valid topics and mainly aim at the top 10 frequent categories including "earn", "acq", "money-fx", "grain", "crude", "trade", "interest", "ship", "wheat" and "corn". In our experiment, all of processed positive vectors are modeled by the proposed finite IBL mixture model and grouped into homogeneous classes based on ML and VB learning approaches. The categorization performance is evaluated by different measures includes Error rate, Recall rate, Precision and F_1, which are widely used in information retrieval.

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

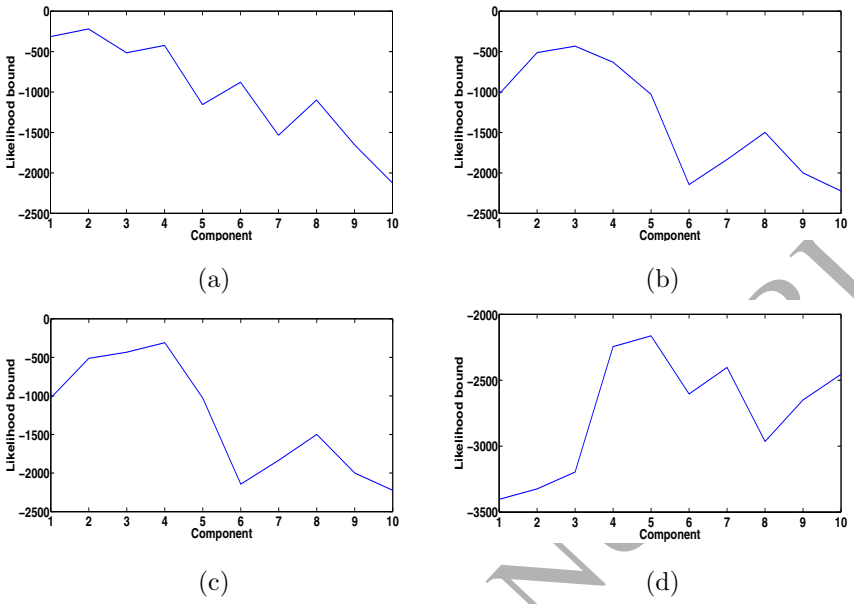


Figure 5: The variational likelihood bound of different number of mixture components by Var-IBLMM. (a) D1, (b) D2, (c) D3, (d) D4.

The average categorization results based on 30 runs are provided in Table 4 for MML-IBLMM and Table 5 for Var-IBLMM, respectively. For comparison, the average categorization performance by different approaches are presented in Fig. 6. As shown in Fig. 6, Var-IBLMM and MML-IBLMM perform better than Var-BLMM, MML-BLMM, Var-GMM and MML-GMM in terms of higher precision, Recall and F_1 scores, and lower error rates (specifically, $2.76 \pm 0.21, 3.21 \pm 0.15$ *vs* $3.54 \pm 0.18, 3.72 \pm 0.20, 6.37 \pm 0.31, 6.80 \pm 0.35$). Furthermore, Var-IBLMM achieves higher accuracy rate and lower error rate than MML-IBLMM (see clearly Table 4 and Table 5), which demonstrated the advantages of using VB algorithm to learn mixture models than ML algorithm. Moreover, it can be seen that GMM and BLMM based on Var or MML provided the worst performance among other all tested approaches. This result also shows that Gaussian distribution and Beta-Liouville are not the optimal choice for dealing with positive high-dimensional vector.

Table 4: Average categorization performance (%) by MML-IBLMM for the "ModApte" data set.

Categories	Error	Recall	Precision	F_1
earn	4.94±0.39	92.56±1.11	93.55±0.38	91.14±0.21
acq	5.46±0.33	88.13±2.01	90.62±0.25	88.30±0.31
money-fx	4.50±0.32	65.82±0.89	81.31±0.33	75.20±1.00
grain	2.46±0.20	76.24±1.14	93.64±0.51	87.00±0.84
crude	2.36±0.14	75.54±3.13	90.23±1.21	84.90±0.55
trade	3.44±0.19	62.74±3.72	85.74±2.01	84.50±1.33
interest	3.90±0.17	56.21±0.74	79.31±1.17	66.00±0.63
ship	2.00±0.19	49.65±1.56	91.56±0.57	66.25±2.19
wheat	1.60±0.09	66.56±2.61	93.14±0.97	74.10±1.30
corn	1.44±0.13	56.14±0.27	94.55±1.14	72.34±0.93

4.3. Diagnosis of Coronary Artery Disease

Coronary artery disease (CAD) is quite common and is one of the main factors which lead to death. Therefore, it is imperative to have an effective in-time diagnosis of CAD. In this section, we test the developed finite IBL mixture model on this challenging application. We evaluate the performance of our mixture model with the proposed two learning approaches on a data set known as the Z-Alizadeh Sani data set [2]. The Z-Alizadeh Sani data set includes the records of 303 patients, with 54 features attached to each patient. These features can act as the indicator for CAD of the patient [36]. These 54 features can be divided into four groups: demographic, symptom and examination, ECG, and laboratory and echo features. Each patient could be possibly classified into CAD or Normal. The patient will be classified as CAD, if the diameter narrowing is over or equal to 50%. Otherwise, the patient is considered as Normal [36]. In

[2]https://archive.ics.uci.edu/ml/datasets/Z-Alizadeh+Saniand

Table 5: Average categorization performance (%) by Var-IBLMM for the "ModApte"data set.

Categories	Error	Recall	Precision	F_1
earn	4.23±0.35	93.33±0.95	94.01±0.13	91.89±0.33
acq	4.86±0.51	91.79±1.21	92.56±0.15	89.00±0.26
money-fx	3.91±0.32	67.51±1.17	83.06±1.05	76.55±0.63
grain	2.01±0.21	77.51±2.30	94.50±1.71	89.62±0.87
crude	2.00±0.16	75.96±3.11	90.89±0.64	85.34±1.21
trade	3.21±0.36	63.61±0.87	88.51±0.88	86.80±0.41
interest	3.35±0.22	57.86±1.16	82.23±1.03	69.78±0.88
ship	1.73±0.22	52.06±2.41	92.33±2.22	70.98±2.03
wheat	1.25±0.13	66.98±0.97	93.97±1.40	75.00±1.41
corn	1.01±0.07	59.76±1.16	95.25±0.78	73.95±1.64

our experiment, we transform some features into Integer or Real values. For instance, "YES" is converted into 1 and "NO" is forced into 0. The resulting feature vectors are then normalized and thus result in positive vectors. In our case, a confusion matrix is considered to measure the performance of the proposed approach as shown in Table 6. In our two-class problem (CAD and Normal), there are 4 different types of measures which are defined as follows: true positive (TP) denotes a patient who suffers from CAD, false positive (FP) denotes a patient who is diagnosed with CAD but is indeed Normal, true negatives (TN) denotes a patient who is Normal, false negatives (FN) represents a patient who is indeed Normal but is incorrectly diagnosed with CAD.

Table 6: Confusion matrix for the Z-Alizadeh Sani data set.

	CAD	Normal
CAD	True positive (TP)	False positive (FP)
Normal	False negative (FN)	True negative (TN)

To better evaluate the performance of the proposed MML-IBLMM and Var-

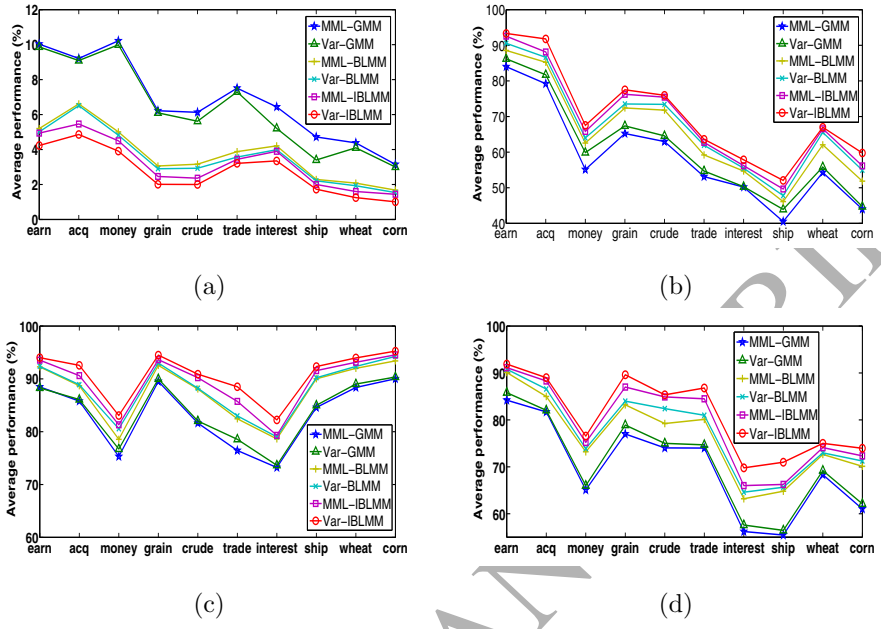


Figure 6: Average categorization performance for different algorithms in terms of (a) Error, (b) Recall, (c) Precision and (d) F_1.

IBLMM, we also adopt three other measures including Accuracy, Sensitivity and Specificity [37], based on the confusion matrix as illustrated in Table 6. The calculations of Accuracy, Sensitivity and Specificity are respectively given by

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}}, \tag{54}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{55}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}. \tag{56}$$

Table 7 shows the classification results for the Z-Alizadeh Sani data set in terms of the confusion matrix by different approaches including Var-IBLMM, MML-IBLMM, Var-BLMM, MML-BLMM, Var-GMM and MML-GMM. After obtaining this confusion matrix, we can calculate the Accuracy, Sensitivity

and Specificity as presented in Table 8. As we can see from these two tables, the classification results obtained by Var-IBLMM and MML-IBLMM are better than the ones based on the Var-BLMM, MML-BLMM, Var-GMM and MML-GMM. According to Table 8, Var-IBLMM achieves the highest accuracy rate (81.84%) while MML-IBLMM also provides the competitive accuracy rate (79.21%). However, the accuracy of MML-GMM is considerably low, which is only 63.04%. These results prove that the IBL distribution possesses better statistical characteristics for clustering positive vectors than the Beta-Liouville and Gaussian distributions do. Also, Var-IBLMM outperforms the MML-IBLMM which again demonstrates the fact that VB inference may provide a better model learning performance than the ML approach does.

Table 7: The confusion matrices for different algorithms.

Algorithms		CAD	Normal
Var-IBLMM	CAD	178	25
	Normal	30	70
MML-IBLMM	CAD	165	29
	Normal	34	75
Var-BLMM	CAD	160	28
	Normal	41	74
MML-BLMM	CAD	151	15
	Normal	59	78
Var-GMM	CAD	132	46
	Normal	60	65
MML-GMM	CAD	129	33
	Normal	69	62

Another interesting observation is that, most tested approaches have the values of sensitivity that are higher than those of specificity, with the exception of MML-BLMM (71.90% vs 83.87%) and MML-GMM (65.15% vs 65.26%). As a result, these two algorithms are more prone to identify patients as Normal rather

Table 8: Results of different algorithms for the Z-Alizadeh Sani data set (%).

Algorithms	Accuracy	Sensitivity	Specificity
Var-IBLMM	**81.84**	**85.58**	73.68
MML-IBLMM	79.21	82.91	72.82
Var-BLMM	77.23	79.60	72.54
MML-BLMM	75.57	71.90	**83.87**
Var-GMM	65.02	68.75	58.56
MML-GMM	63.04	65.15	65.26

than CAD. The rest of the approaches are more inclined to predict patients have CAD compared with MML-BLMM and MML-GMM.

4.4. Software Modules Categorization

Classification of software modules is currently an important area in system engineering. This research field has also been extended other important fields in system engineering [38]. One of the most challenging task is to develop and maintain a software system, which still has a number of obstacles. A lot of relatively independent units called modules (i.e. a set of source-code files) that execute one function are included in software. In this section, we conduct experiments on a data set namely MIS data set [39], which is a widely utilized commercial software including 4500 routines written with about 400,000 lines of codes in the form of Pascal, FORTRAN, and PL/M assembly code. Our goal is to predict the types of modules (i.e. fault-prone or nonfault-prone). The MIS data set in our experiment consists of 390 modules (modules 1-114 are thought as nonfault-prone, the remaining modules are regarded as fault-prone) during three-years system testing and maintenance. Then, in order to analyze the data, each module can be described by 11 complexity metrics as variables [10]. In our experiment, we also give four different types of measures which are defined as: true positive (TP) denotes a nonfault-prone module classified as a nonfault

prone module, true negative(TN) denotes a fault-prone module classified as a fault-prone module, false negative (FN)denotes a fault-prone module wrongly classified as a nonfault-prone module and false positive (FP) denotes a nonfault-prone module mistakenly classified as a fault-prone module. (see Table 9).

Table 9: The confusion matrix for the MIS data set.

	Nonfault-prone (NF)	Fault-prone (F)
Nonfault-prone (NF)	True Positive (TP)	False positive (FP)
Fault-prone (F)	False negative (FN)	True negative (TN)

The main goal of this experiment is to test and compare the performance of clustering for Var-IBLMM, MML-IBLMM as well as Var-BLMM, MML-BLMM, Var-GMM, MML-GMM. The results of confusion matrix for these methods are presented in Table 10. Then, we calculate the responding accuracy, Sensitivity and Specificity which can be seen in Table 11.

Table 10: The confusion matrices for different algorithms.

Algorithms		NF	F
Var-IBLMM	NF	99	15
	F	59	215
MML-IBLMM	NF	100	14
	F	76	200
Var-BLMM	NF	96	18
	F	74	202
MML-BLMM	NF	94	20
	F	87	189
Var-GMM	NF	107	7
	F	157	119
MML-GMM	NF	105	9
	F	159	117

Table 11: Results of different algorithms for the MIS data set (%).

Algorithms	Accuracy	Sensitivity	Specificity
Var-IBLMM	**80.51**	**62.66**	93.48
MML-IBLMM	76.92	56.82	93.46
Var-BLMM	76.41	56.47	90.90
MML-BLMM	72.56	51.93	90.43
Var-GMM	57.95	40.53	**94.44**
MML-GMM	56.92	39.78	92.86

From the Table 11, it can be seen that the clustering results based on IBLMM including Var-IBLMM (80.51%) and MML-IBLMM (76.92%) are more accurate than that based on BLMM and GMM due to the fact that IBL distribution can give better performance to model the positive vector data. Also, the Var-GMM achieves the higher specificity (94.44%), which shows that the classification based on GMM is prone to identify one software module as fault-prone than nonfault-prone.

5. Conclusion

In this work, we have proposed a novel statistical approach for clustering multivariate positive data based on a finite mixture model of IBL distributions. We have developed two approaches to learn the proposed IBL mixture model based on ML and VB learning algorithms. In ML learning algorithm, the right number of mixture components is determined according to the minimum message length criterion. In VB learning, the parameters of the model and the number of mixture components can be determined simultaneously in a unified framework, without the requirement of using information criteria. The effectiveness of our model has been tested though extensive experiments involving both synthetic data sets and real applications such as text category, CAD diagnosis and classification of software modules. One potential future work could

be devoted to the integration of feature selection into the proposed IBL mixture model to improve clustering performance.

Appendix A. Proofs of Eq. (8)and Eq. (9)

Because $\vec{Z}_i$ uses a 1-of-K representation, we can get the distribution in the form as follow

$$p(\vec{Z}_i) = \prod_{j=1}^{M} \pi_j^{Z_{ij}}. \tag{A.1}$$

Similarity, the conditional distribution of $\vec{X}_i$ given Z_{ij} is distribution as follow

$$p(\vec{X}_i|Z_{ij}=1) = p(\vec{X}_i|\theta_j), \tag{A.2}$$

which can also take the form

$$p(\vec{X}_i|\vec{Z}_i) = \sum_{j=1}^{M} p(\vec{X}_i|\theta_j)^{Z_{ij}}. \tag{A.3}$$

Obviously, the complete data set $\{\mathcal{X}, \mathcal{Z}\}$ can be obtained from both Eq. (A.1) and Eq. (A.3) as

$$p(\mathcal{X}, \mathcal{Z}|\vec{\pi}, \Theta) = \prod_{i=1}^{N}\prod_{j=1}^{M} \pi_j^{Z_{ij}} p(\vec{X}_i|\theta_j)^{Z_{ij}}. \tag{A.4}$$

Then, we can have the log-likelihood function of the complete data set $\{\mathcal{X}, \mathcal{Z}\}$ as follow

$$\Phi(\mathcal{X}, \mathcal{Z}|\vec{\pi}, \Theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{ij}\{\log\pi_j + \log\, p(\vec{X}_i|\theta_j)\}. \tag{A.5}$$

Next, as you can see from the reference [24] in chapter 9.3.1, the expectation of the complete-data log likelihood function is therefore obtained by

$$\Omega(\mathcal{X}|\Theta) = E_{\mathcal{Z}}\big[\Phi(\mathcal{X}, \mathcal{Z}|\vec{\pi}, \Theta)\big] = \sum_{i=1}^{N}\sum_{j=1}^{M} \langle Z_{ij}\rangle\{\log\pi_j + \log\, p(\vec{X}_i|\theta_j)\}. \tag{A.6}$$

where $E_{\mathcal{Z}}[\cdot]$ denotes the operation of expectation for the variable $\mathcal{Z}$.

Appendix B. Inverse of Hessian Matrix $H(\theta_j^{(t)})^{-1}$

The second derivatives of $\Omega(\mathcal{X}|\Theta)$ with respect to parameters Θ are

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial^2\alpha_j} = [\Psi'(\alpha_j+\beta_j)-\Psi'(\alpha_j)]\sum_{i=1}^{N}\langle Z_{ij}\rangle, \tag{B.1}$$

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial^2\beta_j} = [\Psi'(\alpha_j+\beta_j)-\Psi'(\beta_j)]\sum_{i=1}^{N}\langle Z_{ij}\rangle, \tag{B.2}$$

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial^2\lambda_j} = \sum_{i=1}^{N}\langle Z_{ij}\rangle\Big[-\frac{\beta_j}{\lambda_j^2}+\frac{\alpha_j+\beta_j}{(\lambda_j+\sum_{d=1}^{D}X_{id})^2}\Big], \tag{B.3}$$

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\alpha_{jd_1}\partial\alpha_{jd_2}} = \begin{cases} \Big[\Psi'(\sum_{d=1}^{D}\alpha_{jd})-\Psi'(\alpha_{jd})\Big]\sum_{i=1}^{N}\langle Z_{ij}\rangle, & \text{if } d_1=d_2 \\ \Psi'(\sum_{d=1}^{D}\alpha_{jd})\sum_{i=1}^{N}\langle Z_{ij}\rangle, & \text{otherwise} \end{cases} \tag{B.4}$$

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\beta_j} = \Psi'(\alpha_j+\beta_j)\sum_{i=1}^{N}\langle Z_{ij}\rangle, \tag{B.5}$$

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\alpha_{jd}} = \frac{\partial^2\Omega(\chi|\Theta)}{\partial\beta_j\partial\alpha_{jd}} = \frac{\partial^2\Omega(\chi|\Theta)}{\partial\lambda_j\partial\alpha_{jd}} = 0, \tag{B.6}$$

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\lambda_j} = -\sum_{i=1}^{N}\langle Z_{ij}\rangle\frac{1}{\lambda_j+\sum_{d=1}^{D}X_{id}}, \tag{B.7}$$

$$\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\beta_j\partial\lambda_j} = \sum_{i=1}^{N}\langle Z_{ij}\rangle\Big[\frac{1}{\lambda_j}-\frac{1}{\lambda_j+\sum_{d=1}^{D}X_{id}}\Big]. \tag{B.8}$$

The Hessian matrix can be expressed as a block-diagonal structure

$$H(\theta_j) = \text{BlockDiag}\{H(\alpha_j,\beta_j,\lambda_j), H(\alpha_{j1},\cdots,\alpha_{jD})\}, \tag{B.9}$$

where

$$H(\alpha_j,\beta_j,\lambda_j) = \begin{bmatrix} \frac{\partial^2\Omega(\chi|\Theta)}{\partial^2\alpha_j} & \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\beta_j} & \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\lambda_j} \\ \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\beta_j\partial\alpha_j} & \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial^2\beta_j} & \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\beta_j\partial\lambda_j} \\ \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\lambda_j\alpha_j} & \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial\lambda_j\beta_j} & \frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial^2\partial\lambda_j} \end{bmatrix}, \tag{B.10}$$

with

$$H(\alpha_{j1},\cdots,\alpha_{jD})=\frac{\partial^2\Omega(\mathcal{X}|\Theta)}{\partial^2\alpha_{jd_1}\partial^2\alpha_{jd_2}}. \tag{B.11}$$

The $H(\alpha_{j1},\cdots,\alpha_{jD})$ can also be written in the form [40]

$$H(\alpha_{j1},\cdots,\alpha_{jD})=G_j+\gamma_j\vec{c}_j*\vec{c}_j^{\,T}, \tag{B.12}$$

where

$$G_j=\mathrm{diag}\Big[-\Psi'(\alpha_{j1})\sum_{i=1}^{N}\langle Z_{ij}\rangle,\cdots,-\Psi'(\alpha_{jD})\sum_{i=1}^{N}\langle Z_{ij}\rangle\Big], \tag{B.13}$$

with

$$\vec{c}_j^{\,T}=(c_{j1},\quad\cdots,c_{jD}),\quad c_{jd}=1,\quad d=1,\cdots,D, \tag{B.14}$$

and

$$\gamma_j=\Psi'\Big(\sum_{d=1}^{D}\alpha_{jd}\Big)\sum_{i=1}^{N}\langle Z_{ij}\rangle,\quad \text{if}\quad \gamma_j\neq\Big(\sum_{d=1}^{D}\frac{c_{jd}^2}{G_{dd}}\Big)^{-1}. \tag{B.15}$$

Then the inverse of Hessian Matrix can be written in the form of a block-diagonal structure as

$$H(\theta_j)^{-1}=\mathrm{BlockDiag}\{H(\alpha_j,\beta_j,\lambda_j)^{-1},H(\alpha_{j1},\cdots,\alpha_{jD})^{-1}\} \tag{B.16}$$

where the inverse of Matrix $H(\alpha_{j1},\cdots,\alpha_{jD})$ takes the form as follows [40]

$$H(\alpha_{j1},\cdots,\alpha_{jD})^{-1}=G_j^*+\gamma_j^*\vec{c}_j^{\,*}\vec{c}_j^{\,*T}, \tag{B.17}$$

where

$$G_j^*=G_j^{-1}=\mathrm{diag}\Big[\frac{1}{-\Psi'(\alpha_{j1})\sum_{i=1}^{N}\langle Z_{ij}\rangle},\cdots,\frac{1}{-\Psi'(\alpha_{jD})\sum_{i=1}^{N}\langle Z_{ij}\rangle}\Big] \tag{B.18}$$

with

$$\vec{c}_j^{\,*T}=\mathrm{diag}\Big[\frac{1}{-\Psi'(\alpha_{j1})\sum_{i=1}^{N}\langle Z_{ij}\rangle},\cdots,\frac{1}{-\Psi'(\alpha_{jD})\sum_{i=1}^{N}\langle Z_{ij}\rangle}\Big], \tag{B.19}$$

and

$$\gamma_j^*=\Psi'\Big(\sum_{d=1}^{D}\alpha_{jd}\Big)\sum_{i=1}^{N}\langle Z_{ij}\rangle\Big(1+\Psi^*\Big(\alpha_{jd}\Big)\sum_{i=1}^{N}\langle Z_{ij}\rangle\sum_{d=1}^{D}\frac{1}{-\Psi^*(\alpha_{jd})\sum_{i=1}^{N}\langle Z_{ij}\rangle}\Big). \tag{B.20}$$

Appendix C. Proof of Eq. (18)

Based on [41], the complete-data Fisher matrix is shown as

$$|F(\Theta,\vec{\pi})| \simeq |F(\vec{\pi})| \prod_{j=1}^{M} |F(\theta_j)|, \tag{C.1}$$

where $|F(\vec{\pi})|$ denotes the Fisher information with respects to mixing parameters which is given by

$$|F(\vec{\pi})| = \frac{N^{M-1}}{\prod_{j=1}^{M} \pi_j}, \tag{C.2}$$

and $|F(\theta_j)|$ in Eq. (C.1) represents the Fisher information with respects to parameter vector θ_j of the IBL distribution. To calculate $|F(\theta_j)|$, We assume that for the jth cluster $\mathcal{X}_j = (\vec{X}_l, \cdots, \vec{X}_{l+n_j-1})$, where $l \leq N$, n_j represents the number of elements in cluster j. Thus, the negative likelihood function about the jth cluster takes the form

$$\Phi(\mathcal{X}_j|\theta_j) = -\log(p(\mathcal{X}_j|\theta_j)) = -\sum_{i=l}^{l+n_j-1} \log(p(\vec{X}_i|\theta_j)). \tag{C.3}$$

Then, we can obtain the second and mixed derivatives of $\Phi(\mathcal{X}_j|\theta_j)$ with respect to parameters θ_j as follows

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial^2\alpha_j} = -n_j[\Psi'(\alpha_j+\beta_j) - \Psi'(\alpha_j)], \tag{C.4}$$

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial^2\beta_j} = -n_j[\Psi'(\alpha_j+\beta_j) - \Psi'(\beta_j)] \tag{C.5}$$

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial^2\lambda_j} = -n_j\Big[-\frac{\beta_j}{\lambda_j^2} + \frac{\alpha_j+\beta_j}{(\lambda_j+\sum_{d=1}^{D} X_{id})^2},\Big], \tag{C.6}$$

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial\alpha_{jd_1}\partial\alpha_{jd_2}} = \begin{cases} -\Big[\Psi'(\sum_{d=1}^{D}\alpha_{jd}) - \Psi'(\alpha_{jd})\Big]n_j, & \text{if } d_1 = d_2 \\ -\Psi'(\sum_{d=1}^{D}\alpha_{jd})n_j, & \text{otherwise} \end{cases} \tag{C.7}$$

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial\alpha_j\partial\beta_j} = -\Psi'(\alpha_j+\beta_j)n_j, \tag{C.8}$$

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial\alpha_j\partial\alpha_{jd}} = \frac{\partial^2\Omega(\chi|\Theta)}{\partial\beta_j\partial\alpha_{jd}} = \frac{\partial^2\Omega(\chi|\Theta)}{\partial\lambda_j\partial\alpha_{jd}} = 0, \tag{C.9}$$

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial\alpha_j\partial\lambda_j} = n_j\frac{1}{\lambda_j + \sum_{d=1}^{D} X_{id}}, \tag{C.10}$$

$$\frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial\beta_j\partial\lambda_j} = -n_j\left[\frac{1}{\lambda_j} - \frac{1}{\lambda_j + \sum_{d=1}^{D} X_{id}}\right]. \tag{C.11}$$

Next, the Hessian matrix related to the jth cluster can also be expressed as a block-diagonal structure, such that

$$\tilde{H}(\alpha_j, \beta_j, \lambda_j) = \begin{bmatrix} \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial^2\alpha_j} & \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial\alpha_j\partial\beta_j} & \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial\alpha_j\partial\lambda_j} \\ \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial\beta_j\partial\alpha_j} & \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial^2\beta_j} & \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial\beta_j\partial\lambda_j} \\ \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial\lambda_j\alpha_j} & \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial\lambda_j\beta_j} & \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j}{\partial^2\partial\lambda_j} \end{bmatrix}, \tag{C.12}$$

with

$$\tilde{H}(\alpha_{j1}, \cdots, \alpha_{jD}) = \frac{\partial^2\Phi(\mathcal{X}_j|\theta_j)}{\partial\alpha_{jd_1}\partial\alpha_{jd_2}}. \tag{C.13}$$

Then, we can rewrite the $\tilde{H}(\alpha_{j1}, \cdots, \alpha_{jD})$ as

$$\tilde{H}(\alpha_{j1}, \cdots, \alpha_{jD}) = R_j + \rho_j\vec{b}_j * \vec{b}_j{}^T, \tag{C.14}$$

where

$$R_j = \text{diag}\left[\Psi'(\alpha_{j1})n_j, \cdots, \Psi'(\alpha_{jD})n_j\right], \tag{C.15}$$

with

$$\vec{b}_j^T = (b_{j1}, \quad \cdots, b_{jD}), \quad b_{jd} = 1, \quad d = 1, \cdots, D, \tag{C.16}$$

and

$$\rho_j = -\Psi'\Big(\sum_{d=1}^{D}\alpha_{jd}\Big)n_j. \tag{C.17}$$

Then, based on the theorem of matrix as described in [40], we can have

$$\begin{aligned} |\tilde{H}(\alpha_{j1}, \cdots, \alpha_{jD})| &= \left(1 + \rho_j\sum_{d=1}^{D}\frac{b_d^2}{R_{dd}}\right)\prod_{d=1}^{D}R_{dd} \\ &= \left(1 - \Psi'(\sum_{d=1}^{D}\alpha_{jd})\sum_{d=1}^{D}\frac{1}{\Psi'(\alpha_{jd})}\right)n_j^D\prod_{d=1}^{D}\Psi'(\alpha_{jd}). \end{aligned} \tag{C.18}$$

By combining Eq. (C.12) and Eq. (C.18), we obtain

$$|F(\theta_j)| = |\tilde{H}(\theta_j)| = |\tilde{H}(\alpha_j, \beta_j, \lambda_j)| \times |\tilde{H}(\alpha_{j1}, \cdots, \alpha_{jD})|. \tag{C.19}$$

Hence, by substituting Eq. (C.19) and Eq. (C.2) into Eq. (C.1), we have

$$\begin{aligned} |F(\Theta, \vec{\pi})| \simeq & \frac{N^{M-1}}{\prod_{j=1}^{M} \pi_j} \prod_{j=1}^{M} |\tilde{H}(\alpha_j, \beta_j, \lambda_j)| \\ & \times \left(1 - \Psi'(\sum_{d=1}^{D} \alpha_{jd}) \sum_{d=1}^{D} \frac{1}{\Psi'(\alpha_{jd})}\right) n_j^D \prod_{d=1}^{D} \Psi'(\alpha_{jd}). \end{aligned}$$

Appendix D. Proofs of Eq. (19)

For $h(\Theta, \vec{\pi})$, due to the fact that the mixing vector $\vec{\pi}$ and parameters Θ are independent, thus we can have

$$\begin{aligned} h(\Theta, \vec{\pi}) = h(\vec{\pi})h(\Theta) = h(\vec{\pi}) \prod_{j=1}^{M} h(\theta_j) = h(\vec{\pi}) \prod_{j=1}^{M} \left[h(\alpha_j, \beta_j, \lambda_j) \prod_{d=1}^{D} h(\alpha_{jd}) \right] \\ = h(\vec{\pi}) \prod_{j=1}^{M} \left[h(\alpha_j)h(\beta_j)h(\lambda_j) \prod_{d=1}^{D} h(\alpha_{jd}) \right]. \end{aligned} \tag{D.1}$$

Since $\sum_{j=1}^{M} \pi_j = 1$, the prior for $\vec{\pi}$ is naturally a Dirichlet distribution that is given by

$$h(\vec{\pi}) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} \pi_j^{\eta_j - 1}, \tag{D.2}$$

where $\vec{\eta} = (\eta_1, \ldots, \eta_M)$ represent parameters of the Dirichlet distribution. Following [42], we can set $\eta_1, \cdots, \eta_M = 1$ and then the prior is uniform and is given by

$$h(\vec{\pi}) = (M-1)!. \tag{D.3}$$

For $h(\alpha_j, \beta_j, \lambda_j)$, we choose the uniform distributions for α_j, β_j and λ_j over $\left[0, e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\alpha}_j}\right]$, $\left[0, e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\beta}_j}\right]$ and $\left[0, e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\lambda}_j}\right]$, respectively, where we can find $e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\alpha}_j} > \alpha_j$, $e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\beta}_j} > \beta_j$ and $e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\lambda}_j} > \lambda_j$, where the hat notation represents estimated parameter. Then we have

$$h(\alpha_j) = \left[e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\alpha}_j} \right]^{-1}, \tag{D.4}$$

$$h(\beta_j) = \left[e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\beta}_j} \right]^{-1}, \tag{D.5}$$

$$h(\lambda_j) = \left[e^6 \frac{\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j}{\hat{\lambda}_j} \right]^{-1}. \tag{D.6}$$

As a result, we can obtain

$$h(\alpha_j, \beta_j, \lambda_j) = h(\alpha_j)h(\beta_j)h(\lambda_j) = \left[\frac{e^{18}(\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j)^3}{\hat{\alpha}_j \hat{\beta}_j \hat{\lambda}_j} \right]^{-1}. \tag{D.7}$$

Similarly, we choose the uniform distribution for α_{jd} over $\left[0, e^6 \frac{\sum_{d=1}^{D} \hat{\alpha}_{jd}}{\hat{\alpha}_{jd}}\right]$ based on the fact that $\alpha_{jd} < e^6 \frac{\sum_{d=1}^{D} \hat{\alpha}_{jd}}{\hat{\alpha}_{jd}}$. Then, we can have

$$h(\alpha_{jd}) = \left[e^6 \frac{\sum_{d=1}^{D} \hat{\alpha}_{jd}}{\hat{\alpha}_{jd}} \right]^{-1}. \tag{D.8}$$

Finally, by substituting Eq. (D.3), Eq. (D.7) and Eq. (D.8) into Eq. (D.1), we obtain

$$h(\Theta) = (M-1)! \prod_{j=1}^{M} \left[\left(e^6 \frac{(\hat{\alpha}_j + \hat{\beta}_j + \hat{\lambda}_j)^3}{\hat{\alpha}_j \hat{\beta}_j \hat{\lambda}_j} \right)^{-3} \prod_{d=1}^{D} e^6 \frac{\sum_{d=1}^{D} \hat{\alpha}_{jd}}{\hat{\alpha}_{jd}} \right]. \tag{D.9}$$

ACKNOWLEDGEMENT

The completion of this work was supported by the National Natural Science Foundation of China (61502183,61673186), and the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (ZQN-PY510).

References

[1] C. Hu, W. Fan, J. Du, Y. Zeng, Model-based segmentation of image data using spatially constrained mixture models, Neurocomputing 283 (2018) 214–227.

[2] A. K. Jain, R. P. W. Duin, J. Mao, Statistical pattern recognition: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.

[3] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.

[4] G. J. McLachlan, D. Peel, Finite Mixture Models, New York: Wiley, 2000.

[5] S. Boutemedjet, D. Ziou, N. Bouguila, Model-based subspace clustering of non-gaussian data, Neurocomputing 73 (10) (2010) 1730–1739.

[6] Y. Lai, Y. Ping, K. Xiao, B. Hao, X. Zhang, Variational bayesian inference for a dirichlet process mixture of beta distributions and application, Neurocomputing 278 (2018) 23–33.

[7] G. Zhou, D. Zhu, Y. Wei, Z. Wang, Y. Zhou, Real-time online learning of gaussian mixture model for opacity mapping, Neurocomputing 211 (2016) 212–220.

[8] W. Fan, N. Bouguila, D. Ziou, Variational learning of finite Dirichlet mixture models using component splitting, Neurocomputing 129 (2014) 3–16.

[9] W. Fan, N. Bouguila, Online variational learning of generalized Dirichlet mixture models with feature selection, Neurocomputing 126 (2014) 166–179.

[10] T. Bdiri, N. Bouguila, Positive vectors clustering using inverted Dirichlet finite mixture models, Expert Systems with Applications 39 (2) (2012) 1869–1882.

[11] T. Bdiri, N. Bouguila, Bayesian learning of inverted Dirichlet mixtures for SVM kernels generation, Neural Computing and Applications 23 (5) (2013) 1443–1458.

[12] K. T. Fang, S. Kotz, K. W. Ng, Symmetric Multivariate and Related Distributions, Chapman and Hall, 1990.

[13] S. Ganesalingam, Classification and mixture approaches to clustering via maximum likelihood, Journal of the Royal Statistical Society 38 (3) (1989) 455–466.

[14] G. J. Mclachlan, T. Krishnan, The EM algorithm and extensions, Biometrics 382 (1) (1997) 154–156.

[15] H. Akaike, A New Look at the Statistical Model Identification, Springer New York, 1974.

[16] G. Schwarz, Estimating dimension of a model, Annals of Statistics (6) (1978) 461464.

[17] J. Rissanen, Modeling by shortest data description, Pergamon Press, Inc., 1978.

[18] C. S. Wallace, D. M. Boulton, An information measure for classification, Computer Journal 11 (2) (1968) 185–194.

[19] M. A. T. Figueiredo, A. K. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 381–396.

[20] N. Bouguila, D. Ziou, High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1716–1731.

[21] N. Bouguila, D. Ziou, Unsupervised selection of a finite Dirichlet mixture model: An MML-based approach, IEEE Trans. Knowl. Data Eng. 18 (8) (2006) 993–1009.

[22] H. Attias, A variational Bayesian framework for graphical models, in: International Conference on Neural Information Processing Systems, 1999, pp. 209–215.

[23] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, L. K. Saul, An introduction to variational methods for graphical models, Machine Learning 37 (2) (1999) 183–233.

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

[25] S. Konishi, G. Kitagawa, Information Criteria and Statistical Modeling, Springer New York, 2008.

[26] C. S. Wallace, Statistical and Inductive Inference by Minimum Message Length, Springer-Verlag New York, 2005.

[27] C. E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (4) (1948) 623–656.

[28] W. Fan, N. Bouguila, D. Ziou, Variational learning for finite Dirichlet mixture models and applications., IEEE Transactions on Neural Networks and Learning Systems 23 (5) (2012) 762–774.

[29] C. M. Bishop, N. Lawrence, T. Jaakkola, M. I. Jordan, Approximating posterior distributions in belief networks using mixtures, in: Conference on Advances in Neural Information Processing Systems, 1998, pp. 416–422.

[30] N. D. Lawrence, C. M. Bishop, M. I. Jordan, Mixture representations for inference and learning in boltzmann machines, in: Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 320–327.

[31] M. M. Ichir, A. Mohammad-Djafari, A mean field approximation approach to blind source separation with lp priors, in: 13th European Signal Processing Conference, 2005, pp. 1–4.

[32] P. Kasarapu, L. Allison, Minimum message length estimation of mixtures of multivariate gaussian and von mises-fisher distributions", journal=Machine Learning, year=2015, volume=100, number=2, pages=333–378,.

[33] N. Nasios, A. G. Bors, Variational learning for gaussian mixture models, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36 (4) (2006) 849–862.

[34] N. Bouguila, Hybrid generative/discriminative approaches for proportional data modeling and classification, IEEE Trans. Knowl. Data Eng. 24 (12) (2012) 2184–2202.

[35] W. Fan, N. Bouguila, Learning finite Beta-Liouville mixture models via variational Bayes for proportional data clustering, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), 2013, pp. 1323–1329.

[36] D. Mann, D. Zipes, P. Libby, R. Bonow, Braunwald's heart disease : a textbook of cardiovascular medicine, 10th Edition, Elsevier, 2014.

[37] R. Alizadehsani, J. Habibi, M. J. Hosseini, H. Mashayekhi, R. Boghrati, A. Ghandeharioun, B. Bahadorian, Z. A. Sani, A data mining approach for diagnosis of coronary artery disease, Computer Methods and Programs in Biomedicine 111 (1) (2013) 52–61.

[38] N. D. Singpurwalla, S. P. Wilson, Software reliability modeling, International Statistical Review 62 (3) (1994) 289–317.

[39] M. R. Lyu, Handbook of software reliability engineering, McGraw-Hill, Inc., 1996.

[40] F. A. Graybill, Matrices With Applications in Statistics, Wadsworth, 1983.

[41] C. S. Wallace, D. L. Dowe, Mml mixture modelling of multi-state, poisson, von mises circular and gaussian distributions, in: Proc. 6th Int. Workshop on Artif. Intelligence and Statistics, 1997, pp. 529–536.

[42] R. A. Baxter, J. J. Oliver, Finding overlapping components with MML, Statistics and Computing 10 (1) (2000) 5–16.

Can Hu received his BE degree from the Department of Computer Science and Technology, Hunan Institute of Science and Technology, Hunan province, China, in 2014. He is currently working for his MSc degree in the Department of Computer Science and Technology, Huaqiao University, Xiamen, China. His research areas include machine learning and pattern recognition, image processing, and bioinformatics.

Wentao Fan received his MSc and PhD degrees in electrical and computer engineering from Concordia University, Montreal, Quebec, Canada, in 2009 and 2014, respectively. He is currently an associate professor in the Department of Computer Science and Technology, Huaqiao University, Xiamen, China. His research interests include machine learning, computer vision, and pattern recognition.

Ji-Xiang Du received his MSc degree in vehicle engineering from Hefei University of Technology in 2002 and his PhD in pattern recognition intelligent system from the University of Science and Technology of China. Currently, he is a professor at the College of Computer Science and Technology, Huaqiao University. He is also the associate dean of the College of Computer Science and Technology. His current research mainly concerns pattern recognition and machine learning.

Nizar Bouguila received the engineer degree from the University of Tunis in 2000, the M.Sc. and Ph.D. degrees from Sherbrooke University in 2002 and 2006, respectively, all in computer science. He is currently a Professor with the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Montreal, QC, Canada. His research interests include image processing, machine learning, 3D graphics, computer vision, and pattern recognition.