Neural Process Lett (2016) 44:431–449
DOI 10.1007/s11063-015-9466-x

Model-Based Clustering Based on Variational Learning of Hierarchical Infinite Beta-Liouville Mixture Models

Wentao Fan[1] · Nizar Bouguila[2]

Published online: 15 August 2015

Abstract In this work, we develop a statistical framework for data clustering which uses hierarchical Dirichlet processes and Beta-Liouville distributions. The parameters of this framework are leaned using two variational Bayes approaches. The first one considers batch settings and the second one takes into account the dynamic nature of real data. Experimental results based on a challenging problem namely visual scenes categorization demonstrate the merits of the proposed framework.

Keywords Mixture models · Beta-Liouville · Variational Bayes · Nonparametric Bayesian · Hierarchical Dirichlet process · Visual scenes categorization

1 Introduction

Model-based clustering has been the topic of extensive research in the past [3]. It is very essential and critical to many computer vision, image processing, data mining and pattern recognition applications [12,15,36,45]. In these applications finite Gaussian mixture models have been widely used as a formal approach to clustering [23,30,38,47]. Unfortunately, the inadequacy of the Gaussian model has been apparent in various applications. Recently, several contributions have occurred to propose other finite mixture models, in order to overcome the limitations related to the Gaussian assumption, and a number of authors began to pay attention to indications of non-Gaussian behaviour in real data [1,6,31,58]. For instance, the authors in [9,10,20] have proposed the consideration of the finite Dirichlet mixture which offers high

✉ Wentao Fan
fwt@hqu.edu.cn

Nizar Bouguila
nizar.bouguila@concordia.ca

1 Department of Computer Science and Technology, Huaqiao University, Xiamen, China

2 The Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada

flexibility and ease of use. This mixture has been successfully applied recently to a variety of challenging problems (see, for instance, [19,52,53]). However, it has some drawbacks, such as its restrictive negative covariance structure, as discussed in [6,7] where the author has proposed the finite Beta-Liouville mixture as an efficient alternative and has discussed central issues related to the adoption of this mixture namely parameters estimation and model selection. The finite Beta-Liouville mixture has been shown to be an efficient alternative to the Gaussian and the Dirichlet especially when dealing with proportional data (e.g. normalized histograms). It has met with significant success in numerous real-world applications such as dynamic textures categorization, human activities modeling and recognition, and facial expressions recognition [17,18].

One of the most challenging problems regarding finite mixture modeling is to determine the appropriate number of mixture components. This difficulty can be tackled by assuming that there is an infinite number of components through Dirichlet process [21,26] as done in [8]. One useful extension of the conventional Dirichlet process framework is to place a Bayesian hierarchy on it, resulting in the so-called hierarchical Dirichlet process, where the base distribution of the Dirichlet process is itself distributed according to another Dirichlet process. Recently, the hierarchical Dirichlet process framework has shown promising results dealing with problems of modeling grouped data where observations are organized into groups and these groups are statistically linked by sharing mixture components [54,55]. Thus, the first contribution of this paper is to go a step further by extending our previous works about the Beta-Liouville mixture via the consideration of a nonparametric Bayesian approach based on hierarchical Dirichlet process [54,55]. The resulting model is learned via a variational Bayes framework that we have developed by considering batch settings. A major goal of machine learning techniques is to provide systems that can improve their performance as they observe new data or information [13,22]. In practice, since data often arrive sequentially in time, one is interested in performing learning on-line. Therefore, our second contribution is to extend the proposed batch algorithm to online settings. We validate both algorithms using a challenging application namely visual scenes categorization.

The rest of this paper is organized as follows. Section 2 describes the hierarchical Dirichlet process mixture model of Beta-Liouville distributions. Both the batch and online variational approaches to learn the proposed model are developed in Section 3. Section 4 is devoted to the experimental results. Finally, concluding remarks are given in Section 5.

2 Model Specification

In this section, first, we present an overview of the finite Beta-Liouville distribution; then, we introduce the framework of hierarchical Dirichlet process mixture model; finally, we propose the hierarchical infinite Beta-Liouville mixture model.

2.1 Finite Beta-Liouville Mixture Model

If a D-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_D)$ is distributed according to a Beta-Liouville distribution, then its probability density function (pdf) is defined by [7]:

$$\mathrm{BL}(\mathbf{X}|\boldsymbol{\theta}) = \frac{\Gamma\left(\sum_{l=1}^{D}\alpha_l\right)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\prod_{l=1}^{D}\frac{X_l^{\alpha_l-1}}{\Gamma(\alpha_l)}\left(\sum_{l=1}^{D}X_l\right)^{\alpha-\sum_{l=1}^{D}\alpha_l}\left(1-\sum_{l=1}^{D}X_l\right)^{\beta-1} \tag{1}$$

where $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_D, \alpha, \beta)$ is the vector of parameters of the Beta-Liouville distribution. For a random vector $\mathbf{X}$ which is generated from a finite Beta-Liouville mixture model with M components, we have

$$p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathrm{BL}(\mathbf{X}|\boldsymbol{\theta}_k) \tag{2}$$

where $\boldsymbol{\theta}_k = (\alpha_{k1}, \ldots, \alpha_{kD}, \alpha_k, \beta_k)$ is the vector of parameters of the Beta-Liouville distribution $\mathrm{BL}(\cdot)$ corresponding to component k. The vector of mixing coefficients $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ is subject to the constraints: $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

2.2 Hierarchical Dirichlet Process Mixture Model

A two-level hierarchical Dirichlet process is defined as the following: Given a grouped data set with M groups where each group is associated with a Dirichlet process G_j, and the indexed set of Dirichlet processes $\{G_j\}$ shares a base distribution G_0 which is itself distributed according to a Dirichlet process with the concentration parameter γ and base distribution H:

$$\begin{aligned} &G_0 \sim \mathrm{DP}(\gamma, H) \\ &G_j \sim \mathrm{DP}(\lambda, G_0) \qquad \text{for each } j,\ j \in \{1, \ldots, M\} \end{aligned} \tag{3}$$

where j is the index for each group of data. Please notice that the above hierarchy can be readily extended to have more than two levels.

In our work, we construct the hierarchical Dirichlet process using the stick-breaking construction [24,51]. In the global-level, the global measure G_0 is distributed according to the Dirichlet process $\mathrm{DP}(\gamma, H)$ as

$$\begin{aligned} G_0 = \sum_{k=1}^{\infty} \psi_k \delta_{\Omega_k} \qquad &\psi_k = \psi'_k \prod_{s=1}^{k-1} (1 - \psi'_s) \\ \psi'_k \sim \mathrm{Beta}(1, \gamma) \qquad &\Omega_k \sim H \end{aligned} \tag{4}$$

where δ_{Ω_k} is an atom at Ω_k, and $\{\Omega_k\}$ is a set of independent random variables drawn from H. ψ_k is the stick-breaking variable and satisfies $\sum_{k=1}^{\infty} \psi_k = 1$. Since G_0 is the base distribution of the Dirichlet processes G_j, then the atoms Ω_k are shared among all $\{G_j\}$ and are only differ in weights based on the property of Dirichlet process [55].

For each group-level Dirichlet process G_j, we also apply the conventional stick-breaking representation according to [57] as

$$\begin{aligned} G_j = \sum_{t=1}^{\infty} \pi_{jt} \delta_{\varpi_{jt}} \qquad &\pi_{jt} = \pi'_{jt} \prod_{s=1}^{t-1} (1 - \pi'_{js}) \\ \pi'_{jt} \sim \mathrm{Beta}(1, \lambda) \qquad &\varpi_{jt} \sim G_0 \end{aligned} \tag{5}$$

where $\delta_{\varpi_{jt}}$ are group-level atoms at ϖ_{jt}, $\{\pi_{jt}\}$ is a set of stick-breaking weights which satisfies $\sum_{t=1}^{\infty} \pi_{jt} = 1$. Since ϖ_{jt} is distributed according to the base distribution G_0, it takes on the value Ω_k with probability ψ_k.

Next, we introduce a binary latent variable $W_{jtk} \in \{0, 1\}$ as an indicator variable: $W_{jtk} = 1$ if ϖ_{jt} maps to the global-level atom Ω_k; otherwise, $W_{jtk} = 0$. Thus, we have $\varpi_{jt} = \Omega_k^{W_{jtk}}$. As a result, group-level atoms ϖ_{jt} do not need to be explicitly represented. Since $\boldsymbol{\psi}$ is a function of $\boldsymbol{\psi}'$ according to the stick-breaking construction of the Dirichlet process as shown in Eq. (4), the indicator variable $\mathbf{W} = (W_{jt1}, W_{jt2}, \ldots)$ is distributed as

$$p(\mathbf{W}|\boldsymbol{\psi}') = \prod_{j=1}^{M}\prod_{t=1}^{\infty}\prod_{k=1}^{\infty}\big[\psi'_k\prod_{s=1}^{k-1}(1-\psi'_s)\big]^{W_{jtk}} \tag{6}$$

According to Eq. (4), the prior distribution of $\boldsymbol{\psi}'$ is a specific Beta distribution as

$$p(\boldsymbol{\psi}') = \prod_{k=1}^{\infty}\text{Beta}(1,\gamma_k) = \prod_{k=1}^{\infty}\gamma_k(1-\psi'_k)^{\gamma_k-1} \tag{7}$$

For the grouped data set $\mathcal{X} = (X_{j1},\ldots X_{jN})$, let i index the observations within each group j. We assume that each variable ζ_{ji} is a factor corresponding to an observation X_{ji}, and the factors $\boldsymbol{\theta}_j = (\zeta_{j1},\zeta_{j2},\ldots)$ are distributed according to the Dirichlet process G_j, one for each j. Thus, we can write the likelihood function in the form

$$\begin{aligned}\zeta_{ji}\mid G_j &\sim G_j\\ X_{ji}\mid\zeta_{ji} &\sim F(\zeta_{ji})\end{aligned} \tag{8}$$

where $F(\zeta_{ji})$ denotes the distribution of the observation X_{ji} given ζ_{ji}, and the base distribution H of G_0 provides the prior distribution for the factors ζ_{ji}. This framework is known as the hierarchical Dirichlet process mixture model, in which each group is associated with an infinite mixture model, and the mixture components are shared among these mixture models due to the sharing of atoms Ω_k among all $\{G_j\}$.

Since each factor ζ_{ji} is distributed according to G_j based on Eq. (8), it takes the value ϖ_{jt} with probability π_{jt}. It is convenient to place a binary indicator variable $Z_{jit}\in\{0,1\}$ for ζ_{ji}. That is, $Z_{jit}=1$ if ζ_{ji} is associated with component t and maps to the group-level atom ϖ_{jt}; otherwise, $Z_{jit}=0$. Therefore, we have $\zeta_{ji}=\varpi_{jt}^{Z_{jit}}$. Because ϖ_{jt} also maps to the global-level atom Ω_k as we mentioned previously, we then have $\zeta_{ji}=\varpi_{jt}^{Z_{jit}}=\Omega_k^{W_{jtk}Z_{jit}}$. Since $\boldsymbol{\pi}$ is a function of $\boldsymbol{\pi}'$ according to the stick-breaking construction as shown in Eq. (5), the indicator variable $\mathbf{Z}=(Z_{ji1},Z_{ji2},\ldots)$ is distributed as

$$p(\mathbf{Z}|\boldsymbol{\pi}') = \prod_{j=1}^{M}\prod_{i=1}^{N}\prod_{t=1}^{\infty}[\pi'_{jt}\prod_{s=1}^{t-1}(1-\pi'_{js})]^{Z_{jit}} \tag{9}$$

The prior distribution of $\boldsymbol{\pi}'$ is a specific Beta distribution in the form

$$p(\boldsymbol{\pi}') = \prod_{j=1}^{M}\prod_{t=1}^{\infty}\text{Beta}(1,\lambda_{jt}) = \prod_{j=1}^{M}\prod_{t=1}^{\infty}\lambda_{jt}(1-\pi'_{jt})^{\lambda_{jt}-1} \tag{10}$$

2.3 Hierarchical Infinite Beta-Lioouville Mixture Model

In this subsection, we propose the hierarchical Dirichlet process mixture model of Beta-Liouville distributions, which may also be referred to as the hierarchical infinite Beta-Liouville mixture model. Given a grouped data set $\mathcal{X}$ with M groups, it is assumed that each D-dimensional data vector $\mathbf{X}_{ji}=(X_{ji1},\ldots,X_{jiD})$ is drawn from a hierarchical infinite Beta-Liouville mixture model where j is the index for each group of the data. Then the likelihood function of the proposed hierarchical infinite Beta-Liouville mixture model with latent variables can be written as

$$p(\mathcal{X}) = \prod_{j=1}^{M}\prod_{i=1}^{N}\prod_{t=1}^{\infty}\prod_{k=1}^{\infty}\big[\text{BL}(\mathbf{X}_{ji}|\boldsymbol{\theta}_k)\big]^{Z_{jit}W_{jtk}} \tag{11}$$

where $\boldsymbol{\theta}_k = (\alpha_{k1}, \ldots, \alpha_{kD}, \alpha_k, \beta_k)$ is the vector of parameters of the Beta-Liouville distribution.

Next, we need to place conjugate priors over parameters α_l, α and β. Since α_l, α and β are positive, Gamma distributions $\mathcal{G}(\cdot)$ are adopted to approximate conjugate priors for these parameters. Then, we have the following prior distributions for α_l, α and β, respectively:

$$p(\alpha_l|u_l, v_l) = \mathcal{G}(\alpha_l|u_l, v_l) = \frac{v_l^{u_l}}{\Gamma(u_l)}\alpha_l^{u_l-1}e^{-v_l\alpha_l} \tag{12}$$

$$p(\alpha|g, h) = \mathcal{G}(\alpha|g, h) = \frac{h^g}{\Gamma(g)}\alpha^{g-1}e^{-h\alpha} \tag{13}$$

$$p(\beta|g', h') = \mathcal{G}(\beta|g', h') = \frac{h'^{g'}}{\Gamma(g')}\beta^{g'-1}e^{-h'\beta} \tag{14}$$

where $\mathcal{G}(\cdot)$ is the Gamma distribution with positive parameters.

3 Model Learning Via Variational Bayes

3.1 Batch Variational Inference for Hierarchical Infinite Beta-Liouville Mixture Model

In this subsection, we develop a batch variational Bayes framework [2,4] for learning the hierarchical infinite Beta-Liouville mixture model. To simplify notations, we define $\Theta = \{\mathbf{Z}, \mathbf{W}, \boldsymbol{\psi}', \boldsymbol{\pi}', \boldsymbol{\theta}\}$ as the set of latent variables and unknown random variables. The main idea of variational learning is to calculate an approximation $q(\Theta)$ for the true posterior distribution $p(\Theta|\mathcal{X})$ by maximizing the lower bound of the logarithm of the marginal likelihood $p(\mathcal{X})$, which is given by

$$\mathcal{L}(q) = \int q(\Theta) \ln[p(\mathcal{X}, \Theta)/q(\Theta)]d\Theta \tag{15}$$

In this work, we adopt the factorial approximation [4] (or mean fields approximation), which has been successfully applied in the past to complex models involving incomplete data, to factorize $q(\Theta)$ into disjoint tractable factors. We also apply the truncation technique as described in [5] to truncate the variational approximations of global- and group-level Dirichlet processes at levels K and T as

$$\psi'_K = 1, \quad \sum_{k=1}^{K} \psi_k = 1, \quad \psi_k = 0 \ \text{ when } \ k > K \tag{16}$$

$$\pi'_{jT} = 1, \quad \sum_{t=1}^{T} \pi_{jt} = 1, \quad \pi_{jt} = 0 \ \text{ when } \ t > T \tag{17}$$

where the truncation levels K and T are variational parameters which can be freely initialized and will be optimized automatically during the learning process.

In variational Bayes learning, the general expression for the optimal solution to a specific variational factor $q_s(\Theta_s)$ is given by [4]:

$$q_s(\Theta_s) = \frac{\exp\langle \ln p(\mathcal{X}, \Theta)\rangle_{i\neq s}}{\int \exp\langle \ln p(\mathcal{X}, \Theta)\rangle_{i\neq s} d\Theta} \tag{18}$$

where $\langle\cdot\rangle_{i\neq s}$ is the expectation with respect to all the distributions of $q_i(\Theta_i)$ except for $i = s$. Therefore, we can obtain the variational solution for each factor as

$$q(\mathbf{Z}) = \prod_{j=1}^{M}\prod_{i=1}^{N}\prod_{t=1}^{T}\rho_{jit}^{Z_{jit}} \tag{19}$$

$$q(\mathbf{W}) = \prod_{j=1}^{M}\prod_{t=1}^{T}\prod_{k=1}^{K}\vartheta_{jtk}^{W_{jtk}} \tag{20}$$

$$q(\boldsymbol{\alpha}_l) = \prod_{k=1}^{K}\prod_{l=1}^{D}\mathcal{G}(\alpha_{kl}|\tilde{u}_{kl}, \tilde{v}_{kl}) \tag{21}$$

$$q(\boldsymbol{\alpha}) = \prod_{k=1}^{K}\mathcal{G}(\alpha_k|\tilde{g}_k, \tilde{h}_k) \tag{22}$$

$$q(\boldsymbol{\beta}) = \prod_{k=1}^{K}\mathcal{G}(\beta_k|\tilde{g}'_k, \tilde{h}'_k) \tag{23}$$

$$q(\boldsymbol{\pi}') = \prod_{j=1}^{M}\prod_{t=1}^{T}\mathrm{Beta}(\pi'_{jt}|a_{jt}, b_{jt}) \tag{24}$$

$$q(\boldsymbol{\psi}') = \prod_{k=1}^{K}\mathrm{Beta}(\psi'_k|c_k, d_k) \tag{25}$$

where the corresponding hyperparameters in the above equations can be calculated by

$$\rho_{jit} = \frac{\exp(\widetilde{\rho}_{jit})}{\sum_{s=1}^{T}\exp(\widetilde{\rho}_{jis})} \tag{26}$$

$$\begin{aligned}\widetilde{\rho}_{jit} = \sum_{k=1}^{K}\langle W_{jtk}\rangle\Bigg[&\widetilde{\mathcal{I}}_k + \widetilde{\mathcal{H}}_k + \left(\bar{\alpha}_k - \sum_{l=1}^{D}\bar{\alpha}_{kl}\right)\ln\left(\sum_{l=1}^{D}X_{jil}\right) + \sum_{l=1}^{D}(\bar{\alpha}_{kl}-1)\ln X_{jil}\\ &+ (\bar{\beta}_k - 1)\ln\left(1 - \sum_{l=1}^{D}X_{jil}\right)\Bigg] + \langle\ln\pi'_{jt}\rangle + \sum_{s=1}^{t-1}\langle\ln(1-\pi'_{js})\rangle\end{aligned} \tag{27}$$

$$\vartheta_{jtk} = \frac{\exp(\widetilde{\vartheta}_{jtk})}{\sum_{s=1}^{K}\exp(\widetilde{\vartheta}_{jts})} \tag{28}$$

$$\begin{aligned}\widetilde{\vartheta}_{jtk} = \sum_{i=1}^{N}\langle Z_{jit}\rangle\Bigg[&\widetilde{\mathcal{I}}_k + \widetilde{\mathcal{H}}_k + \left(\bar{\alpha}_k - \sum_{l=1}^{D}\bar{\alpha}_{kl}\right)\ln\left(\sum_{l=1}^{D}X_{jil}\right) + \sum_{l=1}^{D}(\bar{\alpha}_{kl}-1)\ln X_{jil}\\ &+ (\bar{\beta}_k - 1)\ln\left(1 - \sum_{l=1}^{D}X_{jil}\right)\Bigg] + \langle\ln\psi'_k\rangle + \sum_{s=1}^{k-1}\langle\ln(1-\psi'_s)\rangle\end{aligned} \tag{29}$$

$$a_{jt} = 1 + \sum_{i=1}^{N}\langle Z_{jit}\rangle, \qquad b_{jt} = \lambda_{jt} + \sum_{i=1}^{N}\sum_{s=t+1}^{T}\langle Z_{jis}\rangle \tag{30}$$

$$c_k = 1 + \sum_{j=1}^{M}\sum_{t=1}^{T}\langle W_{jtk}\rangle, \quad d_k = \gamma_k + \sum_{j=1}^{M}\sum_{t=1}^{T}\sum_{s=k+1}^{K}\langle W_{jts}\rangle \tag{31}$$

$$\tilde{u}_{kl} = u_{kl} + \sum_{j=1}^{M}\sum_{t=1}^{T}\langle W_{jtk}\rangle \sum_{i=1}^{N}\langle Z_{jit}\rangle \bar{\alpha}_{kl}\Bigg[\Psi\left(\sum_{l=1}^{D}\bar{\alpha}_{kl}\right) - \Psi(\bar{\alpha}_{kl})$$
$$+ \Psi'\left(\sum_{l=1}^{D}\bar{\alpha}_{kl}\right)\sum_{s\neq l}^{D}(\langle \ln \alpha_{ks}\rangle - \ln \bar{\alpha}_{ks})\bar{\alpha}_{ks}\Bigg] \tag{32}$$

$$\tilde{v}_{kl} = v_{kl} - \sum_{j=1}^{M}\sum_{t=1}^{T}\langle W_{jtk}\rangle \sum_{i=1}^{N}\langle Z_{jit}\rangle \left[\ln X_{jil} - \ln\left(\sum_{l=1}^{D} X_{jil}\right)\right] \tag{33}$$

$$\tilde{g}_{k} = g_{k} + \sum_{j=1}^{M}\sum_{t=1}^{T}\langle W_{jtk}\rangle \sum_{i=1}^{N}\langle Z_{jit}\rangle \bar{\alpha}_{k}\big[\Psi(\bar{\alpha}_{k} + \bar{\beta}_{k})$$
$$- \Psi(\bar{\alpha}_{k}) + \bar{\beta}_{k}\Psi'(\bar{\alpha}_{k} + \bar{\beta}_{k})(\langle \ln \beta_{k}\rangle - \ln \bar{\beta}_{k})\big] \tag{34}$$

$$\tilde{h}_{k} = h_{k} - \sum_{j=1}^{M}\sum_{t=1}^{T}\langle W_{jtk}\rangle \sum_{i=1}^{N}\langle Z_{jit}\rangle \ln\left(\sum_{l=1}^{D} X_{jil}\right) \tag{35}$$

$$\tilde{g}'_{k} = g'_{k} + \sum_{j=1}^{M}\sum_{t=1}^{T}\langle W_{jtk}\rangle \sum_{i=1}^{N}\langle Z_{jit}\rangle \bar{\beta}_{k}\big[\Psi(\bar{\alpha}_{k} + \bar{\beta}_{k})$$
$$- \Psi(\bar{\beta}_{k}) + \bar{\alpha}_{k}\Psi'(\bar{\alpha}_{k} + \bar{\beta}_{k})(\langle \ln \alpha_{k}\rangle - \ln \bar{\alpha}_{k})\big] \tag{36}$$

$$\tilde{h}'_{k} = h'_{k} - \sum_{j=1}^{M}\sum_{t=1}^{T}\langle W_{jtk}\rangle \sum_{i=1}^{N}\langle Z_{jit}\rangle \ln\left(1 - \sum_{l=1}^{D} X_{jil}\right) \tag{37}$$

where $\Psi(\cdot)$ is the digamma function. $\widetilde{\mathcal{I}}_k$ and $\widetilde{\mathcal{H}}_k$ in Eqs. (27) and (29) are the lower bounds of $\mathcal{I}_k = \left\langle \ln \frac{\Gamma(\sum_{l=1}^{D}\alpha_{kl})}{\prod_{l=1}^{D}\Gamma(\alpha_{kl})}\right\rangle$ and $\mathcal{H}_k = \left\langle \ln \frac{\Gamma(\alpha_k+\beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)}\right\rangle$, respectively. Since these expectations are computationally intractable, we apply the second-order Taylor series expansion to calculate their lower bounds. The expected values in the above formulas are defined as

$$\bar{\alpha}_{kl} = \frac{\tilde{u}_{kl}}{\tilde{v}_{kl}}, \quad \bar{\alpha}_{k} = \frac{\tilde{g}_{k}}{\tilde{h}_{k}}, \quad \bar{\beta}_{k} = \frac{\tilde{g}'_{k}}{\tilde{h}'_{k}} \tag{38}$$

$$\langle Z_{jit}\rangle = \rho_{jit}, \quad \langle W_{jtk}\rangle = \vartheta_{jtk}, \quad \big\langle \ln \alpha_{kl}\big\rangle = \Psi(\tilde{u}_{kl}) - \ln \tilde{v}_{kl} \tag{39}$$

$$\big\langle \ln \alpha_{k}\big\rangle = \Psi(\tilde{g}_{k}) - \ln \tilde{h}_{k} \quad \big\langle \ln \beta_{k}\big\rangle = \Psi(\tilde{g}'_{k}) - \ln \tilde{h}'_{k} \tag{40}$$

$$\big\langle \ln \pi'_{jt}\big\rangle = \Psi(a_{jt}) - \Psi(a_{jt} + b_{jt}) \tag{41}$$

$$\big\langle \ln(1 - \pi'_{jt})\big\rangle = \Psi(b_{jt}) - \Psi(a_{jt} + b_{jt}) \tag{42}$$

$$\big\langle \ln \psi'_{k}\big\rangle = \Psi(c_{k}) - \Psi(c_{k} + d_{k}) \tag{43}$$

$$\big\langle \ln(1 - \psi'_{k})\big\rangle = \Psi(d_{k}) - \Psi(c_{k} + d_{k}) \tag{44}$$

Since the update equations for the variational factors are coupled together through the expected values of other factors, the optimization process can be solved in a way analogous to the EM algorithm and the complete algorithm is summarized in Algorithm 1. The convergence of the variational learning algorithm is guaranteed and can be inspected through evaluation of the variational lower bound [4].

Algorithm 1 Batch variational learning of hierarchical infinite Beta-Liouville mixture model

1: Choose the initial truncation levels K and T.
2: Initialize the values for hyperparameters $\lambda_{jt}, \gamma_k, u_{kl}, v_{kl}, g_k, h_k, g'_k, h'_k$.
3: Initialize the value of ρ_{jit} by K-Means algorithm.
4: **repeat**
5: *The variational E-step*:
6: Estimate the expected values in Eqs. (38)–(44), use the current distributions over the model parameters.
7: *The variational M-step*:
8: Update the variational solutions for each factor using Eqs. (19)–(25) and the current values of the moments.
9: **until** Convergence criterion is reached.

3.2 Online Variational Inference for Hierarchical Infinite Beta-Liouville Mixture Model

Compared with batch learning algorithms, online algorithms are more efficient when handling large-scale or sequentially arriving data. In this section, we extend the batch variational framework for learning hierarchical infinite Beta-Liouville mixture model to an online version by adopting the algorithm proposed in [48]. Assume that we have already obtained a data set $\mathcal{X}$ with N data points. In addition, data points are continuously observed in an online manner. Therefore, we have to estimate the variational lower bound corresponding to a fixed amount of data. The expected value of the logarithm of the model evidence $p(\mathcal{X})$ for this finite size of data can be calculated as

$$\langle \ln p(\mathcal{X}) \rangle_\phi = \int \phi(\mathcal{X}) \ln\Big(\int p(X|\Theta) p(\Theta) \mathrm{d}\Theta \Big) \mathrm{d}\mathcal{X} \tag{45}$$

where $\Theta = \{\mathbf{Z}, \mathbf{W}, \boldsymbol{\psi}', \boldsymbol{\pi}', \boldsymbol{\theta}\}$. $\phi(\mathcal{X})$ represents an unknown probability distribution for observed data. Then, the corresponding expected variational lower bound can be calculated as

$$\begin{aligned} \langle \mathcal{L}(q) \rangle_\phi &= \Big\langle \sum_{\mathbf{Z}} \int q(\Lambda) q(\mathbf{Z}) \ln \left[\frac{p(\mathcal{X}, \mathbf{Z}|\Lambda) p(\Lambda)}{q(\Lambda) q(\mathbf{Z})} \right] \mathrm{d}\Lambda \Big\rangle_\phi \\ &= N \int q(\Lambda) \mathrm{d}\Lambda \Big\langle \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left[\frac{p(\mathbf{X}, \mathbf{Z}|\Lambda)}{q(\mathbf{Z})} \right] \Big\rangle_\phi + \int q(\Lambda) \ln \left[\frac{p(\Lambda)}{q(\Lambda)} \right] \mathrm{d}\Lambda \end{aligned} \tag{46}$$

Now consider r as the actual amount of data that we have observed. The corresponding current lower bound for the observed data can be calculated by

$$\mathcal{L}^{(r)}(q) = \frac{N}{r} \sum_{i=1}^{r} \int q(\Lambda) \mathrm{d}\Lambda \sum_{\mathbf{Z}_i} Q(\mathbf{Z}_i) \ln \left[\frac{p(\mathbf{X}_i, \mathbf{Z}_i|\Lambda)}{q(\mathbf{Z}_i)} \right] + \int q(\Lambda) \ln \left[\frac{p(\Lambda)}{q(\Lambda)} \right] \mathrm{d}\Lambda \tag{47}$$

where $\Lambda = \{\mathbf{W}, \boldsymbol{\psi}', \boldsymbol{\pi}', \boldsymbol{\alpha}_l, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$. Please notice that in our case r increases over time while N is fixed. According to [48], the objective function of our online algorithm is the expected log evidence for a fixed amount of data as shown in Eq. (45). Thus, our online learning algorithm calculates the same quantity even if the amount of the observed data increases. As a matter of fact, we use the observed data to improve the estimation quality of the expected variational lower bound Eq. (46) which approximates the expected log evidence Eq. (45).

In order to apply the online variational learning algorithm, we need to successively maximize the current variational lower bound $\mathcal{L}^{(r)}(q)$ with respect to each variational factor.

Assume that we have already observed a data set $\{X_1, \ldots X_{(r-1)}\}$. After obtaining a new observation X_r, we can maximize the current lower bound $\mathcal{L}^{(r)}(q)$ with respect to $q(\mathbf{Z}_r)$ while other variational factors remain fixed to their $(r-1)$th values: $q^{(r-1)}(\mathbf{W})$, $q^{(r-1)}(\boldsymbol{\alpha}_l)$, $q^{(r-1)}(\boldsymbol{\alpha})$, $q^{(r-1)}(\boldsymbol{\beta})$, $q^{(r-1)}(\boldsymbol{\pi}')$ and $q^{(r-1)}(\boldsymbol{\psi}')$. Thus, the variational solution to $q(\mathbf{Z}_r)$ can be updated as

$$q(\mathbf{Z}_r) = \prod_{j=1}^{M}\prod_{t=1}^{T}\rho_{jtr}^{Z_{jtr}} \tag{48}$$

where

$$\rho_{jtr} = \frac{\exp(\widetilde{\rho}_{jtr})}{\sum_{f=1}^{T}\exp(\widetilde{\rho}_{jtr})} \tag{49}$$

and

$$\begin{aligned}\widetilde{\rho}_{jtr} = \sum_{k=1}^{K}\langle W_{jtk}^{(r-1)}\rangle\Bigg[\widetilde{\mathcal{I}}_k^{(r-1)} &+ \left(\bar{\alpha}_k^{(r-1)} - \sum_{l=1}^{D}\bar{\alpha}_{kl}^{(r-1)}\right)\ln\left(\sum_{l=1}^{D}X_{jrl}\right)\\ &+ \sum_{l=1}^{D}\left(\bar{\alpha}_{kl}^{(r-1)} - 1\right)\ln X_{jrl} + \left(\bar{\beta}_k^{(r-1)} - 1\right)\ln\left(1 - \sum_{l=1}^{D}X_{jrl}\right)\\ &+ \widetilde{\mathcal{H}}_k^{(r-1)}\Bigg] + \langle\ln\pi_{jt}'^{(r-1)}\rangle + \sum_{s=1}^{t-1}\langle\ln(1-\pi_{js}'^{(r-1)})\rangle\end{aligned} \tag{50}$$

Next, the current lower bound $\mathcal{L}^{(r)}(q)$ is maximized with respect to to $q^{(r)}(\mathbf{W})$, while $q(\mathbf{Z}_r)$ is fixed and other variational factors remain at their $(r-1)$th values. Thus, the variational factor to $q^{(r)}(\mathbf{W})$ can be updated as

$$q^{(r)}(\mathbf{W}) = \prod_{j=1}^{M}\prod_{t=1}^{T}\prod_{k=1}^{K}\left(\vartheta_{jtk}^{(r)}\right)^{W_{jtk}^{(r)}} \tag{51}$$

where the hyperparameter $\vartheta_{jtk}^{(r)}$ is given by

$$\vartheta_{jtk}^{(r)} = \vartheta_{jtk}^{(r-1)} + \xi_r\Delta\vartheta_{jtk}^{(r)} \tag{52}$$

where ξ_r is the learning rate which is used to reduce the earlier inaccurate estimation effects that contributed to the lower bound and accelerate the convergence of the learning process. In this work, we adopt a learning rate function introduced in [57], such that $\xi_r = (\eta_0 + r)^{-\varsigma}$, subject to the constraints $\varsigma \in (0.5, 1]$ and $\eta_0 \geq 0$. In Eq. (52), $\Delta\vartheta_{jtk}^{(r)}$ is the natural gradient of the hyperparameter $\vartheta_{jtk}^{(r)}$. The natural gradient of a hyperparameter is obtained by multiplying the gradient by the inverse of Riemannian metric, which cancels the coefficient matrix for the posterior parameter distribution. Thus, these natural gradients can be calculated as

$$\Delta\vartheta_{jtk}^{(r)} = \vartheta_{jtk}^{(r)} - \vartheta_{jtk}^{(r-1)} = \frac{\exp\left(\widetilde{\vartheta}_{jtk}^{(r)}\right)}{\sum_{f=1}^{K}\exp\left(\widetilde{\vartheta}_{jtf}^{(r)}\right)} - \vartheta_{jtk}^{(r-1)} \tag{53}$$

where

$$\widetilde{\vartheta}_{jtk}^{(r)} = N\rho_{jtr}\Bigg[\widetilde{\mathcal{I}}_k^{(r-1)} + \left(\bar{\alpha}_k^{(r-1)} - \sum_{l=1}^{D}\bar{\alpha}_{kl}^{(r-1)}\right)\ln\left(\sum_{l=1}^{D}X_{jrl}\right)$$
$$+ \sum_{l=1}^{D}\left(\bar{\alpha}_{kl}^{(r-1)} - 1\right)\ln X_{jrl} + \left(\bar{\beta}_k^{(r-1)} - 1\right)\ln\left(1 - \sum_{l=1}^{D}X_{jrl}\right)$$
$$+ \widetilde{\mathcal{H}}_k^{(r-1)}\Bigg] + \langle \ln \psi_k'^{(r-1)}\rangle + \sum_{s=1}^{k-1}\langle \ln(1-\psi_s'^{(r-1)})\rangle \tag{54}$$

In the following step, the current lower bound $\mathcal{L}^{(r)}(q)$ is maximized with respect to $q^{(r)}(\boldsymbol{\pi}')$, $q^{(r)}(\boldsymbol{\psi}')$, $q^{(r)}(\boldsymbol{\alpha}_l)$, $q^{(r)}(\boldsymbol{\alpha})$ and $q^{(r)}(\boldsymbol{\beta})$ as

$$q^{(r)}(\boldsymbol{\pi}') = \prod_{j=1}^{M}\prod_{t=1}^{T}\mathrm{Beta}\left(\pi_{jt}'^{(r)}|a_{jt}^{(r)}, b_{jt}^{(r)}\right) \tag{55}$$

$$q^{(r)}(\boldsymbol{\psi}') = \prod_{k=1}^{K}\mathrm{Beta}\left(\psi_k'^{(r)}|c_k^{(r)}, d_k^{(r)}\right) \tag{56}$$

$$q^{(r)}(\boldsymbol{\alpha}_l) = \prod_{k=1}^{K}\prod_{l=1}^{D}\mathcal{G}\left(\alpha_{kl}^{(r)}|\tilde{u}_{kl}^{(r)}, \tilde{v}_{kl}^{(r)}\right) \tag{57}$$

$$q^{(r)}(\boldsymbol{\alpha}) = \prod_{k=1}^{K}\mathcal{G}\left(\alpha_k^{(r)}|\tilde{g}_k^{(r)}, \tilde{h}_k^{(r)}\right) \tag{58}$$

$$q^{(r)}(\boldsymbol{\beta}) = \prod_{k=1}^{K}\mathcal{G}\left(\beta_k^{(r)}|\tilde{g}_k'^{(r)}, \tilde{h}_k'^{(r}\right) \tag{59}$$

where the hyperparameters are given by

$$a_{jt}^{(r)} = a_{jt}^{(r-1)} + \xi_r\Delta a_{jt}^{(r)}, \quad b_{jt}^{(r)} = b_{jt}^{(r-1)} + \xi_r\Delta b_{jt}^{(r)} \tag{60}$$
$$c_k^{(r)} = c_k^{(r-1)} + \xi_r\Delta c_k^{(r)}, \quad d_k^{(r)} = d_k^{(r-1)} + \xi_r\Delta d_k^{(r)} \tag{61}$$
$$\tilde{u}_{kl}^{(r)} = \tilde{u}_{kl}^{(r-1)} + \xi_r\Delta\tilde{u}_{kl}^{(r)}, \quad \tilde{v}_{kl}^{(r)} = \tilde{v}_{kl}^{(r-1)} + \xi_r\Delta\tilde{v}_{kl}^{(r)} \tag{62}$$
$$\tilde{g}_k^{(r)} = \tilde{g}_k^{(r-1)} + \xi_r\Delta\tilde{g}_k^{(r)}, \quad \tilde{h}_k^{(r)} = \tilde{h}_k^{(r-1)} + \xi_r\Delta\tilde{h}_k^{(r)} \tag{63}$$
$$\tilde{g}_k'^{(r)} = \tilde{g}_k'^{(r-1)} + \xi_r\Delta\tilde{g}_k'^{(r)}, \quad \tilde{h}_k'^{(r)} = \tilde{h}_k'^{(r-1)} + \xi_r\Delta\tilde{h}_k'^{(r)} \tag{64}$$

The corresponding natural gradients in the above equations can be calculated as

$$\Delta a_{jt}^{(r)} = 1 + N\rho_{jrt} - a_{jt}^{(r-1)} \tag{65}$$

$$\Delta b_{jt}^{(r)} = \lambda_{jt} + N\sum_{s=t+1}^{T}\rho_{jsr} - b_{jt}^{(r-1)} \tag{66}$$

$$\Delta c_k^{(r)} = 1 + \sum_{j=1}^{M}\sum_{t=1}^{T}\vartheta_{jtk}^{(r)} - c_k^{(r-1)} \tag{67}$$

$$\Delta d_k^{(r)} = \gamma_k + \sum_{j=1}^{M}\sum_{t=1}^{T}\sum_{s=k+1}^{K}\vartheta_{jts}^{(r)} - d_k^{(r-1)} \tag{68}$$

$$\Delta\tilde{u}_{kl}^{(r)} = u_{kl} + N\sum_{j=1}^{M}\sum_{t=1}^{T}\vartheta_{jtk}^{(r)}\rho_{jrt}\Bigg[\Psi\left(\sum_{l=1}^{D}\bar{\alpha}_{kl}^{(r-1)}\right) - \Psi\left(\bar{\alpha}_{kl}^{(r-1)}\right) + \Psi'\left(\sum_{l=1}^{D}\bar{\alpha}_{kl}^{(r-1)}\right)\sum_{s\neq l}^{D}\left(\langle\ln\alpha_{ks}^{(r-1)}\rangle - \ln\bar{\alpha}_{ks}^{(r-1)}\right)\bar{\alpha}_{ks}^{(r-1)}\Bigg]\bar{\alpha}_{kl}^{(r-1)} - \tilde{u}_{kl}^{(r-1)} \tag{69}$$

$$\Delta\tilde{v}_{kl}^{(r)} = v_{kl} - N\sum_{j=1}^{M}\sum_{t=1}^{T}\vartheta_{jtk}^{(r)}\rho_{jrt}\left[\ln X_{jrl} - \ln\left(\sum_{l=1}^{D}X_{jrl}\right)\right] - \tilde{v}_{kl}^{(r-1)} \tag{70}$$

$$\Delta\tilde{g}_{k}^{(r)} = g_k + N\sum_{j=1}^{M}\sum_{t=1}^{T}\vartheta_{jtk}^{(r)}\rho_{jrt}\bar{\alpha}_{k}^{(r-1)}\Bigg[\Psi\left(\bar{\alpha}_{k}^{(r-1)} + \bar{\beta}_{k}^{(r-1)}\right) - \Psi(\bar{\alpha}_{k}^{(r-1)}) + \bar{\beta}_{k}^{(r-1)}\Psi'\left(\bar{\alpha}_{k}^{(r-1)} + \bar{\beta}_{k}^{(r-1)}\right)\left(\langle\ln\beta_{k}^{(r-1)}\rangle - \ln\bar{\beta}_{k}^{(r-1)}\right)\Bigg] - \tilde{g}_{k}^{(r-1)} \tag{71}$$

$$\Delta\tilde{h}_{k}^{(r)} = h_k - N\sum_{j=1}^{M}\sum_{t=1}^{T}\vartheta_{jtk}^{(r)}\rho_{jrt}\ln(X_{jrl}) - \tilde{h}_{k}^{(r-1)} \tag{72}$$

$$\Delta\tilde{g}_{k}^{\prime(r)} = g'_k + N\sum_{j=1}^{M}\sum_{t=1}^{T}\vartheta_{jtk}^{(r)}\rho_{jrt}\bar{\beta}_{k}^{(r-1)}\Bigg[\Psi\left(\bar{\alpha}_{k}^{(r-1)} + \bar{\beta}_{k}^{(r-1)}\right) - \Psi\left(\bar{\beta}_{k}^{(r-1)}\right) + \bar{\alpha}_{k}^{(r-1)}\Psi'\left(\bar{\alpha}_{k}^{(r-1)} + \bar{\beta}_{k}^{(r-1)}\right)\left(\langle\ln\alpha_{k}^{(r-1)}\rangle - \ln\bar{\alpha}_{k}^{(r-1)}\right)\Bigg] - \tilde{g}_{k}^{\prime(r-1)} \tag{73}$$

$$\Delta\tilde{h}_{k}^{\prime(r)} = h'_k - N\sum_{j=1}^{M}\sum_{t=1}^{T}\vartheta_{jtk}^{(r)}\rho_{jrt}\ln(1 - X_{jrl}) - \tilde{h}_{k}^{\prime(r-1)} \tag{74}$$

Since the hyperparameters of $q^{(r)}(\boldsymbol{\pi}')$, $q^{(r)}(\boldsymbol{\psi}')$, $q^{(r)}(\boldsymbol{\alpha})$ $q^{(r)}(\boldsymbol{\beta})$ and $q^{(r)}(\boldsymbol{\alpha}_l)$ are independent from each other, they can be updated in parallel. We repeat this online variational inference procedure until all the variational factors are updated with respect to the current arrived observation. It is worth mentioning that the online variational algorithm can be defined as a stochastic approximation method [27] for estimating the expected lower bound and the convergence is guaranteed if the learning rate satisfies the following conditions [48]:

$$\sum_{r=1}^{\infty}\xi_r = \infty\,, \qquad \sum_{r=1}^{\infty}\xi_r^2 < \infty \tag{75}$$

The online variational inference for hierarchical infinite Beta-Liouville mixture model is summarized in Algorithm 2. The computational complexity for the proposed online variational hierarchical infinite Beta-Liouville mixture model is $\mathcal{O}(MTKD)$ in each iteration. In contrast, its batch learning counterpart requires $\mathcal{O}(NMTKD)$ in each iteration, where N in this case represents the size of the data set that is observed. This is due to the fact that the batch algorithm updates the variational factors using the whole data set in each iteration, and thus its estimation quality is improved much more slowly than in the case of the online one. Please also notice that the total computational time depends on the number of iterations that is required to converge.

Algorithm 2 Online variational learning of hierarchical infinite Beta-Liouville mixture model

Choose the initial truncation levels K and T;
Initialize the values for hyperparameters $\lambda_{jt}, \gamma_k, u_{kl}, v_{kl}, g_k, h_k, g'_k, h'_k$;
while There is more data to observe **do**
  The variational E-step:
  Update the variational solution to $q(\mathbf{Z}_r)$ using Eq. (48);
  The variational M-step:
  Compute learning rate $\xi_r = (\eta_0 + r)^{-\varsigma}$.
  Calculate the natural gradient $\Delta\vartheta_{jtk}^{(r)}$ using Eq. (53);
  Update the variational factor $q^{(r)}(\mathbf{W})$ as shown in Eq. (51);
  Calculate the natural gradients of the remaining hyperparameters using Eqs. (65)–(74);
  Update variational factors $q^{(r)}(\boldsymbol{\pi}')$, $q^{(r)}(\boldsymbol{\psi}')$, $q^{(r)}(\boldsymbol{\alpha}_l)$, $q^{(r)}(\boldsymbol{\alpha})$ and $q^{(r)}(\boldsymbol{\beta})$ through Eqs. (55)–(59);
  Repeat the variational *E-step* and *M-step* until new data is observed;
  $r = r + 1$;
end while

3.3 Discussion

Regarding the learning of mixture models, other popular approximation schemes may also be applied. Indeed, two classes of approximation techniques can be broadly defined, depend on whether they rely on stochastic or deterministic approximations. Stochastic techniques, such as Markov chain Monte Carlo (MCMC) [46], are based on numerical sampling and can provide exact results in theory if given infinite computational resource. Nevertheless, in practice, the use of sampling method is limited to small-scale problems due to the high computational cost. Moreover, it is often difficult to analysis the convergence. By contrast, the deterministic approximation schemes such as expectation prorogation [39,40] and variational Bayes are based on analytical approximations to the posterior distribution. Expectation prorogation, which is based on the assumed-density filtering [37], is a recursive approximation scheme based on the minimization of a Kullback-Leibler divergence between the true model's posterior and an approximation. Compared with variational Bayes, expectation prorogation can provide comparable learning results if the data set is relatively small, whereas its performance would be degraded for large-scale data set [35]. In our work, we adopt variational Bayes as the approximation method for model learning. The variational approach is based on analytical approximations to the posterior distribution, which has received a lot of attention and has provided good generalization performance as well as computational tractability in various applications. Additionally, variational Bayes has also increased the power and flexibility of mixture models by allowing full inference about all the involved parameters and then simultaneous model selection and parameters estimation.

4 Experimental Results

We validate the proposed online hierarchical infinite Beta-Liouville mixture model (referred to as *OnHInBL*) through a real-world application namely scene recognition. Indeed, the availability of relatively cheap digital communication and the popularity of the WWW has made image databases largely available. A lot of research works have been devoted to the development of efficient image representation and classification approaches [11,29,34,43,44,49,50]. This task is challenging and has a lot of potential applications. Examples of applications include multimedia retrieval, annotation, summarizing and browsing [14,16,33,41,42]. An important step is the extraction of the images content descriptors [28,56]. According to [59], "scene" represents a place where a human can act within or navigate. The goal of scene recognition is to classify a set of natural images into a number of semantic categories. In our work, we perform the scene recognition using our *OnHInBL* model with a bag-of-visual words representation. In this experiment, our specific choice for initializing the hyperparameters is the following: $(\lambda_{jt}, \gamma_k, u_{kl}, v_{kl}, g_k, h_k, g'_k, h'_k) = (0.1, 0.1, 1, 0.1, 1, 0.1, 1, 0.1)$. Parameters ς and η_0 of the learning rate are set to 0.65 and 64, respectively. Furthermore, the global truncation level K is set to 750 and the group truncation level T is set to 80. It is worth mentioning that these specific choices were found convenient in our experiment.

4.1 Experimental Design

The procedure for performing scene recognition using the proposed *OnHInBL* model and bag-of-visual words representation is described as the following: First, we extract and normalize PCA-SIFT descriptors[1] (36-dimensional) [25] from original images using the Difference-of-Gaussian (DoG) detector [32]. Next, our *OnHInBL* is used to model these obtained PCA-SIFT feature vectors. More specifically, each image $\mathcal{I}_j$ is treated as a "group" in our hierarchical model and is associated with an infinite mixture model G_j. Thus, each PCA-SIFT feature vector X_{ji} of the image $\mathcal{I}_j$ is considered to be drawn from G_j and the mixture components of G_j are treated as "visual words". Then, a global vocabulary is constructed and is shared among all groups (images) through the common global infinite mixture model G_0 within our hierarchical model. It is noteworthy that this setting matches the desired design of a hierarchical Dirichlet process mixture model where observations are organized into groups and these groups are statistically linked by sharing mixture components. Indeed, an important step in bag-of-visual words representation is the construction of visual vocabulary. As we may notice, most of previously proposed approaches have to apply a separate vector quantization method (such as K-means) to build the visual vocabulary, where the size of the vocabulary is normally chosen manually. This problem can be tackled elegantly in our approach since the construction of the visual vocabulary is part of our hierarchical Dirichlet process mixture framework. As a result, the size of the vocabulary (i.e. the number of mixture components in the global-level mixture model) can be inferred automatically from the data thanks to the property of nonparametric Bayesian model. Then, the "bag-of-words" paradigm is employed and a histogram of "visual words" for each image is computed. Since the goal of our experiment is to determine which scene category that a testing image $\mathcal{I}_j$ belongs to, we need to introduce an indicator variable B_{jm} associated with each image (or group) in our hierarchical Dirichlet process mixture framework. B_{jm} denotes image $\mathcal{I}_j$ comes from category m and is drawn from another infinite mixture model which is truncated at level J. This means that we need to add a new level of hierarchy to our hierarchical infinite mixture model with a sharing vocabulary among all scene categories. In our experiment, we truncate J to 50 and initialize the hyperparameter of the mixing probability of B_{jm} to 0.1. Finally, we assign a testing image into the category which results the highest posterior probability according to Bayes' decision rule.

4.2 Data set and Results

We conducted our test on a challenging scene data set namely Scene UNderstanding (SUN) database which contains 899 scene categories and 130,519 images.[2] Images within each cat-

[1] Source code of PCA-SIFT: http://www.cs.cmu.edu/~yke/pcasift.

[2] Database is available at: http://vision.princeton.edu/projects/2010/SUN/.

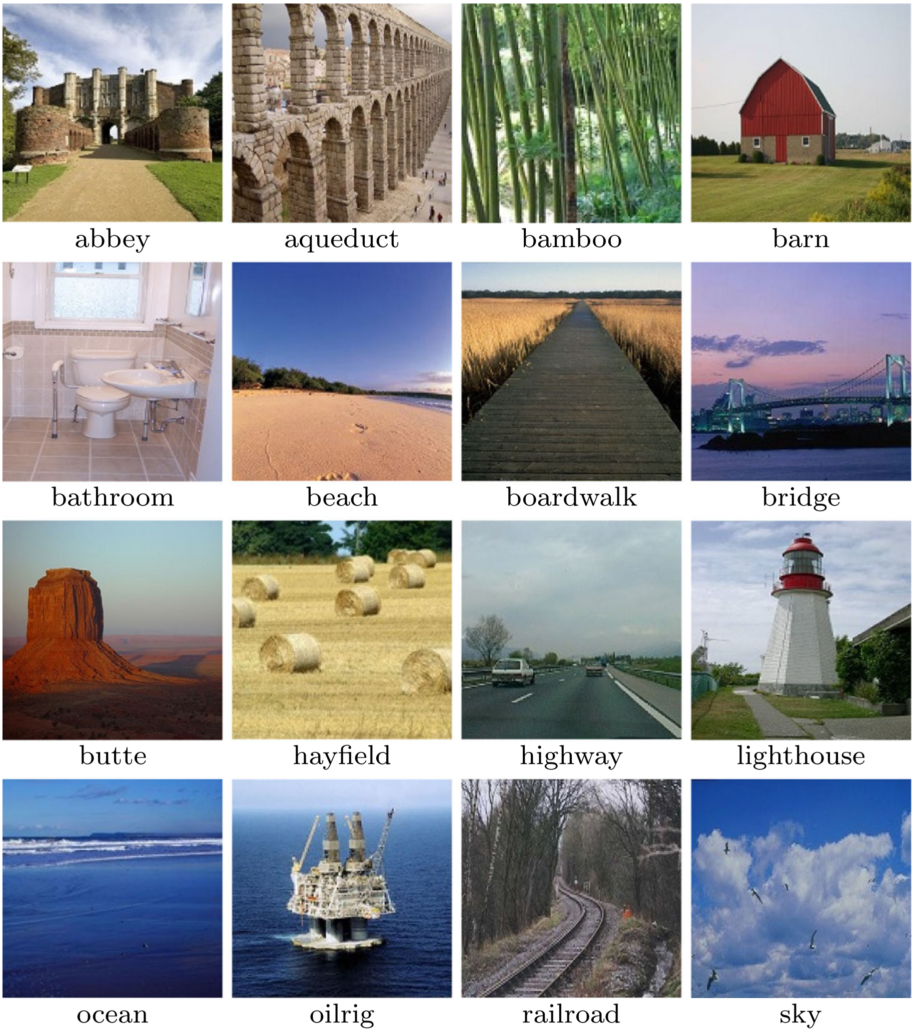

Fig. 1 Sample scene images from the SUN database

egory were obtained using WordNet terminology from various search engines on the internet [59]. In our case, we randomly chose 16 of the 899 categories (e.g. "abbey", "aqueduct", "lighthouse", and "beach") in the SUN database to evaluate the performance of our approach. Furthermore, we have to ensure that each selected category must have at least 100 images. Thus, each of these categories contains 100 scene images and therefore we have 1600 images in total. We randomly divided this data set into two halves: one for training (to learn the model and build the visual vocabulary), the other one for testing. Sample images from each scene tested category can be viewed in Fig. 1.

The proposed approach was evaluated by running it 30 times. We compared our *OnHInBL* approach with several other mixture-modeling approaches for scene recognition, in order to demonstrate its advantages. Our goal for the comparison can be summarized in three-fold: to compare the online learning algorithm with the batch one; to compare the hierarchical

Table 1 The average scene recognition rate (%) and the corresponding the runtime (s) using different methods for the SUN database. The numbers in parenthesis are the standard deviation of the corresponding quantities

Method	Recognition rate (%)	Runtime (s)
OnHInBL	70.81 (1.58)	87.65
BaHInBL	71.27 (1.39)	319.46
OnInBL	66.34 (1.64)	80.52
OnHInGau	63.15 (1.23)	83.13

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
abbey (a)	0.58	0.31	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.03	0.00	0.02	0.02	0.00
aqueduct (b)	0.25	0.61	0.00	0.00	0.03	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00
bamboo(c)	0.00	0.02	0.78	0.00	0.00	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00
barn (d)	0.00	0.00	0.00	0.73	0.00	0.00	0.00	0.03	0.13	0.00	0.00	0.06	0.00	0.05	0.00	0.00
bathroom (e)	0.02	0.01	0.00	0.00	0.83	0.00	0.00	0.07	0.00	0.00	0.00	0.04	0.00	0.03	0.00	0.00
beach (f)	0.00	0.00	0.00	0.06	0.00	0.71	0.05	0.00	0.00	0.08	0.02	0.00	0.08	0.00	0.00	0.00
boardwalk (g)	0.02	0.00	0.11	0.00	0.00	0.03	0.66	0.10	0.00	0.00	0.00	0.01	0.00	0.00	0.07	0.00
bridge (h)	0.04	0.05	0.00	0.02	0.03	0.00	0.09	0.73	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
butte (i)	0.00	0.00	0.00	0.15	0.00	0.03	0.00	0.00	0.64	0.10	0.00	0.06	0.00	0.02	0.00	0.00
hayfield (j)	0.05	0.00	0.00	0.08	0.00	0.02	0.00	0.00	0.00	0.82	0.00	0.00	0.00	0.03	0.00	0.00
highway (k)	0.03	0.00	0.00	0.00	0.00	0.04	0.13	0.00	0.00	0.00	0.72	0.00	0.02	0.00	0.06	0.00
lighthouse (l)	0.00	0.00	0.00	0.05	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.73	0.00	0.15	0.00	0.00
ocean (m)	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.02	0.00	0.71	0.00	0.00	0.21
oilrig (n)	0.04	0.00	0.00	0.00	0.05	0.00	0.00	0.02	0.03	0.00	0.00	0.12	0.00	0.74	0.00	0.00
railroad (o)	0.00	0.00	0.08	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.16	0.00	0.00	0.00	0.65	0.00
sky (p)	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.04	0.00	0.00	0.06	0.00	0.12	0.00	0.00	0.69

Fig. 2 Confusion matrix obtained by *OnHInBL* for the SUN database

Dirichlet process framework with the conventional Dirichlet process one; to compare Beta-Liouville mixture models with Gaussian ones on modeling proportional data. Thus, the proposed *OnHInBL* model was compared with: the batch hierarchical infinite Beta-Liouville mixture model (*BaHInBL*), the online infinite Beta-Liouville mixture (*OnInBL*) model and the online hierarchical infinite Gaussian mixture model (*OnHInGau*). All of these models were learned using variational inference. For the experiment of using *OnInBL* model, its visual vocabulary was built using the K-means algorithm and the size of its visual vocabulary was manually set to 600. The testing data in our experiments is supposed to arrive sequentially in an online manner except for the approach using *BaHInBL* model. Table 1 shows the average results of our *OnHInBL* model and the three other tested model for scene recognition on the SUN database. As illustrated in this table, both the proposed online approach (*OnHInGDFs*) and its batch counterpart (*BaHInBL*) can obtain the highest recognition rates among all tested approaches. Figures 2 and 3 present the confusion matrices obtained by *OnHInBL* and *BaHInBL*, respectively for the tested database. Each entry (i, j) of the confusion matrix denotes the number of images in category i that are assigned into category j. As shown in these two figures, we observed that the overall average recognition accuracy was 70.81 % (error rate of 29.19 %) using *OnHInBL*, and 71.27 % (error rate of 28.73 %) using *BaHInBL*. Although *BaHInBL* provided slightly higher recognition rate (71.27 %) than the one obtained

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
abbey (a)	0.61	0.20	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.04	0.00	0.06	0.01	0.00
aqueduct (b)	0.24	0.67	0.00	0.00	0.04	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00
bamboo (c)	0.00	0.12	0.72	0.00	0.00	0.03	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.07	0.00
barn (d)	0.00	0.00	0.00	0.78	0.00	0.00	0.00	0.09	0.09	0.00	0.00	0.03	0.00	0.01	0.00	0.00
bathroom (e)	0.05	0.09	0.00	0.00	0.75	0.00	0.00	0.06	0.00	0.00	0.00	0.04	0.00	0.01	0.00	0.00
beach (f)	0.00	0.00	0.00	0.06	0.01	0.69	0.15	0.00	0.00	0.01	0.04	0.00	0.04	0.00	0.00	0.00
boardwalk (g)	0.10	0.00	0.05	0.00	0.00	0.10	0.70	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bridge (h)	0.04	0.05	0.00	0.03	0.03	0.00	0.07	0.77	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
butte (i)	0.00	0.00	0.00	0.11	0.00	0.05	0.00	0.01	0.76	0.02	0.00	0.01	0.00	0.04	0.00	0.00
hayfield (j)	0.06	0.00	0.00	0.02	0.00	0.03	0.00	0.00	0.01	0.83	0.00	0.00	0.00	0.05	0.00	0.00
highway (k)	0.03	0.00	0.00	0.00	0.00	0.07	0.05	0.00	0.00	0.01	0.72	0.00	0.05	0.00	0.07	0.00
lighthouse (l)	0.00	0.00	0.00	0.08	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.75	0.00	0.13	0.00	0.00
ocean (m)	0.00	0.00	0.00	0.00	0.00	0.06	0.01	0.00	0.02	0.00	0.08	0.00	0.74	0.00	0.00	0.09
oilrig (n)	0.06	0.00	0.00	0.00	0.03	0.00	0.03	0.05	0.04	0.04	0.00	0.05	0.00	0.69	0.00	0.01
railroad (o)	0.00	0.00	0.10	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.13	0.04	0.00	0.01	0.62	0.01
sky (p)	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.16	0.00	0.00	0.07	0.00	0.09	0.00	0.01	0.60

Fig. 3 Confusion matrix obtained by *BaHInBL* for the SUN database

by *OnHInBL* (70.81 %), their difference is not statistically significant according to Student's t-test (i.e., we have calculated p-values between 0.1351 and 0.2512 for different runs). Therefore, *OnHInBL* is more favorable since it was significantly faster than *BaHInBL* as shown in Table 1, because of its online learning property. This can be explained by the fact that the batch algorithm updates the variational factors by using the whole data set in each iteration, and thus its estimation quality is improved more slowly than in the case of the online one. Moreover, *OnHInBL* outperformed *OnInBL* (70.81 vs. 66.34 %) as shown Table 1, which demonstrates the merits of using hierarchical Dirichlet process framework over the conventional Dirichlet process one. According to Table 1, better performance obtained by *OnHInBL* than the one acquired by *OnHInGau* verifies that the Beta-Liouville mixture model has a better modeling capability for proportional data than Gaussian mixture model.

5 Conclusion

In this paper, we have proposed a statistical clustering framework based on Beta-Liouville distribution. This statistical framework is developed from a nonparametric Bayesian perspective via a hierarchical Dirichlet process. For the learning of the model's parameters both variational approaches are proposed. The first one works on batch settings and the second one is incremental. The experimental results that have concerned image databases categorization have shown that our method is promising and has interesting advantages. It is noteworthy that in the proposed model, all data features are used for model learning. However, in practice, not all the features are important and some may be irrelevant. These irrelevant features may not contribute to the learning process or even degrade clustering results. Thus, one possible future work can be devoted to the inclusion of a feature selection scheme within the proposed framework in order to chooses the "best" feature subset for improving clustering performance.

Acknowledgments The completion of this work was supported by the Scientific Research Funds of Huaqiao University (600005-Z15Y0016). The authors would like to thank the anonymous referees and the associate editor for their comments.

References

1. Andrews JL, McNicholas PD, Subedi S (2011) Model-based classification via mixtures of multivariate t-distributions. Comput Stat Data Anal 55(1):520–529
2. Attias H (1999) A variational Bayes framework for graphical models. In: Proceedings of advances in neural information processing systems (NIPS), pp 209–215
3. Banerjee A, Langford J (2004) An objective evaluation criterion for clustering. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD). ACM, pp 515–520
4. Bishop CM (2006) Pattern recognition and machine learning. Springer, Heidelberg
5. Blei DM, Jordan MI (2005) Variational inference for Dirichlet process mixtures. Bayesian Anal 1:121–144
6. Bouguila N (2011) Bayesian hybrid generative discriminative learning based on finite liouville mixture models. Pattern Recog 44(6):1183–1200
7. Bouguila N (2012a) Hybrid generative/discriminative approaches for proportional data modeling and classification. IEEE Trans Knowl Data Eng 24(12):2184–2202
8. Bouguila N (2012b) Infinite liouville mixture models with application to text and texture categorization. Pattern Recog Lett 33(2):103–110
9. Bouguila N, Ziou D (2005) Using unsupervised learning of a finite Dirichlet mixture model to improve pattern recognition applications. Pattern Recog Lett 26(12):1916–1925
10. Bouguila N, Ziou D, Vaillancourt J (2004) Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application. IEEE Trans Image Process 13(11):1533–1543
11. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recog 37(9):1757–1771
12. Cheng H, Jiang X, Sun Y, Wang J (2001) Color image segmentation: advances and prospects. Pattern Recog 34(12):2259–2281
13. Cohn DA, Ghahramani Z, Jordan MI (1996) Active learning with statistical models. J Artif Intell Res 4:129–145
14. Corridoni JM, Bimbo AD, Pala P (1999) Image retrieval by color semantics. Multimed Syst 7(3):175–183
15. Erdem C, Karabulut G, Yanmaz E, Anarim E (2001) Motion estimation in the frequency domain using fuzzy c-planes clustering. IEEE Trans Image Process 10(12):1873–1879
16. Fan J, Gao Y, Luo H, Keim DA, Li Z (2008) A novel approach to enable semantic and visual image summarization for exploratory image search. In: Proceedings of the 1st ACM international conference on multimedia information retrieval (MIR). ACM, pp 358–365
17. Fan W, Bouguila N (2013a) 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)
18. Fan W, Bouguila N (2013b) Variational learning of finite Beta-liouville mixture models using component splitting. In: Proceedings of the international joint conference on neural networks (IJCNN), pp 1–8
19. Fan W, Bouguila N (2014) Non-gaussian data clustering via expectation propagation learning of finite Dirichlet mixture models and applications. Neural Process Lett 39(2):115–135
20. Fan W, Bouguila N, Ziou D (2012) Variational learning for finite Dirichlet mixture models and applications. IEEE Trans Neural Netw Learn Syst 23(5):762–774
21. Ferguson TS (1983) Bayesian density estimation by mixtures of normal distributions. Recent Adv Stat 24:287–302
22. Graepel T, Herbrich R (2008) Large scale data analysis and modelling in online services and advertising. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD), ACM, pp 2–2
23. Hegerath A, Deselaers T, Ney H (2006) Patch-based object recognition using discriminatively trained gaussian mixtures. In: Proceedings of the British machine vision conference (BMVC), pp 519–528
24. Ishwaran H, James LF (2001) Gibbs sampling methods for stick-breaking priors. J Am Stat Assoc 96:161–173
25. Ke Y, Sukthankar R (2004) Pca-sift: a more distinctive representation for local image descriptors. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR), pp II–506–II–513 Vol 2

26. Korwar RM, Hollander M (1973) Contributions to the theory of Dirichlet processes. Ann Probab 1:705–711
27. Kushner H, Yin G (1997) Stochastic approximation algorithms and applications, applications of mathematics. Springer, Berlin
28. Laaksonen J, Koskela M, Oja E (2002) Picsom-self-organizing image retrieval with mpeg-7 content descriptors. IEEE Trans Neural Netw 13(4):841–853
29. Lampert C, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 951–958
30. Liu X, Fu H, Jia Y (2008) Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images. Pattern Recog 41(2):484–493
31. Liu Z, Song YQ, Chen JM, Xie CH, Zhu F (2012) Color image segmentation using nonparametric mixture models with multivariate orthogonal polynomials. Neural Comput Appl 21(4):801–811
32. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
33. Lu G (2002) Techniques and data structures for efficient multimedia retrieval based on similarity. IEEE Trans Multimed 4(3):372–384
34. Luo J, Boutell M, Gray R, Brown C (2005) Image transform bootstrapping and its applications to semantic scene classification. IEEE Trans Syst Man Cybern Part B: Cybern 35(3):563–570
35. Ma Z, Leijon A (2010) Expectation propagation for estimating the parameters of the Beta distribution. In: Proceedings IEEE international conference on acoustics speech and signal processing (ICASSP), pp 2082–2085
36. Mancas-Thillou C, Gosselin B (2007) Color text extraction with selective metric-based clustering. Comput Vis Image Underst 107(1–2):97–107
37. Maybeck PS (1982) Stochastic models, estimation and control. Academic Press, New York
38. McNicholas PD (2010) Model-based classification using latent gaussian mixture models. Stat Plan Inference 140(5):1175–1181
39. Minka T (2001) Expectation propagation for approximate Bayesian inference. In: Proceedings of the conference on uncertainty in artificial intelligence (UAI), pp 362–369
40. Minka T, Lafferty J (2002) Expectation propagation for the generative aspect model. In: Proceedings of the conference on uncertainty in artificial intelligence (UAI), pp 352–359
41. Mojsilovic A, Rogowitz B (2004) Semantic metric for image library exploration. IEEE Trans Multimed 6(6):828–838
42. Mojsilovic A, Kovacevic J, Hu J, Safranek R, Ganapathy S (2000) Matching and retrieval based on the vocabulary and grammar of color patterns. IEEE Trans Image Process 9(1):38–54
43. Nilsback ME, Zisserman A (2006) A visual vocabulary for flower classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), vol 2, pp 1447–1454
44. Ozuysal M, Fua P, Lepetit V (2007) Fast keypoint recognition in ten lines of code. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
45. Park SH, Yun ID, Lee SU (1998) Color image segmentation based on 3-d clustering: morphological approach. Pattern Recog 31(8):1061–1076
46. Robert C, Casella G (1999) Monte Carlo statistical methods. Springer, New York
47. Santago P, Gage H (1995) Statistical models of partial volume effect. IEEE Trans Image Process 4(11):1531–1540
48. Sato M (2001) Online model selection based on the variational Bayes. Neural Comput 13:1649–1681
49. Schweitzer H (1999) Organizing image databases as visual-content search trees. Image Vis Comput 17(7):501–511
50. Seemann E, Leibe B, Mikolajczyk K, Schiele B (2005) An evaluation of local shape-based features for pedestrian detection. In: Proceedings of the British machine vision conference (BMVC)
51. Sethuraman J (1994) A constructive definition of Dirichlet priors. Statistica Sinica 4:639–650
52. Souden M, Kinoshita K, Nakatani T (2013) An integration of source location cues for speech clustering in distributed microphone arrays. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 111–115
53. Souden M, Kinoshita K, Delcroix M, Nakatani T (2014) Location feature integration for clustering-based speech separation in distributed microphone arrays. IEEE/ACM Trans Audio Speech Lang Process 22(2):354–367
54. Teh YW, Jordan MI (2010) Hierarchical Bayesian nonparametric models. In: Hjort N, Holmes C, Müller P, Walker S (eds) Bayesian nonparametrics: principles and practice. Cambridge University Press, Cambridge
55. Teh YW, Jordan MI, Beal MJ, Blei DM (2006) Hierarchical Dirichlet processes. J Am Stat Assoc 101(476):1566–1581

56. Thureson J, Carlsson S (2004) Appearance based qualitative image description for object class recognition. In: Pajdla T, Matas J (eds) ECCV (2), Springer, Lecture notes in computer science, vol 3022, pp 518–529
57. Wang C, Paisley JW, Blei DM (2011) Online variational inference for the hierarchical Dirichlet process. J Mach Learn Res—Proc Track 15:752–760
58. Wu Y, Huang TS (2000) Self-supervised learning for visual tracking and recognition of human hand. In: Proceedings of the 7th national conference on artificial intelligence and twelfth conference on on innovative applications of artificial intelligence (AAAI/IAAI), pp 243–248
59. Xiao J, Hays J, Ehinger K, Oliva A, Torralba A (2010) Sun database: large-scale scene recognition from abbey to zoo. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3485–3492