

# Learning Finite Beta-Liouville Mixture Models via Variational Bayes for Proportional Data Clustering

Wentao Fan

Electrical and Computer Engineering  
Concordia University, Canada  
wenta\_fa@encs.concordia.ca

Nizar Bouguila

Institute for Information Systems Engineering  
Concordia University, Canada  
nizar.bouguila@concordia.ca

## Abstract

During the past decade, finite mixture modeling has become a well-established technique in data analysis and clustering. This paper focus on developing a variational inference framework to learn finite Beta-Liouville mixture models that have been proposed recently as an efficient way for proportional data clustering. In contrast to the conventional expectation maximization (EM) algorithm, commonly used for learning finite mixture models, the proposed algorithm has the advantages that it is more efficient from a computational point of view and by preventing over- and under-fitting problems. Moreover, the complexity of the mixture model (i.e. the number of components) can be determined automatically and simultaneously with the parameters estimation in a closed form as part of the Bayesian inference procedure. The merits of the proposed approach are shown using both artificial data sets and two interesting and challenging real applications namely dynamic textures clustering and facial expression recognition.

## 1 Introduction

In recent years, there has been a tremendous increasing in the generation of digital data (e.g. text, images, or videos). Consequently, the need for effective tools to cluster and analyze complex data becomes more and more significant. Among many existing clustering methods, finite mixture models have gained increasing interest and have been successfully applied in various fields such as data mining, machine learning, image processing and bioinformatics [McLachlan and Peel, 2000]. Among all mixture models, the Gaussian mixture has been a popular choice in many research works due to its simplicity [Figueiredo and Jain, 2002; Constantinopoulos *et al.*, 2006]. However, the Gaussian mixture is not the best choice in all applications, and it fails to discover true underlying data structure when the partitions are clearly non-Gaussian. Recent works have shown that, in many real-world applications, other models such as the Dirichlet, generalized Dirichlet or Beta-Liouville mixtures can be better alternatives to clustering in particular and data modeling in general, and especially for modeling proportional

data (e.g. normalized histograms), such as data originating from text, images, or videos [Bouguila and Ziou, 2006; 2007; Bouguila, 2012]. It is noteworthy that the Beta-Liouville distribution contains the Dirichlet distribution as a special case and has a smaller number of parameters than the generalized Dirichlet. Furthermore, Beta-Liouville mixture models have shown better performance than both the Dirichlet and the generalized Dirichlet mixtures in [Bouguila, 2012].

In [Bouguila, 2012], a maximum likelihood (ML) estimation approach based on the EM algorithm is proposed to learn finite Beta-Liouville mixture models. However, the ML method has several disadvantages such as its suboptimal generalization performance and over-fitting. These drawbacks of ML estimation can be addressed by adopting a variational inference framework. Recently, variational inference (also known as variational Bayes) [Attias, 1999b; Jordan *et al.*, 1998; Bishop, 2006] has received considerable attention and has provided good generalization performance in many applications including finite mixtures learning [Watanabe *et al.*, 2009; Corduneanu and Bishop, 2001; Constantinopoulos *et al.*, 2006]. Unlike the ML method in which the number of mixture components is detected by applying some typical criteria such as Akaike information criterion (AIC), Bayes information criterion (BIC), minimum description length (MDL) or minimum message length (MML), variational method can estimate model parameters and determine the optimal number of components simultaneously.

The goal of this paper is to develop a variational inference framework for learning finite Beta-Liouville mixture models. We build a tractable lower bound for marginal likelihood by using approximate distributions to replace the real unknown parameter distributions. In contrast to traditional approaches in which the model selection is solved based on cross-validation, our method allows the estimation of model parameters and the detection of the number of components simultaneously in a single training run. The effectiveness of the proposed method is tested on extensive simulations using artificial data along with two challenging real-world applications. The rest of this paper is organized as follows. Section 2 presents the finite Beta-Liouville mixture model. The variational inference framework for model learning is described in Section 3. Section 4 is devoted to the experimental results. Finally, conclusion follows in Section 5.

## 2 Finite Beta-Liouville Mixture Model

Suppose that we have observed a set of  $N$  vectors  $\mathcal{X} = \{\vec{X}_1, \dots, \vec{X}_N\}$ , where each vector  $\vec{X}_i = (X_{i1}, \dots, X_{iD})$  is represented in a  $D$ -dimensional space and is drawn from a finite Beta-Liouville mixture model with  $M$  components as [Bouguila, 2012]

$$p(\vec{X}_i | \vec{\pi}, \vec{\theta}) = \sum_{j=1}^M \pi_j \text{BL}(\vec{X}_i | \theta_j) \quad (1)$$

where  $\theta_j = (\alpha_{j1}, \dots, \alpha_{jD}, \alpha_j, \beta_j)$  are the parameters of the Beta-Liouville,  $\text{BL}(\vec{X}_i | \theta_j)$ , representing component  $j$ . In addition,  $\vec{\theta} = (\theta_1, \dots, \theta_M)$ , and  $\vec{\pi} = (\pi_1, \dots, \pi_M)$  denotes the vector of mixing coefficients that are subject to the constraints  $0 \leq \pi_j \leq 1$  and  $\sum_{j=1}^M \pi_j = 1$ . The Beta-Liouville distribution  $\text{BL}(\vec{X}_i | \theta_j)$  is given by

$$\text{BL}(\vec{X}_i | \theta_j) = \frac{\Gamma(\sum_{d=1}^D \alpha_{jd}) \Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j) \Gamma(\beta_j)} \prod_{d=1}^D \frac{X_{id}^{\alpha_{jd}-1}}{\Gamma(\alpha_{jd})} \times \left( \sum_{d=1}^D X_{id} \right)^{\alpha_j - \sum_{d=1}^D \alpha_{jd}} \left( 1 - \sum_{d=1}^D X_{id} \right)^{\beta_j - 1} \quad (2)$$

More interesting properties of the Beta-Liouville distribution can be found in [Bouguila, 2012]. Next, we assign a binary latent variable  $\vec{Z}_i = (Z_{i1}, \dots, Z_{iM})$  to each observation  $\vec{X}_i$ , such that  $Z_{ij} \in \{0, 1\}$ ,  $\sum_{j=1}^M Z_{ij} = 1$ ,  $Z_{ij} = 1$  if  $\vec{X}_i$  belongs to component  $j$  and equal to 0, otherwise. The prior distribution of  $\mathcal{Z} = (\vec{Z}_1, \dots, \vec{Z}_N)$  is defined by

$$p(\mathcal{Z} | \vec{\pi}) = \prod_{i=1}^N \prod_{j=1}^M \pi_j^{Z_{ij}} \quad (3)$$

Then, we need to place conjugate priors over parameters  $\alpha_d$ ,  $\alpha$  and  $\beta$ . In our case, since  $\alpha_d$ ,  $\alpha$  and  $\beta$  are positive, Gamma distribution  $\mathcal{G}(\cdot)$  is adopted to approximate conjugate priors for these parameters. Thus, we can obtain the following prior distributions for  $\alpha_d$ ,  $\alpha$  and  $\beta$ , respectively:  $p(\alpha_d) = \mathcal{G}(\alpha_d | u_d, v_d)$ ,  $p(\alpha) = \mathcal{G}(\alpha | g, h)$ ,  $p(\beta) = \mathcal{G}(\beta | s, t)$ . The mixing coefficients  $\vec{\pi}$  are treated as parameters rather than random variables within our framework, hence, no prior is placed over  $\vec{\pi}$ . A directed graphical representation of this model is illustrated in Figure 1.

## 3 Variational Model Learning

In this section, we develop a variational inference framework<sup>1</sup> for learning the finite Beta-Liouville mixture model by following the inference methodology proposed in [Corduneanu and Bishop, 2001]. To simplify the notation, we define  $\Lambda = \{\mathcal{Z}, \vec{\theta}\}$  as the set of random variables. The central

<sup>1</sup>Variational learning is actually an approximation of the true fully Bayesian learning. It is noteworthy, however, that in the case of the exponential family of distributions, that contains the Beta-Liouville, the advantage of the Bayesian learning still remains in the variational Bayes learning as shown for instance in [Watanabe and Watanabe, 2007]

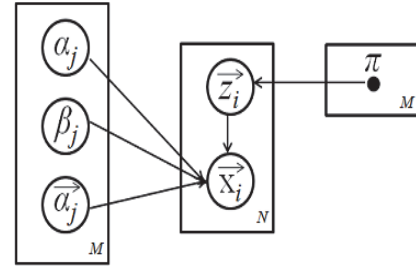


Figure 1: Graphical model representation of the finite Beta-Liouville mixture. Symbols in circles denote random variables; otherwise, they denote model parameters. Plates indicate repetition (with the number of repetitions in the lower right), and arcs describe conditional dependencies between variables.

idea in variational learning is to find a proper approximation for the posterior distribution  $p(\Lambda | \mathcal{X}, \vec{\pi})$ . This is achieved by maximizing the lower bound  $\mathcal{L}(Q)$  of the logarithm of the marginal likelihood  $p(\mathcal{X} | \vec{\pi})$ . By applying Jensen's inequality,  $\mathcal{L}(Q)$  can be found as

$$\mathcal{L}(Q) = \int Q(\Lambda) \ln[p(\mathcal{X}, \Lambda | \vec{\pi}) / Q(\Lambda)] d\Lambda \quad (4)$$

where  $Q(\Lambda)$  is an approximation for  $p(\Lambda | \mathcal{X}, \vec{\pi})$ . In our work, we use a factorial approximation as adopted in many other variational learning works [Constantinopoulos *et al.*, 2006; Corduneanu and Bishop, 2001; Bishop, 2006], to factorize  $Q(\Lambda)$  into disjoint tractable factors as  $Q(\Lambda) = Q(\mathcal{Z})Q(\vec{\alpha}_d)Q(\vec{\alpha})Q(\vec{\beta})$ . It is worth mentioning that this assumption is imposed purely to achieve tractability, and no restriction is placed on the functional forms of the individual factors. In order to maximize the lower bound  $\mathcal{L}(Q)$ , we need to make a variational optimization of  $\mathcal{L}(Q)$  with respect to each of these factors in turn. For a specific factor  $Q_s(\Lambda_s)$ , the general expression for its optimal solution is given by:

$$Q_s(\Lambda_s) = \frac{\exp\langle \ln p(\mathcal{X}, \Lambda) \rangle_{\neq s}}{\int \exp\langle \ln p(\mathcal{X}, \Lambda) \rangle_{\neq s} d\Lambda} \quad (5)$$

where  $\langle \cdot \rangle_{\neq s}$  denotes an expectation with respect to all the factor distributions except for  $s$ . By applying (5) to each variational factor, we obtain the optimal solutions for the factors of the variational posterior as

$$Q(\mathcal{Z}) = \prod_{i=1}^N \prod_{j=1}^M r_{ij}^{Z_{ij}}, \quad Q(\vec{\alpha}_d) = \prod_{j=1}^M \prod_{d=1}^D \mathcal{G}(\alpha_{jd} | u_{jd}^*, v_{jd}^*) \quad (6)$$

$$Q(\vec{\alpha}) = \prod_{j=1}^M \mathcal{G}(\alpha_j | g_j^*, h_j^*), \quad Q(\vec{\beta}) = \prod_{j=1}^M \mathcal{G}(\beta_j | s_j^*, t_j^*) \quad (7)$$

where

$$r_{ij} = \frac{\tilde{r}_{ij}}{\sum_{k=1}^M \tilde{r}_{ik}} \quad (8)$$

$$\tilde{r}_{ij} = \exp \left[ \ln \pi_j + \tilde{\mathcal{S}}_j + \tilde{\mathcal{H}}_j + (\bar{\alpha}_j - \sum_{d=1}^D \bar{\alpha}_{jd}) \ln \left( \sum_{d=1}^D X_{id} \right) \right]$$

$$+ (\bar{\beta}_j - 1) \ln(1 - \sum_{d=1}^D X_{id}) + \sum_{d=1}^D (\bar{\alpha}_{jd} - 1) \ln X_{id} \quad (9)$$

$$u_{jd}^* = u_{jd} + \sum_{i=1}^N \langle Z_{ij} \rangle \bar{\alpha}_{jd} \left[ \Psi\left(\sum_{d=1}^D \bar{\alpha}_{jd}\right) - \Psi(\bar{\alpha}_{jd}) \right. \\ \left. + \Psi'\left(\sum_{d=1}^D \bar{\alpha}_{jd}\right) \sum_{l \neq d}^D (\ln \alpha_{jl}) - \ln \bar{\alpha}_{jl} \right] \bar{\alpha}_{jd} \quad (10)$$

$$v_{jd}^* = v_{jd} - \sum_{i=1}^N \langle Z_{ij} \rangle \left[ \ln X_{id} - \ln\left(\sum_{d=1}^D X_{id}\right) \right] \quad (11)$$

$$g_j^* = g_j + \sum_{i=1}^N \langle Z_{ij} \rangle \left[ \Psi(\bar{\alpha}_j + \bar{\beta}_j) - \Psi(\bar{\alpha}_j) \right. \\ \left. + \bar{\beta}_j \Psi'(\bar{\alpha}_j + \bar{\beta}_j) (\langle \ln \beta_j \rangle - \ln \bar{\beta}_j) \right] \bar{\alpha}_j \quad (12)$$

$$h_j^* = h_j - \sum_{i=1}^N \langle Z_{ij} \rangle \ln\left(\sum_{d=1}^D X_{id}\right) \quad (13)$$

$$t_j^* = t_j - \sum_{i=1}^N \langle Z_{ij} \rangle \ln(1 - \sum_{d=1}^D X_{id}) \quad (14)$$

$$s_j^* = s_j + \sum_{i=1}^N \langle Z_{ij} \rangle \left[ \Psi(\bar{\alpha}_j + \bar{\beta}_j) - \Psi(\bar{\beta}_j) \right. \\ \left. + \bar{\alpha}_j \Psi'(\bar{\alpha}_j + \bar{\beta}_j) (\langle \ln \alpha_j \rangle - \ln \bar{\alpha}_j) \right] \bar{\beta}_j \quad (15)$$

where  $\Psi(\cdot)$  in the above equations is the digamma function.  $\tilde{\mathcal{S}}_j$  and  $\tilde{\mathcal{H}}_j$  are the lower bounds of  $\mathcal{S}_j = \left\langle \ln \frac{\Gamma(\sum_{d=1}^D \alpha_{jd})}{\prod_{d=1}^D \Gamma(\alpha_{jd})} \right\rangle$  and  $\mathcal{H}_j = \left\langle \ln \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \right\rangle$ , respectively. Since these expectations are intractable, we use the second-order Taylor series expansion to calculate their lower bounds.

In this work, the suitable number of components in the mixture model is determined by evaluating the values of mixing coefficients. Since the mixing coefficients  $\tilde{\pi}$  are treated as parameters, we can make point estimates of their values by maximizing the lower bound  $\mathcal{L}(Q)$  with respect to  $\tilde{\pi}$ . By setting the derivative of this lower bound with respect to  $\tilde{\pi}$  to zero, we then have:

$$\pi_j = \frac{1}{N} \sum_{i=1}^N r_{ij} \quad (16)$$

Within this framework, components which provide insufficient contribution to explaining the data would have their mixing coefficients driven to zero during the variational optimization, and so they can be effectively eliminated from the model through *automatic relevance determination* [Mackay, 1995; Bishop, 2006]. Therefore, we can obtain the proper number of components in a single training run by starting with a relatively large initial value of  $M$  and then remove the redundant components after convergence. Another important property of variational inference is that we can trace the convergence systematically by monitoring the variational lower

bound during the re-estimation step [Attias, 1999b]. This is because at each step of the iterative re-estimation procedure, the value of this bound should never decrease.

The variational inference for finite Beta-Liouville mixture model can be approached via an EM-like framework with a guaranteed convergence [Attias, 1999a; Wang and Titterton, 2006] and is summarized as follows:

#### 1. Initialization

- Choose the initial number of components  $M$ .
- Initialize the values for hyper-parameters  $u_{jd}$ ,  $v_{jd}$ ,  $g_j$ ,  $h_j$ ,  $s_j$  and  $t_j$ .
- Initialize the values of  $r_{ij}$  by  $K$ -Means algorithm.

#### 2. The variational E-step: Update the variational solutions for each factor using (6) and (7).

#### 3. The variational M-step: Maximize the lower bound $\mathcal{L}(Q)$ with respect to the current value of $\tilde{\pi}$ using (16).

#### 4. Repeat steps 2 and 3 until a convergence criterion is reached.

#### 5. Detect the optimal value of $M$ by eliminating the components with small mixing coefficients close to 0.

## 4 Experimental Results

In this section, we evaluate the effectiveness of the proposed variational finite Beta-Liouville mixture model (denoted as *varBLM*) through both artificial data and two challenging real-world applications involving dynamic textures clustering and facial expression recognition. The goal of the artificial data is to investigate the accuracy of *varBLM* in terms of estimation (estimating the model's parameters) and selection (selecting the number of components of the mixture model). The real applications have two main goals. The first goal is to compare our approach to the MML-based approach (*MML-BLM*) previously proposed in [Bouguila, 2012]. The second goal is to demonstrate the advantages of *varBLM* by comparing its performance with three other finite mixture models including the finite generalized Dirichlet (*varGDM*), finite Dirichlet (*varDM*) and finite Gaussian (*varGM*) mixtures, all learned in a variational way. In all the experiments, the initial number of components  $M$  is set to 15 with equal mixing coefficients. Our specific choice for the initial values of hyperparameters is  $(u_{jd}, g_j, h_j, v_{jd}, s_j, t_j) = (1, 1, 1, 0.01, 0.01, 0.01)$  which was found convenient according to our experimental results.

### 4.1 Artificial Data

First, we have tested the proposed *varBLM* algorithm for estimating model parameters on four 2-dimensional artificial data sets. Here we choose  $D = 2$  just for ease of representation. Table 1 shows the actual and estimated parameters of the four generated data sets by using *varBLM*. According to the results shown in this table, it is clear that our approach is able to estimate accurately the parameters of the mixture models. Figure 2 represents the resultant mixtures with different shapes (symmetric and asymmetric modes) by using *varBLM*. Figure 3 demonstrates the estimated mixing coefficients of each mixture component for each data set obtained

Table 1: Parameters of the synthetic data.  $N$  denotes the total number of elements,  $N_j$  denotes the number of elements in cluster  $j$ .  $\alpha_{j1}, \alpha_{j2}, \alpha_j, \beta_j$  and  $\pi_j$  are the real parameters.  $\hat{\alpha}_{j1}, \hat{\alpha}_{j2}, \hat{\alpha}_j, \hat{\beta}_j$  and  $\hat{\pi}_j$  are the estimated parameters using *varBLM*.

|                | $N_j$ | $j$ | $\alpha_{j1}$ | $\alpha_{j2}$ | $\alpha_j$ | $\beta_j$ | $\pi_j$ | $\hat{\alpha}_{j1}$ | $\hat{\alpha}_{j2}$ | $\hat{\alpha}_j$ | $\hat{\beta}_j$ | $\hat{\pi}_j$ |
|----------------|-------|-----|---------------|---------------|------------|-----------|---------|---------------------|---------------------|------------------|-----------------|---------------|
| Data set 1     | 100   | 1   | 15            | 10            | 8          | 15        | 0.25    | 14.89               | 10.08               | 7.91             | 15.22           | 0.253         |
| ( $N = 400$ )  | 300   | 2   | 22            | 25            | 17         | 10        | 0.75    | 21.78               | 24.82               | 17.15            | 9.87            | 0.747         |
| Data set 2     | 150   | 1   | 15            | 10            | 8          | 15        | 0.25    | 15.15               | 10.11               | 8.26             | 15.29           | 0.245         |
| ( $N = 600$ )  | 150   | 2   | 22            | 25            | 17         | 10        | 0.25    | 22.23               | 25.32               | 16.77            | 10.25           | 0.252         |
|                | 300   | 3   | 35            | 12            | 30         | 18        | 0.50    | 35.41               | 12.14               | 30.36            | 17.72           | 0.503         |
| Data set 3     | 160   | 1   | 15            | 10            | 8          | 15        | 0.20    | 14.76               | 9.82                | 8.45             | 14.69           | 0.201         |
| ( $N = 800$ )  | 160   | 2   | 22            | 25            | 17         | 10        | 0.20    | 22.34               | 24.61               | 17.29            | 10.32           | 0.197         |
|                | 240   | 3   | 35            | 12            | 30         | 18        | 0.30    | 34.52               | 12.28               | 29.63            | 18.27           | 0.297         |
|                | 240   | 4   | 20            | 43            | 15         | 15        | 0.30    | 20.38               | 42.34               | 14.68            | 15.21           | 0.305         |
| Data set 4     | 200   | 1   | 15            | 10            | 8          | 15        | 0.20    | 15.49               | 10.30               | 7.65             | 15.34           | 0.199         |
| ( $N = 1000$ ) | 200   | 2   | 22            | 25            | 17         | 10        | 0.20    | 21.55               | 25.63               | 16.72            | 10.41           | 0.197         |
|                | 200   | 3   | 35            | 12            | 30         | 18        | 0.20    | 34.37               | 11.75               | 30.61            | 18.25           | 0.202         |
|                | 200   | 4   | 20            | 43            | 15         | 15        | 0.20    | 19.63               | 43.89               | 15.37            | 15.29           | 0.198         |
|                | 200   | 5   | 50            | 30            | 25         | 5         | 0.20    | 51.28               | 30.64               | 25.43            | 5.15            | 0.204         |

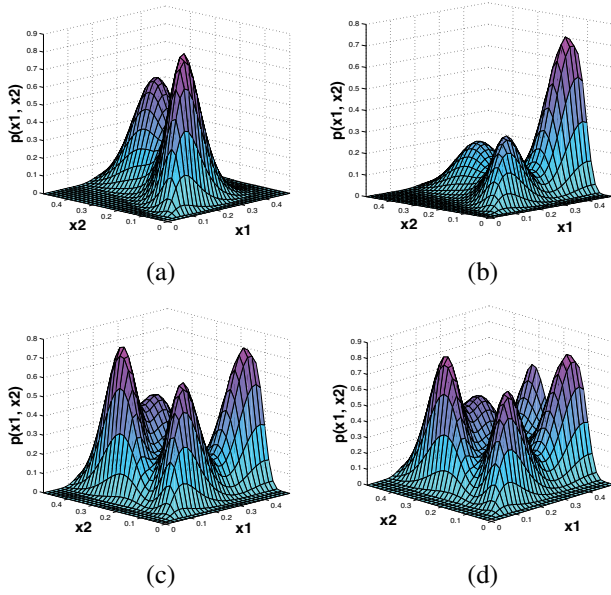


Figure 2: Estimated mixture densities for the artificial data sets. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.

by our algorithm. From this figure, we can see that redundant components have estimated mixing coefficients close to 0 after convergence. Thus, by eliminating these redundant components, we can obtain the correct number of components for each generated data set.

## 4.2 Real Applications

In this section, we attempt to develop effective approaches for addressing two challenging real applications namely dynamic textures clustering and facial expression recognition,

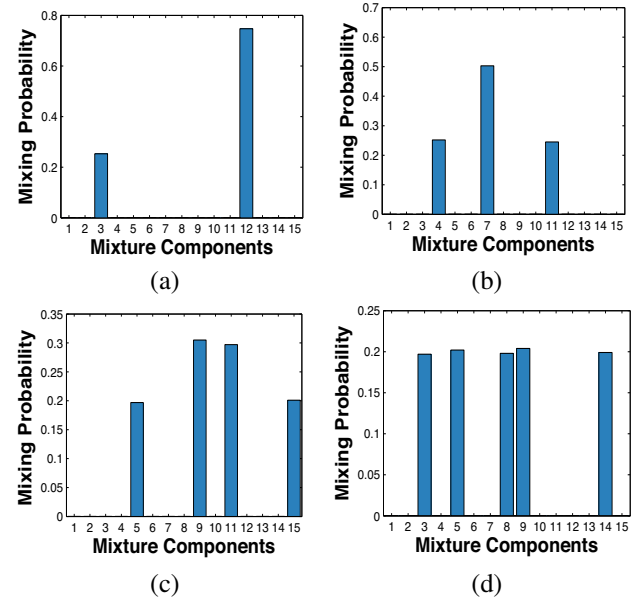


Figure 3: Estimated mixing coefficients for the artificial data sets. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.

based on the proposed *varBLM*. Generally, approaches to tackle these two tasks are based on two vital aspects. The first one concerns the extraction of visual features to obtain a good and effective representation of the dynamic texture or the facial expression. The second one concerns the design of an efficient clustering approach. The optimal visual features should be able to minimize within-class variations while maximizing between class variations. Even the best clustering method may provide poor performance if inadequate features are used. In our work, we adopt a powerful and effective

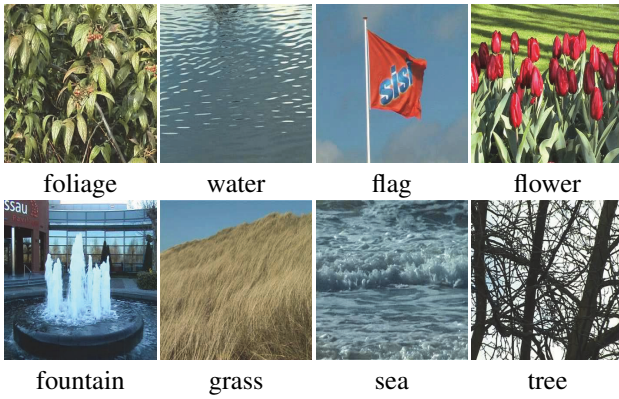


Figure 4: Sample frames from the DynTex database.

|          |         |       |      |        |          |       |      |      |
|----------|---------|-------|------|--------|----------|-------|------|------|
| foliage  | 0.96    | 0.00  | 0.00 | 0.02   | 0.00     | 0.00  | 0.00 | 0.02 |
| water    | 0.00    | 0.78  | 0.00 | 0.00   | 0.00     | 0.16  | 0.06 | 0.00 |
| flag     | 0.00    | 0.00  | 1.00 | 0.00   | 0.00     | 0.00  | 0.00 | 0.00 |
| flower   | 0.06    | 0.00  | 0.00 | 0.77   | 0.00     | 0.00  | 0.00 | 0.17 |
| fountain | 0.00    | 0.00  | 0.00 | 0.00   | 0.83     | 0.00  | 0.17 | 0.00 |
| grass    | 0.10    | 0.00  | 0.00 | 0.00   | 0.00     | 0.81  | 0.09 | 0.00 |
| sea      | 0.00    | 0.13  | 0.00 | 0.00   | 0.13     | 0.07  | 0.67 | 0.00 |
| tree     | 0.08    | 0.00  | 0.00 | 0.08   | 0.00     | 0.00  | 0.00 | 0.84 |
|          | foliage | water | flag | flower | fountain | grass | sea  | tree |

Figure 5: Confusion matrix obtained by *varBLM* for the DynTex database.

feature descriptor which is extracted by concatenating Local Binary Pattern histograms from three orthogonal planes (denoted as LBP-TOP features) [Zhao and Pietikainen, 2007]<sup>2</sup>. The motivation of choosing the LBP-TOP descriptor in our work stems from its superior performance to other comparable descriptors for representing both dynamic textures and facial images. In our experiments, we adopt a specific setting of the LBP-TOP descriptor as suggested in [Zhao and Pietikainen, 2007] with  $LBP - TOP_{4,4,4,1,1,1}$ . The subscript represents using the operator in a (4, 4, 4, 1, 1, 1) neighborhood. This specific choice of the LBP-TOP descriptor can achieve good performance while preserving a comparably shorter length of feature vector (48).

### Dynamic textures clustering

In recent years, dynamic textures have drawn considerable attention and have been used for various applications such as screen savers, personalized web pages and computer games [Kwatra *et al.*, 2003; Ravichandran *et al.*, 2013; Doretto *et*

<sup>2</sup>The source code of the LBP-TOP descriptor can be obtained at: <http://www.ee.oulu.fi/~gyzhao>

Table 2: Average categorization accuracy and the number of categories ( $\hat{M}$ ) computed by different algorithms over 30 runs.

| Methods       | $\hat{M}$       | Accuracy (%)     |
|---------------|-----------------|------------------|
| <i>varBLM</i> | $7.54 \pm 0.39$ | $83.25 \pm 1.35$ |
| <i>MMLBLM</i> | $7.39 \pm 0.45$ | $82.46 \pm 1.53$ |
| <i>varGDM</i> | $7.47 \pm 0.43$ | $81.14 \pm 1.49$ |
| <i>varDM</i>  | $7.43 \pm 0.41$ | $79.53 \pm 1.38$ |
| <i>varGM</i>  | $7.36 \pm 0.52$ | $77.39 \pm 1.64$ |

*et al.*, 2003]. Dynamic texture is defined as a video sequence of moving scenes that exhibit some stationarity characteristics in time (e.g., fire, sea-waves, swinging flag in the wind, etc.). It can be considered as an extension of the image texture to the temporal domain. Here, we highlight the application of dynamic textures clustering using the proposed *varBLM* algorithm and LBP-TOP features.

We conducted our test on a challenging dynamic textures data set which is known as the DynTex database [Péteri *et al.*, 2010]<sup>3</sup>. This data set contains around 650 dynamic texture video sequences from various categories. In our case, we use a subset of this data set which contains 8 categories of dynamic textures: foliage (29 sequences), water (30 sequences), flag (31 sequences), flower (29 sequences), fountain (37 sequences), grass (23 sequences), sea (38 sequences), and tree (25 sequences). Sample frames from each category are shown in Figure 4. The first step in our approach is to extract LBP-TOP features from the available dynamic texture sequences. After this step, each dynamic texture is represented as a 48-dimensional normalized histogram. The obtained normalized histograms are clustered using the proposed *varBLM*. We evaluated the performance of the proposed algorithm by running it 30 times.

The confusion matrix for the DynTex database provided by *varBLM* is shown in Figure 5. For comparison, we have also performed *MMLBLM*, *varGDM*, *varDM* and *varGM* algorithms with the same experimental methodology. The average results of the categorization accuracy and the estimated number of categories are illustrated in Table 2. According to this table, it is obvious that the best performance is obtained by using *varBLM* in terms of the highest categorization accuracy rate (83.25%) and most accurately selected number of components (7.54). This demonstrates the advantage of using variational inference over the EM algorithm. Moreover, we may notice that *varGM* provided the worst performance among all tested algorithms. This result proves that Gaussian mixture model is not suitable for dealing with proportional data.

### 4.3 Facial expression recognition

Facial expression recognition is an interesting and challenging problem which deals with the classification of human facial motions and impacts important applications in many areas such as image understanding, psychological studies, fa-

<sup>3</sup><http://projects.cwi.nl/dyntex/index.html>



Figure 6: Sample facial expression images from the Cohn-Kanade database.

| Anger    | 0.85  | 0.02    | 0.00 | 0.00 | 0.05    | 0.00     | 0.08    |
|----------|-------|---------|------|------|---------|----------|---------|
| Disgust  | 0.00  | 0.98    | 0.02 | 0.00 | 0.00    | 0.00     | 0.00    |
| Fear     | 0.00  | 0.04    | 0.64 | 0.19 | 0.01    | 0.00     | 0.12    |
| Joy      | 0.00  | 0.00    | 0.04 | 0.96 | 0.00    | 0.00     | 0.00    |
| Sadness  | 0.10  | 0.00    | 0.00 | 0.00 | 0.73    | 0.02     | 0.15    |
| Surprise | 0.00  | 0.00    | 0.03 | 0.00 | 0.00    | 0.97     | 0.00    |
| Neutral  | 0.02  | 0.00    | 0.00 | 0.01 | 0.04    | 0.00     | 0.93    |
|          | Anger | Disgust | Fear | Joy  | Sadness | Surprise | Neutral |

Figure 7: Confusion matrix obtained by *varBLM* for the Cohn-Kanade database.

cial nerve grading in medicine, synthetic face animation and virtual reality. It has attracted growing attention during the last decade [Shan *et al.*, 2009; Zhao and Pietikainen, 2007; Black and Yacoob, 1997; Pantic and Rothkrantz, 1999; Koelstra *et al.*, 2010]. In this experiment, we investigate the problem of facial expression recognition using our learning approach and LBP-TOP features.

We use a challenging facial expression database known as the Cohn-Kanade database [Kanade *et al.*, 2000; Lucey *et al.*, 2010] to evaluate the performance of our method. The Cohn-Kanade database is one of the most widely used databases by the facial expression research community<sup>4</sup>. It contains 486 image sequences from 97 subjects. In this experiment, we use 300 image sequences from the database. The selected sequences are those that could be labeled as one of the six basic emotions, i.e. Angry, Disgust, Fear, Joy, Sadness, and Surprise. Our sequences come from 94 subjects, with 16 basic emotions per subject. The neutral face and three peak frames of each sequence were used for facial expression recognition,

<sup>4</sup><http://www.pitt.edu/~jeffcohn/CKandCK+.htm>

Table 3: Average recognition accuracy and the number of components ( $\hat{M}$ ) computed by different algorithms for the Cohn-Kanade database.

| Methods       | $\hat{M}$       | Accuracy (%)     |
|---------------|-----------------|------------------|
| <i>varBLM</i> | $6.65 \pm 0.31$ | $86.57 \pm 1.17$ |
| <i>MMLBLM</i> | $6.51 \pm 0.39$ | $85.21 \pm 1.32$ |
| <i>varGDM</i> | $6.63 \pm 0.28$ | $83.75 \pm 1.28$ |
| <i>varDM</i>  | $6.58 \pm 0.33$ | $81.38 \pm 1.13$ |
| <i>varGM</i>  | $6.55 \pm 0.25$ | $79.43 \pm 1.49$ |

resulting in 1200 images (90 Anger, 111 Disgust, 90 Fear, 270 Joy, 120 Sadness, 219 Surprise and 300 Neutral). Sample images from this database with different facial expressions are shown in Figure 6. We use a preprocessing step to normalize and crop original images into  $110 \times 150$  pixels to reduce the influences of background. Then, we extract LBP-TOP features from the cropped images and represent each of it as a normalized histogram. Finally, our method is applied to cluster the obtained normalized histograms. We have also compared our approach with: *MMLBLM*, *varGDM*, *varDM* and *varGM*. We evaluate the performance of each algorithm by repeating it 30 times. The confusion matrix for the Cohn-Kanade database obtained by *varBLM* is shown in Figure 7. As we can notice, the majority of misclassification errors are caused by “Fear” being confused with “Joy”. Table 3 shows the average recognition accuracy of the facial expression sequences and the average number of clusters obtained by each algorithm. According to this table, *varBLM* outperforms the other three algorithms in terms of the highest classification accuracy (86.57%) and the most accurately detected number of categories (6.65).

## 5 Conclusion

We have proposed a variational inference framework for learning finite Beta-Liouville mixture model. Within this framework, model parameters and the number of components are determined simultaneously. Extensive experiments have been conducted and have involved artificial data and real applications namely dynamic textures clustering and facial expression recognition approached as proportional data clustering problems. The obtained results are very promising. Future works could be devoted to the extension of the proposed model to the infinite case using Dirichlet processes or the development of an online learning approach to tackle the crucial problem of dynamic data clustering.

## Acknowledgments

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC). We would like to thank Prof. Jeffery Cohn for making the Cohn-Kanade database available.

## References

[Attias, 1999a] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proc.*

- of the *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 21–30, 1999.
- [Attias, 1999b] H. Attias. A variational Bayes framework for graphical models. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 209–215, 1999.
- [Bishop, 2006] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [Black and Yacoob, 1997] M.J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.
- [Bouguila and Ziou, 2006] N. Bouguila and D. Ziou. Unsupervised selection of a finite Dirichlet mixture model: An mml-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):993–1009, 2006.
- [Bouguila and Ziou, 2007] N. Bouguila and D. Ziou. High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1716–1731, 2007.
- [Bouguila, 2012] N. Bouguila. Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2184–2202, 2012.
- [Constantinopoulos *et al.*, 2006] C. Constantinopoulos, M.K. Titsias, and A. Likas. Bayesian feature and model selection for gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1013–1018, 2006.
- [Corduneanu and Bishop, 2001] A. Corduneanu and C. M. Bishop. Variational Bayesian model selection for mixture distributions. In *Proc. of the 8th International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 27–34, 2001.
- [Doretto *et al.*, 2003] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.
- [Figueiredo and Jain, 2002] M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.
- [Jordan *et al.*, 1998] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In *Learning in Graphical Models*, pages 105–162, 1998.
- [Kanade *et al.*, 2000] T. Kanade, J.F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition (AFGR)*, pages 46–53, 2000.
- [Koelstra *et al.*, 2010] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1940–1954, 2010.
- [Kwatra *et al.*, 2003] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003.
- [Lucey *et al.*, 2010] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 94–101, 2010.
- [Mackay, 1995] D. J. C. Mackay. Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.
- [McLachlan and Peel, 2000] G. McLachlan and D. Peel. *Finite Mixture Models*. New York: Wiley, 2000.
- [Pantic and Rothkrantz, 1999] M. Pantic and L.J.M. Rothkrantz. An expert system for multiple emotional classification of facial expressions. In *Proc. of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 113–120, 1999.
- [Péteri *et al.*, 2010] R. Péteri, S. Fazekas, and M.J. Huiskes. DynTex : a comprehensive database of dynamic textures. *Pattern Recognition Letters*, 31(12):1627–1632, 2010.
- [Ravichandran *et al.*, 2013] A. Ravichandran, R. Chaudhry, and R. Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):342–353, 2013.
- [Shan *et al.*, 2009] C. Shan, S. Gong, and P.W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.
- [Wang and Titterton, 2006] B. Wang and D. M. Titterton. Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model. *Bayesian Analysis*, 1(3):625–650, 2006.
- [Watanabe and Watanabe, 2007] K. Watanabe and S. Watanabe. Stochastic complexity for mixture of exponential families in generalized variational bayes. *Theoretical Computer Science*, 387(1):4–17, 2007.
- [Watanabe *et al.*, 2009] K. Watanabe, S. Akaho, S. Omachi, and M. Okada. Variational Bayesian mixture model on a subspace of exponential family distributions. *IEEE Transactions on Neural Networks*, 20(11):1783–1796, 2009.
- [Zhao and Pietikainen, 2007] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.