Neural Process Lett
DOI 10.1007/s11063-013-9293-x

Non-Gaussian Data Clustering via Expectation Propagation Learning of Finite Dirichlet Mixture Models and Applications

Wentao Fan · Nizar Bouguila

Abstract Learning appropriate statistical models is a fundamental data analysis task which has been the topic of continuing interest. Recently, finite Dirichlet mixture models have proved to be an effective and flexible model learning technique in several machine learning and data mining applications. In this article, the problem of learning and selecting finite Dirichlet mixture models is addressed using an expectation propagation (EP) inference framework. Within the proposed EP learning method, for finite mixture models, all the involved parameters and the model complexity (i.e. the number of mixture components), can be evaluated simultaneously in a single optimization framework. Extensive simulations using synthetic data along with two challenging real-world applications involving automatic image annotation and human action videos categorization demonstrate that our approach is able to achieve better results than comparable techniques.

Keywords Mixture models · Dirichlet distribution · Expectation propagation · Image annotation · Human action videos categorization

1 Introduction

As the availability of digital multimedia data (e.g. images, videos or text) continue to increase, powerful approaches for analyzing, managing and clustering these data become extremely important in various fields including machine learning, data mining, computer vision, etc. In particular, clustering is a common unsupervised learning technique used to discover groups of similar examples within a data set which is crucial for knowledge acquisition and has

W. Fan
Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada
e-mail: wenta_fa@encs.concordia.ca

N. Bouguila (✉)
The Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

Published online: 10 March 2013

been the subject of extensive research. A powerful approach to clustering is the use of finite mixture models which has important advantages such as its flexibility and addressing unsupervised learning in a formal way [11,23]. A finite mixture model is formed by taking linear combinations of a finite number of basic distributions. These basic distributions are called components of the mixture model. Traditionally, several clustering methods have been based implicitly or explicitly on the Gaussian assumption [11,17]. Although assuming that the per-components densities are Gaussians has been widely considered in the past, due to their approximation properties and simplicity, recent works have shown that other models may provide better fitting capabilities in the case of non-Gaussian data. For instance, it has been shown that the Dirichlet mixture can be a better alternative in several applications especially those involving proportional data in [2–5]. Therefore, motivated by its flexibility and good performance obtained in these previous works, we shall focus in this paper on the finite Dirichlet mixture model.

A maximum likelihood (ML) approach based on the expectation-maximization (EM) algorithm has been proposed in [5] to learn finite Dirichlet mixtures. Although the EM algorithm is commonly used to estimate the parameters of finite mixture models, it has several limitations such as the fact that it only guarantees convergence to a local maximum of the likelihood and the necessity to know the appropriate number of components in advance. The later limitation is especially serious since choosing too many components leads generally to over-fitting and the specification of a comparatively small number of components causes under-fitting. A common solution for selecting appropriate number of mixture components is to consider model selection criteria such as those discussed in [4] where a minimum message length criterion (MML) has been developed. Recently, some research works have shown that the drawbacks of the EM algorithm can be addressed by adopting an expectation propagation (EP) framework [26,27]. As a better alternative to the EM, the EP framework has received considerable attention and has provided good generalization performance in many applications including finite Gaussian mixtures learning [7,26]. EP is a recursive approximation scheme based on the minimization of a Kullback-Leibler (KL) divergence between the true model's posterior and an approximation [26,27]. It can provide full posterior distribution of model parameters that represent the underlying structure of the data. Notice that, the EP algorithm is an extension of the assumed-density filtering (ADF) [22] which is a one pass, sequential approximation method. In contrast to the ADF, the order of the input data points is not crucial in the EP inference and its inference accuracy could be improved by re-using the data points many times. In the case of finite mixture modeling, unlike the ML method in which the number of component is detected by applying some typical criteria, the EP inference framework can estimate model parameters and determine the number of component (i.e. model selection) simultaneously.

The major contribution of this paper is that we construct a statistical Bayesian framework based on finite Dirichlet mixture models using EP inference framework, such that the model complexity selection and the model-parameters estimation can be performed simultaneously in a single optimization framework. Furthermore, we apply the proposed approach to solve two challenging problems involving automatic image annotation and human action videos categorization. We are motivated mainly by the good results obtained in the past using EP techniques in machine learning applications in general [12,21] and for the finite mixture modeling in particular [7].

The rest of this paper is organized as follows. Section 2 introduces in sufficient details the finite Dirichlet mixture model. In Sect 3, we describe our EP inference procedure for the proposed model learning. Section 4 presents results on synthetic data and two challenging real applications. Section 5 closes with conclusions.

2 Finite Dirichlet Mixture Model

In this section, we briefly review the finite Dirichlet mixture model that we shall propose a new learning approach and algorithm for it. Assume that we have a D-dimensional vector $\mathbf{X} = (X_1, \ldots, X_D)$ which follows a Dirichlet distribution with positive parameters $\boldsymbol{\alpha}_j = (\alpha_{j1}, \ldots, \alpha_{jD})$, then the probability density function of $\mathbf{X}$ is given by

$$\mathrm{Dir}(\mathbf{X}|\boldsymbol{\alpha}_j) = \frac{\Gamma(\sum_{l=1}^{D} \alpha_{jl})}{\prod_{l=1}^{D} \Gamma(\alpha_{jl})} \prod_{l=1}^{D} X_l^{\alpha_{jl}-1} \tag{1}$$

where $\sum_{l=1}^{D} X_l = 1$ and $0 \leq X_l \leq 1$ for $l = 1, \ldots, D$. The mean and variance of the Dirichlet distribution are given by

$$\mathbb{E}[X_l] = \frac{\alpha_{jl}}{\sum_{l=1}^{D} \alpha_{jl}} \tag{2}$$

$$\mathrm{Var}[X_l] = \frac{\alpha_{jl}(\sum_{l=1}^{D} \alpha_{jl} - \alpha_{jl})}{(\sum_{l=1}^{D} \alpha_{jl})^2 (\sum_{l=1}^{D} \alpha_{jl} + 1)} \tag{3}$$

Assume now we have observed a set of N vectors $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, where each vector $\mathbf{X}_i = (X_{i1}, \ldots, X_{iD})$ is represented in a D-dimensional space and assumed to be generated from a finite Dirichlet mixture model with M components as [5]

$$p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}) = \sum_{j=1}^{M} \pi_j \mathrm{Dir}(\mathbf{X}|\boldsymbol{\alpha}_j) \tag{4}$$

where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M)$, and $\mathrm{Dir}(\mathbf{X}|\boldsymbol{\alpha}_j)$ is the Dirichlet distribution of component j with its own parameters $\boldsymbol{\alpha}_j = (\alpha_{j1}, \ldots, \alpha_{jD})$. $\boldsymbol{\pi} = \{\pi_j\}$ are called the mixing coefficients that are subject to the constraints $0 \leq \pi_j \leq 1$ and $\sum_{j=1}^{M} \pi_j = 1$. Accordingly, the likelihood function of $\mathcal{X}$ can be written as

$$p(\mathcal{X}|\boldsymbol{\pi}, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \left[\sum_{j=1}^{M} \pi_j \mathrm{Dir}(\mathbf{X}_i|\boldsymbol{\alpha}_j) \right] \tag{5}$$

3 EP-Based Learning of the Finite Dirichlet Mixture Model

3.1 Expectation Propagation

In this subsection, a brief introduction to the EP approximation scheme is presented. Consider an observed data set of N i.i.d vectors $\mathcal{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ which follows a model with unknown parameter Θ, then the joint distribution of $\mathcal{X}$ and Θ can be represented in the form of a product of factors as [12,21]

$$p(\mathcal{X}, \Theta) = \prod_i f_i(\Theta) \tag{6}$$

One possible factorization of Eq. (6) is that there is one factor $f_i(\Theta) = p(\mathbf{X}_i|\Theta)$ for each data point X_i, along with a factor $f_0(\Theta) = p(\Theta)$ which corresponds to the prior. The main idea of the EP algorithm is to approximate the posterior distribution $p(\Theta|\mathcal{X})$ by a product of factors:

$$q^*(\Theta) = \frac{\prod_i \widetilde{f}_i(\Theta)}{\int \prod_i \widetilde{f}_i(\Theta) d\Theta} \tag{7}$$

where each factor $\widetilde{f}_i(\Theta)$ is an approximation to $f_i(\Theta)$. In the EP learning framework, the first step is to initialize all the factors $\widetilde{f}_i(\Theta)$. Then, each factor is optimized sequentially in the context of the remaining factors. For a specific factor $f_j(\Theta)$, we first remove it from the current approximation to the posterior by

$$q^{\setminus j}(\Theta) = \frac{q^*(\Theta)}{\widetilde{f}_j(\Theta)} \tag{8}$$

Then, a new distribution can be obtained by combining Eq. (8) with the true factor $f_j(\Theta)$ as

$$\widehat{p}(\Theta) = \frac{f_j(\Theta) q^{\setminus j}(\Theta)}{\int f_j(\Theta) q^{\setminus j}(\Theta) d\Theta} \tag{9}$$

Next, the approximated posterior $q^*(\Theta)$ can be evaluated by minimizing the KL divergence: $\mathrm{KL}\big(\widehat{p}(\Theta) \parallel q^*(\Theta)\big)$. This is achieved by matching the sufficient statistics of $q^*(\Theta)$ to the corresponding moments of $\widehat{p}(\Theta)$. Then, the approximating factor $\widetilde{f}_j(\Theta)$ can be updated as

$$\widetilde{f}_j(\Theta) = Z_j \frac{q^*(\Theta)}{q^{\setminus j}(\Theta)} \tag{10}$$

where $Z_j = \int f_j(\Theta) q^{\setminus j}(\Theta) d\Theta$ is a normalization constant. In EP learning, each factor can be updated iteratively in the context of remaining factors as described in the above steps until convergence. For more details about the EP learning framework, the reader is referred to [26,27].

3.2 Expectation Propagation for the Dirichlet Mixture

In the following section, we adopt the EP framework for learning the Dirichlet mixture model. In Bayesian modeling, we need to assign to each unknown parameter a prior distribution. In our case, a Dirichlet distribution with positive parameters $\mathbf{a} = (a_1, \ldots, a_M)$ is adopted as the conjugate prior of $\boldsymbol{\pi}$:

$$p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\mathbf{a}) = \frac{\Gamma(\sum_{j=1}^{M} a_j)}{\prod_{j=1}^{M} \Gamma(a_j)} \prod_{j=1}^{M} \pi_j^{a_j - 1} \tag{11}$$

For the parameter $\boldsymbol{\alpha}_j$ of the Dirichlet mixture model, since the formal conjugate prior of Dirichlet is analytically intractable, we adopt a Gaussian distribution to approximate the prior which has shown good results in the case of the Beta (i.e. one-dimensional case of the Dirichlet) [21]. This is motivated by the fact that the Gaussian allows analytically tractable calculations and can fairly capture the correlation among the elements in $\boldsymbol{\alpha}$. Thus, a D-dimensional Gaussian, with mean vector $\boldsymbol{\mu}_j$ and covariance matrix A_j, is considered for $\boldsymbol{\alpha}_j$, such that:

$$p(\boldsymbol{\alpha}_j) = \mathcal{N}(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_j, A_j) = \frac{|A_j|^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_j)^T A_j (\boldsymbol{\alpha}_j - \boldsymbol{\mu}_j) \right) \tag{12}$$

The first step in the EP inference is to initialize all the approximating factors $\widetilde{f}_i(\Theta)$. This is done by initializing all the involved hyperparameters $\{a_j, \boldsymbol{\mu}_j, A_j\}$. Next, we initialize

the posterior approximation $q^*(\Theta)$ by setting $q^*(\Theta) \propto \prod_i \widetilde{f}_i(\Theta)$. Therefore, we can easily compute the hyperparameters of $q^*(\Theta)$ as

$$a_j^* = \sum_i a_{i,j} - N \tag{13}$$

$$\boldsymbol{\mu}_j^* = \Big(\sum_i A_{i,j}^{-1}\Big)\Big(\sum_i A_{i,j}\boldsymbol{\mu}_{i,j}\Big) \tag{14}$$

$$A_j^* = \sum_i A_{i,j} \tag{15}$$

In order to update the factor $\widetilde{f}_i(\Theta)$, we have to remove it from the posterior $q^*(\Theta)$ as shown in Eq. (8). Then, the corresponding hyperparameters can be computed analytically as

$$a_j^{\backslash i} = a_j^* - a_{i,j} + 1\ , \tag{16}$$

$$\boldsymbol{\mu}_j^{\backslash i} = \big(A_j^{\backslash i}\big)^{-1}\big(A_j^*\boldsymbol{\mu}_j^* - A_{i,j}\boldsymbol{\mu}_{i,j}\big) \tag{17}$$

$$A_j^{\backslash i} = A_j^* - A_{i,j} \tag{18}$$

Next, the updated posterior $\widehat{p}(\Theta)$ can be calculated as

$$\widehat{p}(\Theta) = \frac{1}{Z_i} f_i(\Theta) q^{\backslash i}(\Theta) \tag{19}$$

where

$$Z_i = \int f_i(\Theta) q^{\backslash i}(\Theta) d\Theta = \sum_{j=1}^{M} \frac{a_{i,j}}{\sum_j a_{i,j}} \int \mathrm{Dir}(\mathbf{X}_i|\boldsymbol{\alpha}_j) N(\boldsymbol{\alpha}_j|\mu_j^{\backslash i}, A_j^{\backslash i}) d\boldsymbol{\alpha}_j \tag{20}$$

Notice that, the integration in Eq. (20) is intractable and that the moments cannot be calculated analytically. One way to tackle this problem is to adopt the Laplace approximation to approximate the integrand with a Gaussian distribution as suggested in [21]. First, we can define a normalized distribution for the integrand in Eq. (20) which is indeed a product of a Dirichlet distribution and a Gaussian distribution as

$$\mathcal{H}(\boldsymbol{\alpha}_j) = \frac{h(\boldsymbol{\alpha}_j)}{\int h(\boldsymbol{\alpha}_j) d\boldsymbol{\alpha}_j} \tag{21}$$

where

$$h(\boldsymbol{\alpha}_j) = \mathrm{Dir}(\mathbf{X}_i|\boldsymbol{\alpha}_j)\mathcal{N}(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_j^{\backslash i}, A_j^{\backslash i}) \tag{22}$$

Then, we can obtain the logarithm of $h(\theta_{jl})$ as

$$\begin{aligned} \ln h(\boldsymbol{\alpha}_j) = \ln \frac{\sum_{l=1}^{D} \Gamma(\alpha_{jl})}{\prod_{l=1}^{D} \Gamma(\alpha_{jl})} + \sum_{l=1}^{D} (\alpha_{jl} - 1) \ln X_{il} \\ -\frac{1}{2}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_j^{\backslash i})^T A_j^{\backslash i} (\boldsymbol{\alpha}_j - \boldsymbol{\mu}_j^{\backslash i}) + \text{const.} \end{aligned} \tag{23}$$

Subsequently, we can calculate the first and second derivatives with respect to $\boldsymbol{\alpha}_j$ as

$$\begin{aligned}\frac{\partial \ln h(\boldsymbol{\alpha}_j)}{\partial \boldsymbol{\alpha}_j} &= \begin{bmatrix} \partial \ln h(\boldsymbol{\alpha}_j)/\partial \alpha_{j1} \\ \vdots \\ \partial \ln h(\boldsymbol{\alpha}_j)/\partial \alpha_{jD} \end{bmatrix} \\ &= \begin{bmatrix} \Psi(\sum_{l=1}^{D} \alpha_{jl}) - \Psi(\alpha_{j1}) + \ln X_{i1} \\ \vdots \\ \Psi(\sum_{l=1}^{D} \alpha_{jl}) - \Psi(\alpha_{jD}) + \ln X_{iD} \end{bmatrix} - A_j^{\backslash i}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_j^{\backslash i}) \end{aligned} \tag{24}$$

and

$$\begin{aligned}\frac{\partial^2 \ln h(\boldsymbol{\alpha}_j)}{\partial \boldsymbol{\alpha}_j^2} &= \begin{bmatrix} \partial^2 \ln h(\boldsymbol{\alpha}_j)/\partial \alpha_{j1}^2 & \cdots & \partial^2 \ln h(\boldsymbol{\alpha}_j)/\partial \alpha_{j1}\partial \alpha_{jD} \\ \vdots & \ddots & \vdots \\ \partial^2 \ln h(\boldsymbol{\alpha}_j)/\partial \alpha_{jD}\partial \alpha_{j1} & \cdots & \partial^2 \ln h(\boldsymbol{\alpha}_j)/\partial \alpha_{jD}^2 \end{bmatrix} \\ &= \begin{bmatrix} \Psi'(\sum_{l=1}^{D} \alpha_{jl}) - \Psi'(\alpha_{j1}) & \cdots & \Psi'(\sum_{l=1}^{D} \alpha_{jl}) \\ \vdots & \ddots & \vdots \\ \Psi'(\sum_{l=1}^{D} \alpha_{jl}) & \cdots & \Psi'(\sum_{l=1}^{D} \alpha_{jl}) - \Psi'(\alpha_{jD}) \end{bmatrix} - A_j^{\backslash i} \end{aligned} \tag{25}$$

where $\Psi(\cdot)$ is the digamma function. In the Laplace method the goal is to find a Gaussian approximation which is centered on the mode of the distribution $\mathcal{H}(\boldsymbol{\alpha}_j)$. We could obtain the mode $\boldsymbol{\alpha}_j^*$ numerically by setting the first derivative of Eq. (24) to 0. Then, we can approximate $h(\boldsymbol{\alpha}_j)$ using its mode as

$$h(\boldsymbol{\alpha}_j) \simeq h(\boldsymbol{\alpha}_j^*) \exp\left(-\frac{1}{2}(\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_j^*)\widehat{A}_j(\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_j^*)\right) \tag{26}$$

where

$$\widehat{A}_j = -\frac{\partial^2 \ln h(\boldsymbol{\alpha}_j)}{\partial \boldsymbol{\alpha}_j^2}|_{\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_j^*} \tag{27}$$

Therefore, the integration of $h(\boldsymbol{\alpha}_j)$ can be approximated by using Eq. (26) as

$$\begin{aligned}\int h(\boldsymbol{\alpha}_j)d\boldsymbol{\alpha}_j &\simeq h(\boldsymbol{\alpha}_j^*) \int \exp\left(-\frac{1}{2}(\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_j^*\right) \widehat{A}_j(\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_j^*))d\boldsymbol{\alpha}_j \\ &= h(\boldsymbol{\alpha}_j^*)\frac{(2\pi)^{D/2}}{|\widehat{A}_j|^{1/2}} \end{aligned} \tag{28}$$

Hence, we can rewrite Eq. (20) as following:

$$Z_i = \sum_{j=1}^{M} \frac{a_{i,j}}{\sum_j a_{i,j}} h(\boldsymbol{\alpha}_j^*)\frac{(2\pi)^{D/2}}{|\widehat{A}_j|^{1/2}} \tag{29}$$

Then, we can revise the posterior distribution $q^*(\Theta)$ by matching its sufficient statistics to the corresponding moments of $\widehat{p}(\Theta)$. This is done by calculating the partial derivative of

$\ln Z_i$ with respect to the model hyperparameters. For $a_j^{\setminus i}$, we can get

$$\begin{aligned}\nabla_{a_j}^{\setminus i} \ln Z_i &= \frac{1}{Z_i} \int f_i(\Theta) \frac{q^{\setminus i}(\Theta)}{q^{\setminus i}(\pi_j^{\setminus i})} \frac{\partial}{\partial a_j^{\setminus i}} q^{\setminus i}(\pi_j^{\setminus i}) d\Theta \\ &= \int \widehat{p}(\Theta) \Big[\ln \pi_j^{\setminus i} + \Psi\left(\sum_{j=1}^{M} a_j^{\setminus i} \right) - \Psi(a_j^{\setminus i}) \Big] d\Theta \\ &= E_{\widehat{p}}[\ln \pi_j] + \Psi\left(\sum_{j=1}^{M} a_j^{\setminus i} \right) - \Psi(a_j^{\setminus i}) \end{aligned} \tag{30}$$

By applying moment matching, we obtain

$$E_{\widehat{p}}[\ln \pi_j] = E_{q^*}[\ln \pi_j] = \Psi(a_j^*) - \Psi\left(\sum_{j=1}^{M} a_j^* \right) \tag{31}$$

Similarly, we can compute the partial derivatives of $\ln Z_i$ with respect to the other model hyperparameters:

$$\begin{aligned}\nabla_{\boldsymbol{\mu}_j}^{\setminus i} \ln Z_i &= \frac{1}{Z_i} \int f_i(\Theta) \frac{q^{\setminus i}(\Theta)}{q^{\setminus i}(\boldsymbol{\alpha}_j^{\setminus i})} \frac{\partial}{\partial \boldsymbol{\mu}_j^{\setminus i}} q^{\setminus i}(\boldsymbol{\alpha}_j^{\setminus i}) d\Theta \\ &= \int \widehat{p}(\Theta) \big[A_j^{\setminus i} \boldsymbol{\alpha}_j^{\setminus i} - A_j^{\setminus i} \boldsymbol{\mu}_j^{\setminus i} \big] d\Theta \\ &= A_j^{\setminus i} E_{\widehat{p}}[\boldsymbol{\alpha}_j] - A_j^{\setminus i} \boldsymbol{\mu}_j^{\setminus i} \end{aligned} \tag{32}$$

$$\begin{aligned}\nabla_{A_j}^{\setminus i} \ln Z_i &= \frac{1}{Z_i} \int f_i(\Theta) \frac{q^{\setminus i}(\Theta)}{q^{\setminus i}(\boldsymbol{\alpha}_j^{\setminus i})} \frac{\partial}{\partial A_j^{\setminus i}} q^{\setminus i}(\boldsymbol{\alpha}_j^{\setminus i}) d\Theta \\ &= \int \widehat{p}(\Theta) \left\{ \frac{1}{2} |(A_j^{\setminus i})^{-1}| - \frac{1}{2} \left[\sum_{l=1}^{D} (\alpha_{jl}^{\setminus i})^2 - 2\alpha_{jl}^{\setminus i} \mu_{jl}^{\setminus i} + (\mu_{jl}^{\setminus i})^2 \right] \right\} d\Theta \\ &= \frac{1}{2} \left\{ |(A_{jl}^{\setminus i})^{-1}| - \left[\sum_{l=1}^{D} E_{\widehat{p}}[\alpha_{jl}^2] - 2E_{\widehat{p}}[\alpha_{jl}] \mu_{jl}^{\setminus i} + (\mu_{jl}^{\setminus i})^2 \right] \right\} \end{aligned} \tag{33}$$

The right hand sides in the above equations can be computed analytically by using Eq. (29). Furthermore, the expectations in the above equations can be acquired by applying the moment matching technique as

$$E_{\widehat{p}}[\boldsymbol{\alpha}_j] = E_{q^*}[\boldsymbol{\alpha}_j] = \boldsymbol{\mu}_{jl}^* \tag{34}$$

$$E_{\widehat{p}}[\boldsymbol{\alpha}_j^2] = E_{q^*}[\boldsymbol{\alpha}_j^2] = (\boldsymbol{\mu}_j^*)^2 \tag{35}$$

By substituting the above expectations into the corresponding partial derivative equations, we can update the hyperparameters of $q^*(\Theta)$. After obtaining $q^*(\Theta)$ and $q^{\setminus i}(\Theta)$, we can update the revised hyperparameters for the approximating factor f_i as

$$a_{i,j} = a_j^* - a_j^{\setminus i} + 1 \tag{36}$$

$$\boldsymbol{\mu}_{i,j} = A_{i,j}^{-1} (A_j^* \boldsymbol{\mu}_j^* - A_j^{\setminus i} \boldsymbol{\mu}_j^{\setminus i}) \tag{37}$$

$$A_{i,j} = A_j^* - A_j^{\setminus i} \tag{38}$$

The above procedure is repeated until the hyperparameters of the approximating factor converge. The same procedure is applied sequentially for the remaining factors. Moreover, we can estimate the expected values of the mixing coefficients as

$$E[\pi_j] = \frac{a_j^*}{\sum_j a_j^*} \tag{39}$$

The complete learning process is summarized in Algorithm 1.[1]

Algorithm 1 EP learning of finite Dirichlet mixtures

1: Choose the initial number of components.
2: Initialize the approximating factors $\widetilde{f}_i(\Theta)$ by initializing all the involved hyperparameters $\{a_j, \boldsymbol{\mu}_j, A_j\}$.
3: Initialize the posterior approximation by setting $q^*(\Theta) \propto \prod_i \widetilde{f}_i(\Theta)$. The hyperparameters of of $q^*(\Theta)$ are calculated by Eqs. (13)–(15).
4: **repeat**
5: Choose a factor $\widetilde{f}_i(\Theta)$ to refine.
6: Remove $\widetilde{f}_i(\Theta)$ from the posterior $q^*(\Theta)$ by division $q^{\setminus i}(\Theta) = q^*(\Theta)/\widetilde{f}_i(\Theta)$.
7: Evaluate the new posterior by setting the sufficient statistics (moments) of $q^*(\Theta)$ to the corresponding moments of $\widehat{p}(\Theta)$.
8: Update the factor $\widetilde{f}_i(\Theta)$ by updating the corresponding hyperparameters as in Eqs. (36)–(38).
9: **until** Convergence criterion is reached.
10: Compute the estimated values of the mixing coefficients π_j as in Eq. (39).
11: Detect the optimal number of components M by eliminating the components with small mixing coefficients close to 0.

4 Experimental Results

In this section, the effectiveness of the proposed EP-based framework for learning the Dirichlet mixture model (denoted as *EPDMM*) is tested on both synthetic data and two real-world applications namely automatic image annotation and human action videos categorization. In our experiments, we initialize the number of components M to 15. The specific choice for the hyperparameters of each factor $f_i(\Theta)$ in all the experiments is $(a_{i,j}, \boldsymbol{\mu}_{i,j}, A_{i,j}) = (0.1, 0.5, 0.01)$. Notice that these specific choices were found convenient according to our experiments.

4.1 Synthetic Data

The aim of the synthetic data is to evaluate the performance of the proposed EP-based algorithm (*EPDMM*) and compare it with the ML-based technique proposed in [5]. First, we investigate the accuracy of *EPDMM* in terms of estimation (estimating the model's parameters) and selection (selecting the number of components of the mixture model) on four three-dimensional synthetic data sets. Note that, here we choose $D = 3$ purely for ease of representation. We ran the proposed algorithm 10 times. Table 1 shows the actual and average estimated parameters obtained from *EPDMM* for each data set. Based on this table, we can see that our algorithm is able to correctly estimate both the parameters and the mixing coefficients of the synthetic mixture models for all data sets. The resultant mixtures for these data sets are illustrated in Fig. 1. Figure 2 demonstrates the estimated mixing coefficients of

[1] The complete source code of this work is available upon request.

Table 1 Parameters of the different generated data sets

	N_j	j	α_{j1}	α_{j2}	α_{j3}	π_j	$\hat{\alpha}_{j1}$	$\hat{\alpha}_{j2}$	$\hat{\alpha}_{j3}$	$\hat{\pi}_j$
Data set 1	100	1	10	15	20	0.50	9.31	14.29	21.04	0.506
($N = 200$)	100	2	5	20	12	0.50	4.86	19.55	11.71	0.494
Data set 2	100	1	10	15	20	0.25	10.12	15.75	19.28	0.253
($N = 400$)	100	2	5	20	12	0.25	5.37	20.66	11.87	0.244
	200	3	30	13	26	0.50	31.64	12.51	27.19	0.503
Data set 3	150	1	10	15	20	0.25	10.41	14.37	20.34	0.246
($N = 600$)	150	2	5	20	12	0.25	4.63	21.12	12.72	0.241
	150	3	30	13	26	0.25	28.96	13.58	24.85	0.257
	150	4	40	8	22	0.25	41.82	8.72	21.28	0.256
Data set 4	160	1	10	15	20	0.20	9.49	14.45	19.27	0.191
($N = 800$)	160	2	5	20	12	0.20	5.58	20.34	12.62	0.208
	160	3	30	13	26	0.20	29.17	12.61	27.44	0.193
	160	4	40	8	22	0.20	38.78	7.69	22.56	0.207
	160	5	4	33	8	0.20	4.31	34.83	7.45	0.201

N denotes the total number of elements, N_j denotes the number of elements in cluster j. α_{j1}, α_{j2}, α_{j3} and π_j are the real parameters. $\hat{\alpha}_{j1}$, $\hat{\alpha}_{j2}$, $\hat{\alpha}_{j3}$ and $\hat{\pi}_j$ are the estimated parameters by *EP*

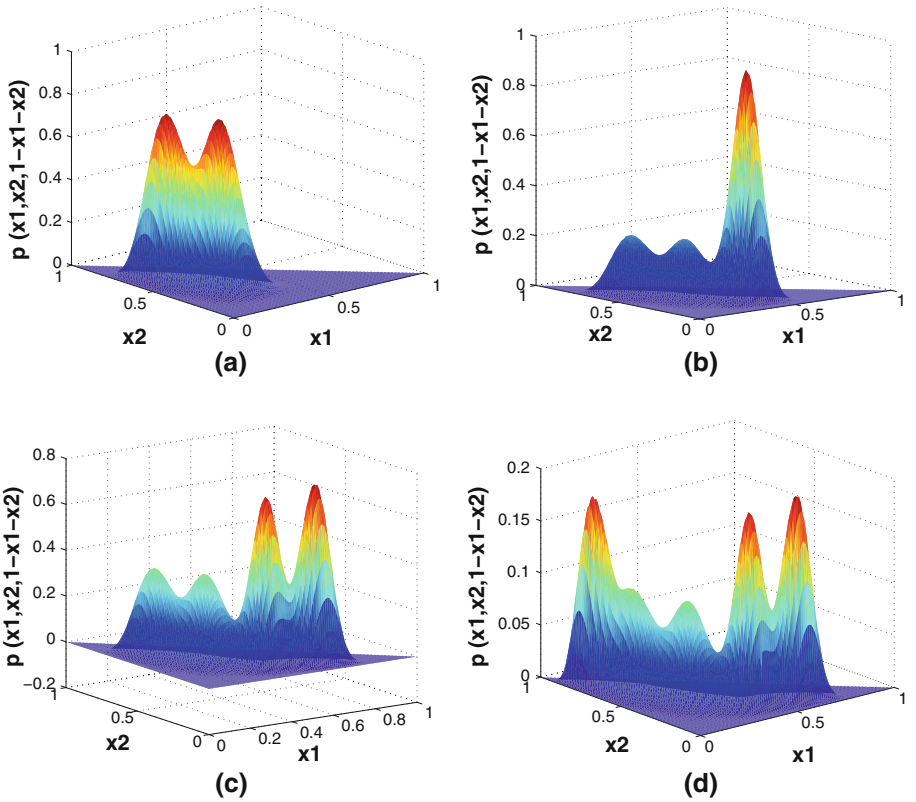

Fig. 1 Estimated mixture densities for the synthetic data sets. **a** Data set 1, **b** Data set 2, **c** Data set 3, **d** Data set 4

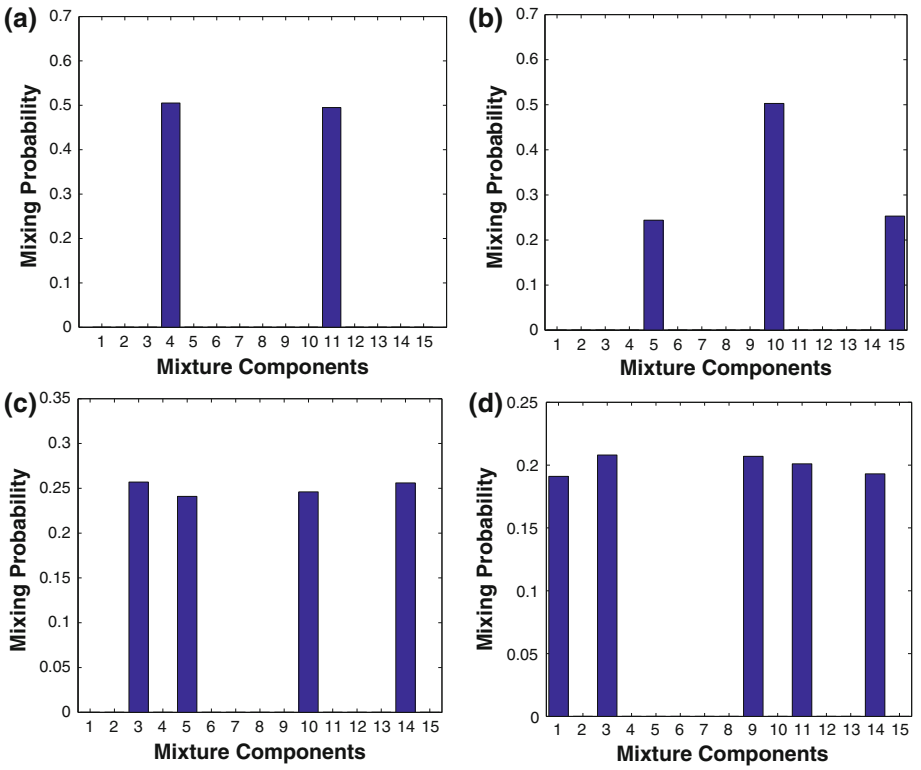

Fig. 2 Estimated mixing coefficients for the synthetic data sets. **a** Data set 1, **b** Data set 2, **c** Data set 3, **d** Data set 4

each mixture component for each data set. According to this figure, we can see that redundant components have estimated mixing coefficients close to 0 after convergence. By removing the components with very small mixing coefficients (close to 0), we obtain the correct number of components for each generated data set.

For comparison, we have also performed the ML-based approach to learn finite Dirichlet mixture models (*DMM*) as proposed in [5] on these four synthetic data sets. According to our results, the *DMM* can provide comparable results in estimating the model parameters of finite Dirichlet mixture models as *EPDMM*. Nevertheless, the dominant factor of the *EPDMM* is the computational time which is shown in Table 2.

Table 2 Average computational time (in seconds) required before convergence for *EPDMM* and *DMM*

Method	*EPDMM*	*DMM*
Data set 1	3.97	9.58
Data set 2	5.78	16.34
Data set 3	10.21	29.16
Data set 4	15.08	43.65

4.2 Automatic Image Annotation

Automatic image annotation (also known as automatic image tagging or linguistic indexing) is the process of automatically assigning captions or keywords to a digital image. It is a crucial step in image retrieval systems to organize and locate images of interest in large volumes of images. During the last decade, automatic image annotation has drawn significant attention and has been the topic of extensive research [6,10,20,28,32]. One of the most successful approaches for automatic image annotation is to divide this problem into two independent steps where the first step categorizes images and the second one affects labels to them using the top ranked categories (see, for instance, [6,8]). Thus, the goal of this experiment is to develop an effective automatic image annotation approach, based on the methodology proposed in [8], via categorization results obtained with the proposed *EPDMM* using a bag of visual key words representation.

4.2.1 Experimental Design

In the categorization stage, the proposed *EPDMM* is integrated with the probabilistic latent semantic analysis (pLSA) model [13] to categorize images through a bag of key visual words representation. First, the Difference-of-Gaussian (DoG) detector [24] is applied to detect interest points (or keypoints) in input images followed with PCA-SIFT descriptors[2] [14] extracted from each image and resulting in a 36-dimensional vector for each key point. Then, we build a visual vocabulary by quantizing these PCA-SIFT vectors into visual words using the K-Means algorithm. Notice that, the vocabulary size is set to 1,000 in our experiment. Each image is then represented by a frequency histogram over the visual words. Subsequently, we apply the pLSA model on the obtained histograms to represent each image by a 50-dimensional proportional vector where 50 is the number of latent aspects. Finally, our *EPDMM* is applied to cluster the images.

The obtained categorization results are then exploited to perform image annotation. In our experiment, the performance of image annotation is affected by three aspects as proposed in [8]: (1) the frequency of occurrence of potential tags based on the categorization results; (2) saliency of the given tags; (3) the congruity of a word among all the candidate tags. Assume that we have a training image data set that contains several categories. Each category is annotated by 4–5 tags where common tags may appear in different categories. First, all the tags from each category are collected together. The total number of categories in the data set is denoted as C and the number of categories that have each unique tag t is represented as $F(t)$. Then, tag saliency can be evaluated similarly as for inverse document frequency in the field of document retrieval. For a testing image, a ranked list of predicted categories is generated according to the Bayes' decision rule via classification. Next, the top 5 predicted categories are chosen and the union of all involved unique tags denoted as $U(I)$ forms the set of candidate tags. Thus, we define $f(t|I)$ as the frequency of the occurrence of each unique tag t among the top 5 predicted categories. We follow the idea proposed in [8] to determine the word congruity using WordNet[3] [25] with the Leacock and Chowdrow measure [18]. Thus, the congruity for a candidate tag t can be calculated by [8]:

$$G(t|I) = \frac{d_{tot}(I)}{d_{tot}(I) + |U(I)| \sum_{x \in U(I)} d_{LCH}(x,t)} \tag{40}$$

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

[3] WordNet is a large lexical database for English, which groups English words into sets of cognitive synonyms called synsets.

Table 3 The classification accuracies computed by different algorithms

Method	Accuracy (%)
EPDMM	77.45
SC-GM	75.17
EPGMM	73.93
K-Means	70.29

Table 4 Performance evaluation on the automatic annotation system based on different categorization methods

Method	*EPDMM*	*SC-GM*	*EPGMM*	*K-Means*
Mean precision (%)	28.61	26.78	24.33	22.06
Mean recall (%)	41.75	38.59	36.17	33.34

We adopt the same settings for d_{LCH} and r_{LCH} as in [8], such that the distance between two tags t_1 and t_2 is: $d_{LCH}(t_1, t_2) = \exp(-r_{LCH}(t_1, t_2) + 3.584) - 1$. In addition, $d_{tot}(I)$ evaluates the pairwise semantic distance among all candidate tags and is defined as: $d_{tot}(I) = \sum_{x \in U(I)} \sum_{y \in U(I)} d_{LCH}(x, y)$. By having all the three annotation factors on hand, we can compute the overall score for a candidate tag as

$$A(t|I) = a_1 f(t|I) + \frac{a_2}{\ln C} \ln\left(\frac{C}{1 + F(t)}\right) + a_3 G(t|I) \tag{41}$$

where $a_1 + a_2 + a_3 = 1$ represents the degree of importance of the three factors. Then, a tag t is chosen for annotation only if its score is within the top ε percentile among the candidate tags. According to our experimental results, we set $a_1 = 0.5$, $a_2 = 0.2$, $a_3 = 0.3$, and $\varepsilon = 0.7$.

4.2.2 Results

We evaluate the performance of the proposed image annotation approach using a subset of LabelMe data set [30] which contains both class labels and annotations. First, we use the LabelMe Matlab toolbox[4] to obtain images online from 8 scene classes (4 indoor and 4 outdoor): "bedroom", "kitchen", "living room", "bathroom", "forest", "highway", "coast", and "mountain" and then randomly choose 200 images from each category. Overall, we have 1600 images in total. Each category is associated with 4–5 tags. This data set was randomly divided into two partitions: one for training and the other for testing. First, we performed the categorization step as described in the previous section. We compare our approach with three other well-defined approaches: the EP-based Gaussian mixture model (*EPGMM*), the combination of a structure-composition model and a Gaussian mixture model (we denote it as *SC-GM*) proposed in [8], and the traditional *K-Means* algorithm. The categorization result of 8 classes of scene images is illustrated in Table 3. Findings observed from the comparison are that (1) the *EPDMM* provides the best categorization performance among all approaches and (2) it verifies that images represented by vectors of proportions are modeled more appropriately by the Dirichlet rather than the Gaussian.

[4] http://labelme.csail.mit.edu/.

Table 5 Sample annotation results

				
Our labels:	wall, bathtub, door, cabinet	window, floor, bed	cabinet, stove, chair	sofa, cushion, wall
LabelMe labels:	faucet, mirror, bathtub, cabinet, washbasin	bed, cushion, pillow, desk lamp, window	stove, worktop, ceiling lamp, window, cabinet	painting, sofa, chair, window, cushion
				
Our labels:	mountain, sea water, rock	tree, mountain, cloud	sky, car, road, tree	cloud, tree, mountain
LabelMe labels:	sand, cloud, sea water, sky, rock	tree, tree trunk, bush, sky	car, road, sign, truck	mountain, sky, tree, field, grass

Then, annotation was performed based on results obtained from the categorization stage. The performance of our automatic annotation system was evaluated by precision and recall which are defined in the standard way as follows: the annotation precision for a keyword is defined as the number of tags correctly predicted, divided by the total number of predicted tags. The annotation recall is defined as the number of tags correctly predicted, divided by the number of tags in the ground-truth annotation. In our experiments, the average number of tags generated for each test image is 4.12. Table 4 illustrates the average annotation precision and recall results over all the testing images according to the categorization results using different methods. According to this table, we can observe that the best annotation performance is achieved by using the categorization result through *EPDMM*. It confirms that the choice of categorization techniques has an significant impact on our annotation performance. Some examples of the annotation results obtained from *EPDMM* categorization method are displayed in Table 5.

4.3 Human Action videos Categorization

4.3.1 Experimental design

Categorizing multimedia data such as videos is a critical and challenging research topic [9,15,16]. With thousands of videos readily available, grouping them according to their contents is highly important for a variety of visual tasks such as event analysis, video indexing, browsing and retrieval, and digital libraries organization [33]. Videos categorization remains, however, an extremely challenging task due to several typical scenarios such as unconstrained motions, cluttered scenes, moving backgrounds, variations of illumination conditions and

Fig. 3 Examples of frames of different human actions from video sequences in the UCF sports dataset action dataset

viewpoints. In this section, we focus on applying the proposed *EPDMM* approach for categorizing human action videos. A similar methodology was adopted in this application as for the image categorization step in our previous application of automatic image annotation. The major difference is that in this application, instead of using PCA-SIFT features, we employ the space-time interest point detector proposed in [16] to extract local spatio-temporal features from each video sequence. After that, K-Means algorithm was applied on the obtained spatio-temporal features to construct a visual vocabulary with a size of 1,200. Then, the pLSA model was adopted to represent each video sequence by a 55-dimensional proportional vector. Lastly, we employ the *EPDMM* as a classifier to categorize videos by assigning the video sequence to the group which has the highest posterior probability according to Bayes' decision rule.

4.3.2 Evaluation on UCF Datasets

First, the experiments were conducted on two very challenging and popular datasets, namely the UCF sports [29] action and the UCF11 datasets [19].[5] The UCF sports dataset is collected by the UCF group from various sports featured on broadcast television channels such as the BBC and ESPN. It consists of over 200 video sequences at a resolution of 720 $\times$ 480 with nine actions, such as: "diving", "golf swinging" (g_swinging), "kicking", "lifting", "horseback riding" (riding), "running", "skating", "swinging", and "walking". Some examples of frames from each action class are displayed in Fig. 3. The UCF11 dataset contains 1168 video sequences in total with 11 action categories: "cycling", "diving", "golf swinging" (g_swinging), "soccer juggling" (s_juggling), "trampoline jumping" (t_jumping), "horse-back riding" (h_riding), "basketball shooting" (b_shooting), "volleyball spiking" (v_spiking), "swinging", "tennis swinging" (t_swinging), and "walking with a dog" (walking). Sample frames from each action class are shown in Fig. 4. For the UCF sports dataset, we used 70% of the video sequences to construct the visual vocabulary. In the case of the UCF11 dataset, we used half of the data set to construct the visual vocabulary.

[5] Datasets are available at: http://vision.eecs.ucf.edu/datasetsActions.html.

Fig. 4 Examples of frames of different human actions from video sequences in the UCF11 dataset

Table 6 The average classification accuracy and the number of components ($\hat{M}$) obtained using different algorithms over 20 runs

Dataset	*EPDMM*		*EPGMM*	
	$\hat{M}$	Acc. (%)	$\hat{M}$	Acc. (%)
UCF sports	8.13 (0.41)	78.52 (1.61)	7.45 (0.72)	73.08 (1.93)
UCF11	10.04 (0.58)	72.02 (1.37)	9.29 (0.83)	65.17 (1.82)

The numbers in parenthesis are the standard deviations

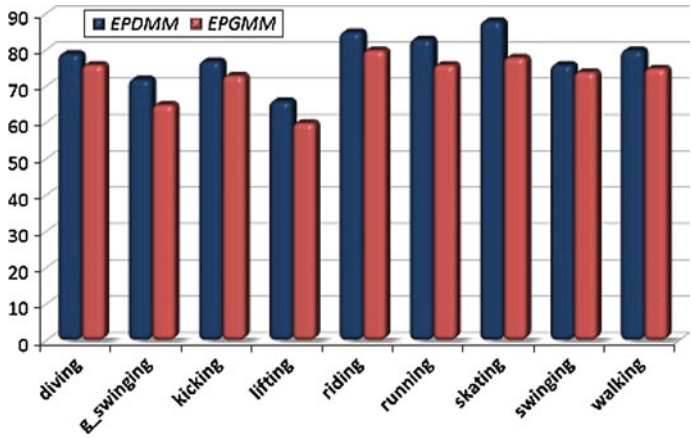

Fig. 5 Performance comparison in terms of the classification accuracy between *EPDMM* and *EPGMM* for the UCF sports dataset

The results that we will discuss in the following are obtained over 20 runs. Table 6 shows the average number of clusters and the average categorization accuracies using both Dirichlet and Gaussian mixture (*EPGMM*) models learned by running their respective EP learning algorithms 20 times. Figures 5 and 6 show the performance comparison of categorization accuracy among all action categories for the UCF sports and UCF11 datasets, respectively. According to the obtained results we can clearly see that the *EPDMM* outperforms the

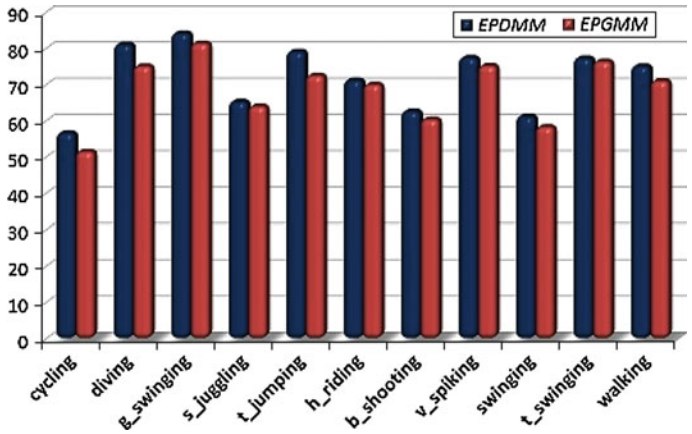

Fig. 6 Performance comparison in terms of the classification accuracy between *EPDMM* and *EPGMM* for the UCF11 dataset

Table 7 The average classification accuracy rate (%) with the corresponding standard deviations when considering different datasets using different methods

Method	Dataset	
	UCF sports	UCF11
EPDMM	78.52 % (1.61)	72.02 % (1.37)
EPGMM	73.08 % (1.93)	65.17 % (1.82)
SVM	76.75 % (1.56)	70.92 % (1.41)
Naive Bayes	73.91 % (1.77)	66.05 % (1.69)
K-NN	72.49 % (1.82)	63.54 % (1.58)

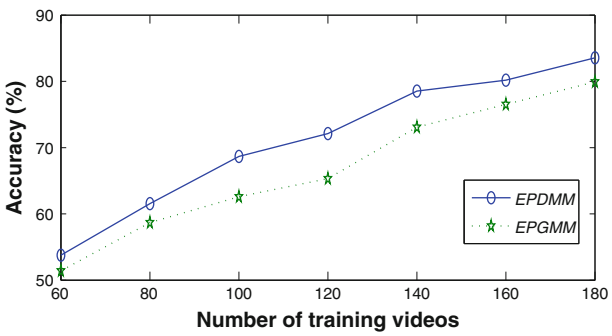

Fig. 7 Classification accuracy as a function of the number of videos used to construct the visual vocabulary in the case of the UCF sports dataset

EPGMM in terms of both categorization accuracy and selection of the optimal number of video categories. Given the difficulty of the datasets, these results are rather encouraging. The fact that the proposed *EPDMM* performs better than the *EPGMM* is actually expected, since in our case videos are represented by vectors of proportions and Dirichlet mixture models provide better modeling capabilities than Gaussian mixtures in this case. Furthermore, we

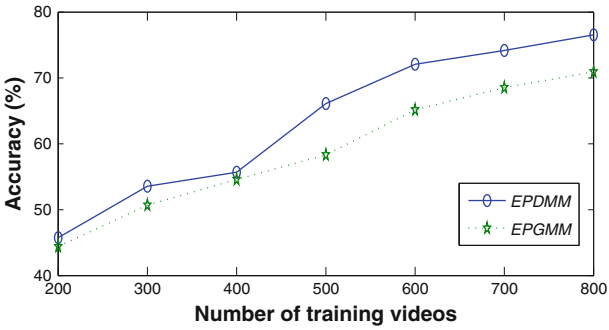

Fig. 8 Classification accuracy as a function of the number of videos used to construct the visual vocabulary in the case of the UCF11 dataset

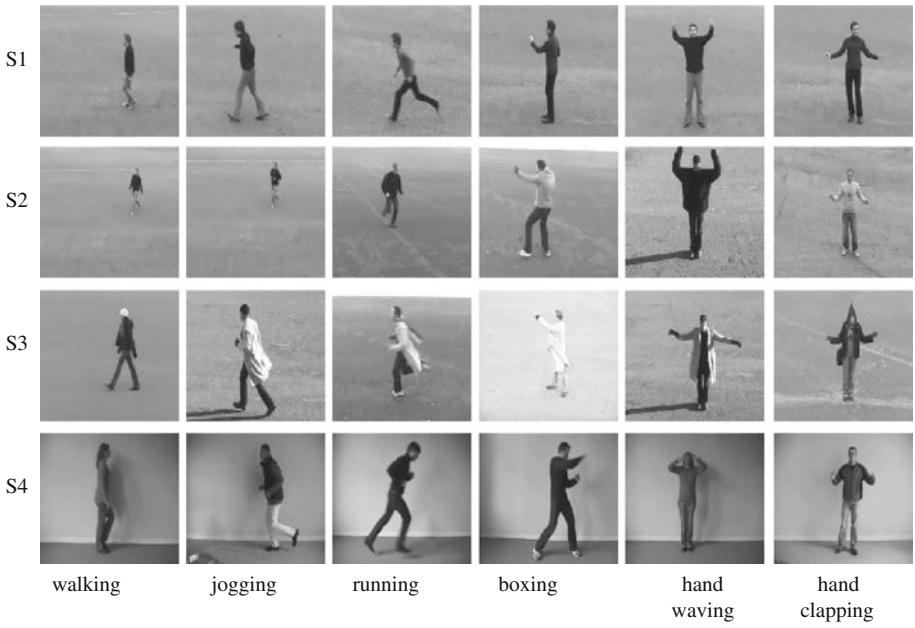

Fig. 9 Examples of frames, representing different human actions in different scenarios, from video sequences in the KTH dataset

have also tested the performance of categorizing human action videos for the UCF sports and UCF11 datasets using three other well-known classifiers: support vector machine (SVM), k-nearest neighbor (K-NN) and Naive Bayes. The corresponding test results are shown in Table 7. Obviously, the proposed *EPDMM* provides the best performance among all the tested methods for both datasets. Figures 7 and 8 illustrate the accuracy of the classification, when using *EPDMM* and *EPGMM* approaches, as a function of the number of videos used to construct the visual vocabulary. It is clear that the accuracy increases as the number of videos used to construct the visual vocabulary increases.

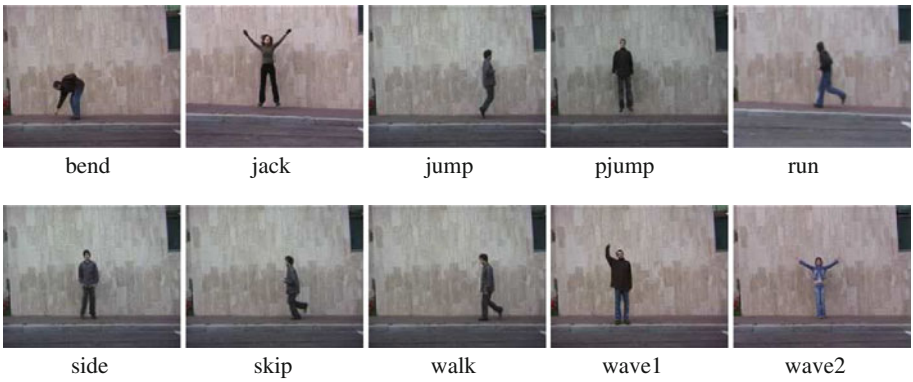

Fig. 10 Examples of frames of different human actions from video sequences in the Weizmann human action dataset

Table 8 The average classification accuracy rate for the KTH and Weizmann datasets using different methods

Method	Dataset	
	KTH	Weizmann
EPDMM	77.83 % (0.93)	85.91 % (1.13)
EPGMM	74.17 % (0.89)	81.72 % (1.08)
SVM	76.92 % (1.02)	84.85 % (0.94)
Naive Bayes	74.23 % (1.15)	83.08 % (1.21)
K-NN	73.64 % (0.97)	80.26 % (1.29)

4.3.3 Evaluation Using Other Datasets

In this subsection, we have evaluated the performance of our approach for categorizing human action videos on two older but classic datasets: the KTH dataset [31] and the Weizmann dataset [1]. The KTH human action dataset is one of the largest available video sequences datasets of human actions. It contains 2391 video sequences from six types of human actions: "walking", "jogging", "running", "boxing", "hand waving" and "hand clapping". Each action class is performed several times by 25 subjects in four different scenarios: outdoors (S1), outdoors with scale variation (S2), outdoors with different clothes (S3) and indoors (S4). All video samples were downsampled to the spatial resolution of 160×120 pixels and have a length of four seconds in average. Examples of frames from video sequences of each category are shown in Fig. 9. For this dataset, we construct the visual vocabulary from video sequences related to 16 subjects and evaluate the performance on the sequences of the remaining 9 subjects. The Weizmann dataset consists of 90 video sequences with ten different types of human actions that were performed by 9 subjects. These action categories include: "run", "walk", "skip", "jumping-jack" (or shortly "jack"), "jump-forward-on-two-legs" (or "jump"), "jump-in-place-on-two-legs" (or "pjump"), "gallop-sideways" (or "side"), "wave-two-hands" (or "wave2"), "wave-one-hand" (or "wave1"), and "bend". Some example frames of each action class can be viewed in Fig. 10. Since this dataset is small and has only has 90 sequences, we adopt a common scheme which extends the dataset by adding a horizontally flipped version of each video sequence to the original dataset. A leave-one-out setup is adopted for this dataset to test the performance of our categorization approach. That is, we construct our

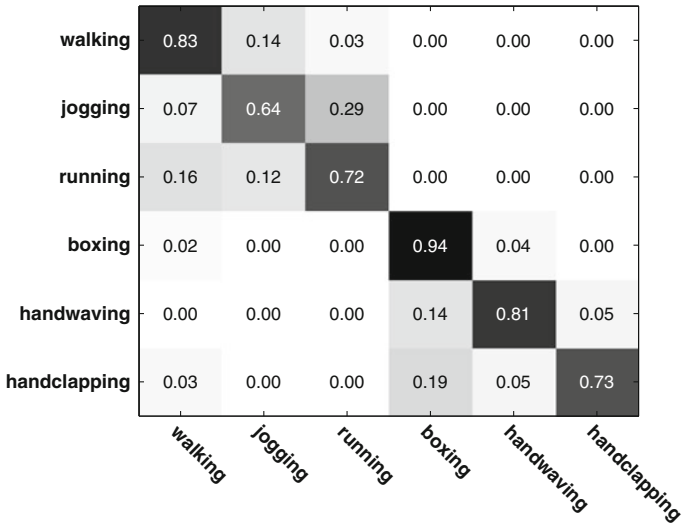

Fig. 11 The confusion matrix obtained by *EPDMM* for the KTH dataset

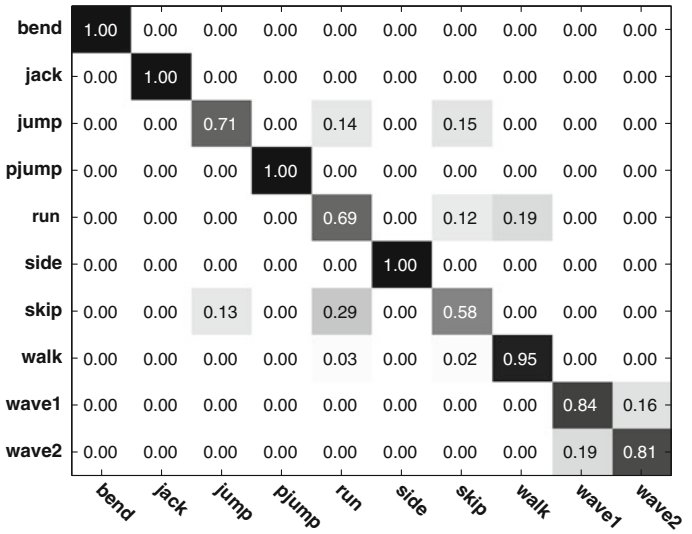

Fig. 12 The confusion matrix obtained by *EPDMM* for the Weizmann dataset

visual vocabulary from the video sequences of eight subjects (original + the flipped versions), and test the efficiency on the sequences of the remaining subject (only original ones). The categorization results were obtained over 20 runs.

We have applied *EPGMM*, SVM, Naive Bayes and K-NN on these two datasets for comparison and shown the corresponding results in Table 8. As we can see in this table, it is clear that our algorithm outperforms the other algorithms for categorizing human action videos for these two datasets. Furthermore, the confusion matrices calculated using *EPDMM* for the KTH and Weizmann datasets are shown in Figs. 11 and 12, respectively. We may notice that, for the KTH dataset, most of the confusion occurs between similar actions such as "walking"

and “jogging”, “jogging” and “running”, “hand clapping” and “boxing”. For the Weizmann dataset, most errors are also generated from similar action categorizes, such as “run” with “walk”, “jump” with “skip”, and “skip” with “jump” and “run”.

5 Conclusion

In this paper, we have proposed an EP framework for learning finite Dirichlet mixture models. Within this framework, all model’s parameters and the number of clusters can be determined simultaneously, which allows to avoid under- or over-fitting. Extensive experiments have been conducted and have involved synthetic data and real-world challenging applications namely automatic image annotation and human action videos categorization using the pLSA model and the bag of visual words representation. Possible future works can be devoted to integrate feature selection within the proposed framework, to extend the learning approach proposed in this paper to online settings or to the extension of the proposed model to the infinite case using Dirichlet processes.

Acknowledgments The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank the anonymous referees and the associate editor for their comments. The complete source code of this work is available upon request.

References

1. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as Space-Time Shapes. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp 1395–1402.
2. Bouguila N, Ziou D (2005) Using unsupervised learning of a finite dirichlet mixture model to improve pattern recognition applications. Pattern Recognit Lett 26(12):1916–1925
3. Bouguila N, Ziou D (2006a) Online clustering via finite mixtures of dirichlet and minimum message length. Eng Appl Artif Intell 19(4):371–379
4. Bouguila N, Ziou D (2006b) Unsupervised selection of a finite dirichlet mixture model: an MML-based approach. IEEE Trans Knowl Data Eng 18(8):993–1009
5. 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
6. Chang E (2003) CBSA: content-based soft annotation for multinomial image retrieval using bayes point machines. IEEE Trans Circuit Syst Video Technol 13(1):26–38
7. Chang S, Dasgupta N, Carin L (2005) A Bayesian approach to unsupervised feature selection and density estimation using expectation propagation. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), pp 1043–1050
8. Datta R, Ge W, Li J, Wang JZ (2006) Toward bridging the annotation-retrieval gap in image search by a generative modeling approach. In: Proceedings of the 14th annual ACM international conference on multimedia (MM), ACM, pp 977–986
9. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal feature. In: Proceedings of the IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance (VS-PETS), pp 65–72
10. Fan J, Gao Y, Luo H, Xu G (2005) Statistical modeling and conceptualization of natural images. Pattern Recognit 38:865–885
11. Figueiredo M, Jain A (2002) Unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell 24(3):381–396
12. Heskes T, Zoeter O (2002) Expectation propagation for approximate inference in dynamic Bayesian networks. In: Proceedings of the conference on uncertainty in artificial intelligence (UAI), pp 216–223
13. Hofmann T (2001) Unsupervised learning by probabilistic latent semantic analysis. Mach Learn 42(1/2):177–196
14. Ke Y, Sukthankar R (2004) PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 506–513

15. Laptev I (2005) On space-time interest points. Int J Comput Vis 64(2/3):107–123
16. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
17. Law MHC, Figueiredo MAT, Jain AK (2004) Simultaneous feature selection and clustering using mixture models. IEEE Trans Pattern Anal Mach Intell 26(9):1154–1166
18. Leacock C, Chodorow M (1998) In: Fellbaum C (Ed) WordNet: an electronic lexical database. MIT Press, pp 305–332
19. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos "In The Wild". In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), pp 1996–2003
20. Luo J, Savakis AE, Singhal A (2005) A Bayesian network-based framework for semantic image understanding. Pattern Recognit 38:919–934
21. Ma Z, Leijon A (2010) Expectation propagation for estimating the parameters of the beta distribution. In: Proceedings of IEEE international conference on acoustics speech and signal processing (ICASSP), pp 2082–2085
22. Maybeck PS (1982) Stochastic models, estimation and control. Academic Press
23. McLachlan G, Peel D (2000) Finite mixture models. Wiley, New York
24. Mikolajczyk K, Schmid C (2004) Scale and affine invariant interest point detectors. Int J Comput Vis 60:63–86
25. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38:39–41
26. Minka T (2001) Expectation propagation for approximate Bayesian inference. In: Proceedings of the conference on uncertainty in artificial intelligence (UAI), pp 362–369
27. 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
28. Naphade MR, Huang TS (2001) A probabilistic framework for semantic video indexing, filtering, and retrieval. IEEE Trans Multimed 3:141–151
29. Rodriguez M, Ahmed J, Shah M (2008) Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
30. Russell B, Torralba A, Murphy K, Freeman W (2008) LabelMe: a database and Web-based tool for image annotation. Int J Comput Vis 77:157–173
31. Schüldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: Proceedings of the 17th international conference on pattern recognition (ICPR), pp 32–36
32. Zhao R, Grosky WI (2000) From features to semantics: some preliminary results. In: Proceedings of the IEEE international conference on multimedia and expo (ICME). IEEE Computer Society, pp 679–682
33. Zhong D, Zhang H, Chang SF (1997) Clustering methods for video browsing and annotation. In: Proceedings of the SPIE conference on storage and retrieval for video and image databases, pp 239–246