Unsupervised Feature Selection for Proportional Data Clustering via Expectation Propagation

Wentao Fan and Nizar Bouguila

Abstract—In this paper, an expectation propagation (EP) inference framework for unsupervised feature selection is proposed for modeling proportional data which naturally appear in many applications such as text and image modeling, in the context of finite mixture-based clustering. Within our framework, simultaneous clustering and feature selection is formalized using finite mixtures of generalizing Dirichlet (GD) distributions. The proposed EP-based inference framework allows to obtain a full posterior distribution on all our unsupervised feature selection model's parameters. Moreover, the complexity of the deployed mixture models and all the involved model parameters can be evaluated simultaneously. The effectiveness and efficiency of the proposed algorithm are evaluated on both synthetic data and two challenging applications namely human action videos categorization and facial expression recognition.

I. INTRODUCTION

In recent years, finite mixture models have been increasingly applied in unsupervised learning problems, such as clustering, for both univariate and multivariate data [1]. 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. The most common choice for the basic distribution is the Gaussian. Although Gaussian mixtures have been widely employed due to their approximation properties, when the data are not Gaussian, other mixture models such as the generalized Dirichlet (GD) mixture have been shown to provide a better alternative to clustering and data modeling in general, and especially for modeling proportional data (e.g. normalized histograms) [2], [3]. Thus, GD mixture is adopted in this paper. Commonly, all the available features in an observed data set are exploited by the clustering algorithm for learning a model. However, in practice, some features can be noisy, and thus do not contribute to the clustering process [4]. These irrelevant features in a data set may compromise the clustering performance. One solution to tackle this problem is to consider only the most relevant features which is known as feature selection. In unsupervised learning, however, the problem of feature selection becomes much more challenging since inference has to be made on both the selected features and the clustering structure [5], [4], [6], [7], [8], [9]. An early influential work advocating the use of finite mixture models for unsupervised feature selection has been presented in [4]. The main idea is to suppose that irrelevant features are generated from an univariate distribution which is common to all clusters (i.e. independent from class labels). This unsupervised feature selection scheme, based on the Gaussian mixture, has been extended in [7] by considering GD mixture and by assuming that rather than a single univariate distribution, irrelevant features may be distributed according to a mixture of overlapped Beta distributions that are common to all clusters. The learning approach developed in [7] has been based on the expectation-maximization (EM) algorithm to optimize a minimum message length (MML) objective. This approach, like most deterministic learning techniques, provides a point estimate of the model's parameters and cannot address in a single formulation both model selection and parameter estimation [10], [11]. Indeed, several works have shown that pure Bayesian or approximate Bayesian formulations, such as expectation propagation (EP), provide generally better results [10]. 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 [12], [13], [14]. It is an extension to assumed-density filtering (ADF) [15] 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 is improved by re-using the data points many times. Moreover, compared to MCMC and Gibbs sampling techniques, the major advantage of EP is that it is much more computationally efficient. The major contribution of this paper is that we construct a statistical Bayesian framework based on finite GD mixture models using EP inference, such that the model complexity and the model parameters can be estimated simultaneously in a single optimization framework. Furthermore, we apply the proposed approach to solve two challenging problems involving human action videos categorization and facial expression recognition. We are mainly motivated by the good results obtained recently using EP techniques in machine learning applications in general [16], [17] and for the unsupervised feature selection problem in particular [10]. The rest of the paper is organized as follows. Section 2 presents the details of our unsupervised feature selection model. In Section 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, discussions and future directions.

Wentao Fan is with the Department of Electrical and Computer Engineering, Concordia University, QC, Canada, H3G 1T7. (email: wentao.fan@encs.concordia.ca). Nizar Bouguila is with the Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, H3G 1T7, Canada. (email:nizar.bouguila@concordia.ca).

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC).
II. FINITE GD MIXTURES WITH FEATURE SELECTION

A GD distribution of a $D$-dimensional random vector $\bar{Y}$ is defined as

$$GD(\bar{Y} | \vec{\alpha}_j, \vec{\beta}_j) = \prod_{i=1}^{D} \frac{\Gamma(\alpha_{j} + y_{ij})}{\Gamma(\alpha_{j})} \Gamma(y_{ij}, \beta_{j}) \left(1 - \sum_{k=1}^{L} Y_k\right)^{\gamma_{j}}$$

where $\sum_{i=1}^{D} Y_i < 1$ and $0 < Y_i < 1$ for $l = 1, \ldots, D$. $\vec{\alpha}_j = (\alpha_{j1}, \ldots, \alpha_{jD})$ and $\vec{\beta}_j = (\beta_{j1}, \ldots, \beta_{jD})$ are the parameters of the GD distribution, such that, $\alpha_{jl} > 0$, $\beta_{jl} > 0$, $\gamma_{jl} = \beta_{jl} - \alpha_{jl+1} - \beta_{jl+1}$ for $l = 1, \ldots, D - 1$, and $\gamma_{jD} = \beta_{jD} - 1$. Assume that we have a set of $N$ independent and identically distributed vectors $\bar{Y} = (Y_1, \ldots, Y_N)$, where each vector $Y_i = (Y_{i1}, \ldots, Y_{iD})$ is assumed to be sampled from a finite GD mixture model with $M$ components [18], [19], [2], [20], [21]:

$$p(\bar{Y}_i | \vec{\pi}, \vec{\alpha}, \vec{\beta}) = \sum_{j=1}^{M} \pi_j GD(\bar{Y}_i | \vec{\alpha}_j, \vec{\beta}_j)$$

where $\vec{\alpha} = (\vec{\alpha}_1, \ldots, \vec{\alpha}_M)$ and $\vec{\beta} = (\vec{\beta}_1, \ldots, \vec{\beta}_M)$. $\vec{\alpha}_j$ and $\vec{\beta}_j$ are the parameters of the GD distribution representing component $j$. $\vec{\pi} = (\pi_1, \ldots, \pi_M)$ represents the mixing coefficients with the constraints that are positive and sum to one. Based on the mathematical properties of the GD thoroughly discussed in [7], the finite GD mixture model is equivalent to the following mixture model 1

$$p(\bar{X}_i | \vec{\pi}, \vec{\alpha}, \vec{\beta}) = \sum_{j=1}^{M} \pi_j \prod_{l=1}^{D} Beta(X_{il} | \alpha_{jl}, \beta_{jl})$$

where $\bar{X}_i = (X_{i1}, \ldots, X_{iD})$, $X_{il} = Y_{il}$ and $X_{il} = Y_{il}/(1 - \sum_{a=1}^{l-1} Y_{ia})$ for $l > 1$, and $Beta(X_{il} | \alpha_{jl}, \beta_{jl})$ is a Beta distribution defined with parameters $(\alpha_{jl}, \beta_{jl})$.

It is noteworthy that in practice the features $\{X_{il}\}$ are generally not equally important for the clustering task since some features may be “noise” that do not contribute to clustering process. Thus, in our work, we adopt the unsupervised feature selection framework as proposed in [7] where irrelevant features are modeled as a mixture of Beta distributions independent from the class labels, such that

$$p(X_{il}) = \sum_{k=1}^{K} \eta_{kl} Beta(X_{il} | \lambda_{kl}, \tau_{kl})$$

where $\{\eta_{kl}\}$ are the mixing parameters which are positive and sum to one so that each $\eta_{kl}$ is the prior probability that $X_{il}$ is generated from the $k$th component of the finite Beta mixture representing irrelevant features. The motivation of using a Beta mixture to model irrelevant features is based on the fact that this specific mixture is flexible enough to approximate any univariate distribution including the uniform one [7]. Therefore, we can approximate our mixture model as

$$p(\bar{X}_i | \Theta) = \sum_{j=1}^{M} \pi_j \prod_{l=1}^{D} [\epsilon_{l1} Beta(X_{il} | \theta_{jl})$$

$$+ \epsilon_{l2} \sum_{k=1}^{K} \eta_{kl} Beta(X_{il} | \xi_{kl})$$

where $\Theta = \{\vec{\pi}, \vec{\alpha}, \vec{\beta}, \vec{\xi}\}$ represents the set of model parameters. $\theta_{jl} = (\alpha_{jl}, \beta_{jl})$ and $\xi_{kl} = (\lambda_{kl}, \tau_{kl})$ are the parameters of Beta distributions for the relevant and irrelevant features, respectively. The vector $\vec{\xi} = (\vec{\epsilon}_1, \ldots, \vec{\epsilon}_D)$ denotes the features saliencies (i.e. the probabilities that the features are relevant) with $\vec{\epsilon}_l = (\epsilon_{l1}, \epsilon_{l2})$ and $\epsilon_{l1} + \epsilon_{l2} = 1$.

III. EP-BASED LEARNING FOR GD MIXTURE WITH FEATURE SELECTION

In this section, EP framework is adopted for learning the GD mixture model with feature selection. In Bayesian modeling, we assign to each unknown parameter a prior distribution. Due to the computational efficiency and tractability, conjugate priors are adopted in our model. In our case, for parameters $\vec{\pi}, \vec{\alpha}$ and $\vec{\beta}$, Dirichlet distributions are adopted as conjugate priors, such that

$$p(\vec{\pi}) = Dir(\vec{\pi} | \vec{\theta}), \quad p(\vec{\alpha}) = \prod_{l=1}^{D} Dir(\vec{\alpha}_{il} | \vec{\beta}_{jl}), \quad p(\vec{\beta}) = \prod_{l=1}^{D} Dir(\vec{\beta}_{il} | \vec{\xi}_{kl})$$

For the parameters $\theta_{jl}$ and $\xi_{kl}$ of the Beta mixtures, we adopt Gaussian assumption which has been shown to provide good results in the case of the Beta [17]. This is motivated by the fact that the Gaussian allows analytically tractable calculation and is able to take into account the correlation between the Beta distribution parameters. Thus, two-dimensional Gaussians are considered for $\theta_{jl}$ and $\xi_{kl}$, such that

$$p(\theta_{jl}) = N(\theta_{jl} | \bar{\theta}_{jl}, A_{jl}), \quad p(\xi_{kl}) = N(\xi_{kl} | \bar{\rho}_{kl}, B_{kl})$$

The central idea of the EP framework is that the posterior distribution of our model can be represented in product of factors in the form $p(\Theta | X) \propto \prod_{i} f_i(\Theta)$, and can then be approximated by:

$$q^*(\Theta) = \frac{\prod_{i} \tilde{f}_i(\Theta)}{\int \prod_{i} f_i(\Theta) d\Theta}$$

where each factor $\tilde{f}_i(\Theta)$ is an approximation corresponds to one of the factors $f_i(\Theta)$ in the true posterior. The first step in the EP inference is to initialize all the approximating factors $\tilde{f}_i(\Theta)$. This is done by initializing all the involved hyper-parameters $\{\alpha_j, b_k, c_a, \beta_{jl}, A_{jl}, \rho_{kl}, B_{kl}\}$. Next, we initialize the posterior approximation by setting $q^*(\Theta) \propto \prod_{i} f_i(\Theta)$. Thus, we can easily compute the hyperparameters of $q^*(\Theta)$
as
\[ a_j^* = \sum_i a_{i,j} - N, \quad b_k^* = \sum_i b_{i,k} - N \] (9)
\[ c_s^* = \sum_i c_{i,s} - N, \quad s \in \{1,2\} \] (10)
\[ \bar{\mu}_{ji}^* = \left( \sum_i A_{i,ji}^{-1} \right) \left( \sum_i A_{i,ji} \bar{\mu}_{i,ji} \right) \] (11)
\[ A_j^* = \sum_i A_{i,ji}, \quad B_{kl}^* = \sum_i B_{i,kl} \] (12)
\[ \bar{\rho}_{kl}^* = \left( \sum_i B_{i,kl}^{-1} \right) \left( \sum_i B_{i,kl} \bar{\rho}_{i,kl} \right) \] (13)

In order to update the factor \( f_i(\Theta) \), we have to remove it from the posterior \( q^i(\Theta) \) as: \( q^i(\Theta) = q^*(\Theta) / f_i(\Theta) \). Then, the corresponding hyperparameters can be computed analytically as
\[ a_j^{vi} = a_j^* - a_{i,j} + 1, \quad b_k^{vi} = b_k^* - b_{i,k} + 1 \] (14)
\[ c_s^{vi} = c_s^* - c_{i,s} + 1, \quad s \in \{1,2\} \] (15)
\[ \bar{\mu}_{ji}^{vi} = (A_j^{vi})^{-1} (A_j^{vi} \bar{\mu}_{ji} - A_{i,ji} \bar{\mu}_{i,ji}) \] (16)
\[ A_j^{vi} = A_j^* - A_{i,ji}, \quad B_{kl}^{vi} = B_{kl}^* - B_{i,kl} \] (17)
\[ \bar{\rho}_{kl}^{vi} = (B_{kl}^{vi})^{-1} (B_{kl}^{vi} \bar{\rho}_{kl} - B_{i,kl} \bar{\rho}_{i,kl}) \] (18)

Next, the updated posterior \( \bar{\rho}(\Theta) \) can be calculated as
\[ \bar{\rho}(\Theta) = \frac{1}{Z_i} f_i(\Theta) q^{vi}(\Theta) \] (19)

where we have
\[ Z_i = \sum_{j=1}^M \frac{a_{i,j}}{a_{i,j}} \prod_{l=1}^D \left[ \frac{c_{i,1} + c_{i,2}}{c_{i,1} + c_{i,2}} \right] \int \text{Beta}(X_{il} | \theta_{ji}) q^{vi}(\theta_{ji}) d\theta_{ji} + \frac{c_{i,1} + c_{i,2}}{c_{i,1} + c_{i,2}} \int \text{Beta}(X_{il} | \xi_{kl}) q^{vi}(\xi_{kl}) d\xi_{kl} \] (20)

Notice that, the integrations in (20) are 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 [17]. First, we can define a normalized distribution for the integrand \( \text{Beta}(X_{il} | \theta_{ji}) q^{vi}(\theta_{ji}) \) which is indeed a product of a Beta distribution and a Gaussian distribution as \( H(\theta_{ji}) = \frac{h(\theta_{ji})}{\int h(\theta_{ji}) d\theta_{ji}} \), where we have
\[ h(\theta_{ji}) = \text{Beta}(X_{il} | \theta_{ji}) N(\theta_{ji} | \bar{\mu}_{ji}^{vi}, A_j^{vi}) \] (21)

In the Laplace method the goal is to find a Gaussian approximation which is centered on the mode of the distribution \( H(\theta_{ji}) \). We could obtain the mode \( \theta_{ji}^* \) numerically by setting the first derivative of \( \ln h(\theta_{ji}) \) to 0. Then, we can approximate \( h(\theta_{ji}) \) via the mode as
\[ h(\theta_{ji}) \approx h(\theta_{ji}^*) \exp \left( -\frac{1}{2} (\theta_{ji} - \theta_{ji}^*) \bar{A}_{ji} (\theta_{ji} - \theta_{ji}^*) \right) \] (22)

where \( \bar{A}_{ji} = -\frac{\partial^2 \ln h(\theta_{ji})}{\partial \theta_{ji}^2} |_{\theta_{ji}^*} \). Thus, the integration of \( h(\theta_{ji}) \) can be approximated by using (22) as
\[ \int h(\theta_{ji}) d\theta_{ji} \approx h(\theta_{ji}^*) \frac{2\pi}{|\bar{A}_{ji}|^{1/2}} \] (23)

We can apply the same Laplace approximation approach to obtain an approximation of the integrand \( \text{Beta}(X_{il} | \xi_{kl}) q^{vi}(\xi_{kl}) \) in (20) with the corresponding mode \( \xi_{kl}^* \). Hence, we can rewrite (20) as following:
\[ Z_i = \sum_{j=1}^M \frac{a_{i,j}}{a_{i,j}} \prod_{l=1}^D \left[ \frac{c_{i,1} + c_{i,2}}{c_{i,1} + c_{i,2}} \right] \int \text{Beta}(X_{il} | \theta_{ji}) q^{vi}(\theta_{ji}) d\theta_{ji} + \frac{c_{i,1} + c_{i,2}}{c_{i,1} + c_{i,2}} \int \text{Beta}(X_{il} | \xi_{kl}) q^{vi}(\xi_{kl}) d\xi_{kl} \] (24)

where \( \bar{B}_{kl} \) is calculated in a similar way as for \( \bar{A}_{ji} \). Then, we can revise the posterior \( q^*(\Theta) \) by matching its sufficient statistics to the corresponding moments of \( \bar{\rho}(\Theta) \). This is done by calculating the partial derivative of \( \ln Z_i \) with respect to the model hyperparameters. For \( a_j^{vi} \), we can get
\[ \nabla a_j^{vi} \ln Z_i = E_{\bar{\rho}}[\ln \pi_j] + \Psi \left( \sum_{j=1}^M a_j^{vi} \right) - \Psi \left( a_j^{vi} \right) \] (25)

where \( \Psi(\cdot) \) is the digamma function. By applying moment matching, we obtain
\[ E_{\bar{\rho}}[\ln \pi_j] = E_{q^v}[\ln \pi_j] = \Psi(a_j^*) - \Psi \left( \sum_{j=1}^M a_j^* \right) \] (26)

The right hand side of (25) can be computed analytically by using (24). Similarly, we can compute the partial derivatives of \( \ln Z_i \) with respect to the other model hyperparameters. After obtaining \( q^*(\Theta) \) and \( q^v(\Theta) \), we can update the revised hyperparameters for the approximating factor \( f_i \) as
\[ a_{i,j} = a_j^* - a_{i,j} + 1, \quad b_{i,k} = b_k^* - b_{i,k} + 1 \] (27)
\[ c_{i,s} = c_s^* - c_{i,s} + 1, \quad s \in \{1,2\} \] (28)
\[ \bar{\mu}_{i,ji} = A_{i,ji}^{-1} (A_{i,ji} \bar{\mu}_{j,ji} - A_{i,ji} \bar{\mu}_{i,ji}) \quad A_{i,ji} = A_{i,ji}^* \] (29)
\[ \bar{\rho}_{i,kl} = B_{i,kl}^{-1} (B_{i,kl} \bar{\rho}_{j,kl} - B_{i,kl} \bar{\rho}_{i,kl}) \quad B_{i,kl} = B_{i,kl}^* \] (30)

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 and the features saliencies in the posterior distributions as
\[ E[\pi_j] = \frac{a_j^*}{\sum_j a_j^*}, \quad E[\eta_k] = \frac{b_k^*}{\sum_k b_k^*}, \quad E[\xi_{kl}] = \frac{c_{i,1} + c_{i,2}}{c_{i,1} + c_{i,2}} \] (31)

The complete learning algorithm is summarized in Algorithm 1.
Algorithm 1

1: Choose the initial number of components $M$ and $K$.
2: Initialize the approximating factors $\tilde{f}_i(\Theta)$ by initializing all the involved hyperparameters $\{a_i, b_i, c_i, \hat{\nu}_i, \hat{A}_i, \hat{\rho}_{kl}, B_{kl}\}$.
3: Initialize the posterior approximation by setting $q^*(\Theta) \propto \prod_i \tilde{f}_i(\Theta)$. The hyperparameters of of $q^*(\Theta)$ are calculated by (9)–(13).
4: repeat
5: Choose a factor $\tilde{f}_i(\Theta)$ to refine.
6: Remove $\tilde{f}_i(\Theta)$ from the posterior $q^*(\Theta)$ by division $q^i(\Theta) = q^*(\Theta)/\tilde{f}_i(\Theta)$.
7: Evaluate the new posterior by setting the sufficient statistics (moments) of $q^*(\Theta)$ to the corresponding moments of $\tilde{P}(\Theta)$.
8: Update the factor $\tilde{f}_i(\Theta)$ by updating the corresponding hyperparameters as in (27)–(30).
9: until Convergence criterion is reached.
10: Compute the estimated values of the mixing coefficients and the features saliences as in (31).
11: Detect the optimal number of components $M$ and $K$ by eliminating the components with small mixing coefficients close to 0.

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the effectiveness of the proposed EP framework for learning GD mixture model with feature selection algorithm ($\text{EPGDFS}$) through synthetic data and two challenging applications namely human action videos categorization and facial expression recognition. It is noteworthy that comparing all the approaches that have been proposed for both applications is clearly out of the scope of this work. Indeed, our main goal is to validate the proposed learning algorithm and compare it to similar approaches. In all our experiments, we initialize the number of components $M$ and $K$ to 15 and 10, respectively. For experiments involved in this paper, we have noticed that poor initialization values of the hyperparameters considerably slow down the convergence speed. Based on our experiments, an optimal choice of the initial values of the hyperparameters for each factor $f_i(\Theta)$ can be as following: $a_{i,j}$, $b_{i,k}$, and $c_{i,s}$ are set to 1, $\hat{\nu}_{i,j}$ and $\hat{\nu}_{i,k}$ are initialized to 0.5, $A_{i,j}$ and $B_{i,k}$ are set to 0.01. Our simulations have supported these specific choices.

A. Synthetic data

The goal of this experiment is to evaluate the performance of the proposed EP mixture learning algorithm in terms of estimation and selection, through quantitative analysis on four ten-dimensional (two relevant features and eight irrelevant features) synthetic data sets. The relevant features were generated in the transformed space from mixtures of Beta distributions with well-separated components, while irrelevant ones were from mixtures of overlapped components. We run our algorithm ten times to evaluate its performance. The real and estimated parameters of the distributions representing the relevant features for each data set using the proposed EP algorithm are shown in Table I. Based on this table, both the parameters and the mixing coefficients of the mixture model representing relevant features are accurately estimated by our EP algorithm. We have also obtained accurate estimate values for the parameters of the mixture models representing irrelevant features (the eight remaining features) by adopting the proposed algorithm. The estimated feature saliences of all the 10 features for each synthetic data set are illustrated in Fig. 1. It obviously shows that
for all the synthetic data.


EPGDFS

Here, 5.06 provide comparable results in estimating the 8.62.

AND

A set of 3, (d)

EMGDFS

is almost 13.21.

REQUIRED BEFORE CONVERGENCE.

EMGDFS, we

EPGDFS

I.

D
e

one standard 14.32.

n
EMGDFS

3.11 (D
e

AVERAGE RUNTIME (IN SECONDS) REQUIRED BEFORE CONVERGENCE FOR EPGDFS AND EMGDFS.

<table>
<thead>
<tr>
<th>Data set</th>
<th>EPGDFS</th>
<th>EMGDFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.11</td>
<td>8.62</td>
</tr>
<tr>
<td>2</td>
<td>3.85</td>
<td>9.13</td>
</tr>
<tr>
<td>3</td>
<td>4.68</td>
<td>13.21</td>
</tr>
<tr>
<td>4</td>
<td>5.06</td>
<td>14.32</td>
</tr>
</tbody>
</table>

B. Human Action Videos Categorization

With thousands of videos on hand, grouping them according to their contents is highly important for a variety of visual tasks such as event analysis [22], video indexing, browsing and retrieval, and digital libraries organization [23]. Recently, videos categorization has attracted many research efforts (see, for instance, [24], [25], [26]). However, it remains an extremely challenging task due to several typical scenarios such as cluttered scenes, moving backgrounds, geometric changes and deformation of objects and variations of illumination conditions and viewpoints. In this section, we present our results using the proposed EPGDFS algorithm and the bag-of-visual words representation, for categorizing human action videos.

The methodology that we have adopted for unsupervised videos categorization can be summarized as follows. First, local spatio-temporal features from each video sequence are extracted from their detected space-time interest points. Here, we employ the space-time interest point detector proposed in [26]. Next, K-means algorithm is applied to construct a visual vocabulary by quantizing these spatio-temporal features into visual words and each video is then represented as a frequency histogram over the visual words. It is known that probabilistic topic models, such as the Probabilistic Latent Semantic Analysis (pLSA) model [27], find a low dimensional representation of data under the assumption that each data instance can reveal multiple components or “topics”. Thus, in the following step, we apply the pLSA model to the obtained histograms, such that each video is represented now by a D-dimensional proportional vector where D is the number of latent aspects. Finally, we employ the EPGDFS to cluster videos by assigning each video sequence to the group which has the highest posterior probability according to Bayes’ decision rule.

In this experiment, we consider the Weizmann human action data set [28]. It contains 90 video sequences at a resolution of 180×144 pixels. Ten different types of human actions are performed by nine subjects. Some examples of frames from each action class are displayed in Fig. 2. A leave-one-out setup is adopted to test the performance of our categorization approach. More specifically, we construct our visual vocabulary from the video sequences of eight subjects, by setting the number of clusters to 800 in the K-Means algorithm, and test the efficiency on the sequences of the remaining subject. The results that we shall discuss in the following are obtained over twenty runs. Figure 3 illustrates the confusion matrix for the Weizmann data set using the EPGDFS. The overall accuracy is around 87.25%.

As we can see, most errors are generated from similar action categorizes, such as “run” with “walk”, “jump” with “skip”, “skip” with “jump” and “run”.

TABLE II

Average runtime (in seconds) required before convergence for EPGDFS and EMGDFS.

<table>
<thead>
<tr>
<th>Data set</th>
<th>EPGDFS</th>
<th>EMGDFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.11</td>
<td>8.62</td>
</tr>
<tr>
<td>2</td>
<td>3.85</td>
<td>9.13</td>
</tr>
<tr>
<td>3</td>
<td>4.68</td>
<td>13.21</td>
</tr>
<tr>
<td>4</td>
<td>5.06</td>
<td>14.32</td>
</tr>
</tbody>
</table>

Fig. 1. Features saliencies for the synthetic data sets with one standard deviation over ten runs. (a) Data set 1, (b) Data set 2, (c) Data set 3, (d) Data set 4.

Fig. 2. Sample frames from the Weizmann human action data set.
proposed in [10] and the model with feature selection via EP inference (EPGDFS) feature saliencies of the 50-dimensional aspects obtained when the number of aspects is set to 50. The corresponding performance for the Weizmann data set using three other well-developed algorithms while detecting the number of categories is illustrated in Fig. 5. As shown in this figure, the highest accuracy can be obtained when the number of aspects is set to 50. The corresponding feature saliencies of the 50-dimensional aspects obtained by EPGDFS are illustrated in Fig. 5. As shown in this figure, the

Moreover, we have applied three other well-developed approaches with the same experimental settings for comparison: the EP framework for learning GD mixture model without feature selection (EPGD), the Gaussian mixture model with feature selection via EP inference (EPGauFS) proposed in [10] and the EMGDFS. The average classification accuracy, the average number of components and the average runtime obtained by each algorithm over twenty runs are shown in Table III. According to this table, our algorithm provides higher classification accuracy than the other algorithms while detecting the number of categories more accurately. Additionally, the number of components for the mixture model representing irrelevant features is estimated as 2. We have also tested the classification performance for the Weizmann data set using three other well-known classifiers: SVM (85.01%), k-NN (80.83%) and naive Bayes (81.52%). Obviously, our algorithm outperforms all of them. Furthermore, the choice of the number of aspects also influences the accuracy of detection as shown in Fig. 4. Based on this figure, the highest accuracy can be obtained when the number of aspects is set to 50. The corresponding feature saliencies of the 50-dimensional aspects obtained by EPGDFS are illustrated in Fig. 5. As shown in this figure, the features have different relevancy degrees and then contribute differently to the clustering task. For instance, there are six features (features number 7, 26, 34, 36, 43 and 49) that have saliencies lower than 0.5, and then provide less contribution in clustering. By contrast, seven features (features number 2, 9, 13, 20, 31, 45 and 50) have high relevancy degrees with feature saliencies greater than 0.9.

C. Facial Expression Recognition

Facial expression recognition is a type of visual learning process which deals with the classification of facial motions and has been applied in various fields such as image understanding, psychological studies, facial nerve grading in medicine, synthetic face animation and virtual reality [29], [30], [31], [32]. In this experiment, we address the problem of facial expression recognition using the proposed EPGDFS algorithm and Local Binary Pattern (LBP) features [33]. LBP features have been originally developed for texture analysis, and recently have shown promising results in facial image analysis [34], [35], [36]. Compared to other proposed facial expression features (such as Gabor features [30]), LBP features are more robust against illumination changes and are more computationally efficient [34]. The data set that we have used for evaluation in this experiment is the well-known Japanese Female Facial Expression (JAFFE) data set [30]. It includes 213 images of 7 facial expressions (6 basic facial expressions: anger, disgust, fear, happiness, sadness and surprise + 1 neutral) posed by 10

Fig. 3. Confusion matrix obtained by EPGDFS for the human action videos categorization application.

![Confusion Matrix](image)

**TABLE III**

<table>
<thead>
<tr>
<th>Method</th>
<th>$M$ (Standard Deviation)</th>
<th>Accuracy (%) (Standard Deviation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPGDFS</td>
<td>9.80 (0.15)</td>
<td>87.25 (1.28)</td>
</tr>
<tr>
<td>EPGD</td>
<td>9.70 (0.19)</td>
<td>85.97 (1.69)</td>
</tr>
<tr>
<td>EPGauFS</td>
<td>9.65 (0.26)</td>
<td>85.03 (1.37)</td>
</tr>
<tr>
<td>EMGDFS</td>
<td>9.60 (0.23)</td>
<td>84.72 (2.34)</td>
</tr>
</tbody>
</table>

Fig. 4. Classification accuracy vs. number of aspects for different algorithms applied to the human action videos categorization task.

![Classification Accuracy](image)

Fig. 5. Feature saliency for each aspect in the case of the human action videos categorization application.

![Feature Saliency](image)
Japanese female models aged from 20 to 40. Sample images from this data set with different facial expressions are shown in Fig. 6.

In our approach, a preprocessing step suggested in [37] is adopted by cropping original images into $110 \times 150$ pixels to reduce the influences of background. Notice that, the cropped images remain the central part of facial expression. Next, LBP features are extracted from face images. Specifically, each cropped face image is first divided into small regions from which LBP histograms are extracted and concatenated into a single feature histogram representing the face image [34]. We adopt the same experimental setting for extracting LBP features as in [34]: we use a 59-bin LBP operator in the $(8, 2)$ neighborhood (which means 8 sampling points on a circle of radius of 2) and divide each image $(110 \times 150)$ into $18 \times 21$ pixels regions. Thus, face images are divided into 42 ($6 \times 7$) regions and are then represented by LBP histograms with a length of 2478 ($59 \times 42$). Then, these histograms are normalized. Finally, the EPGDFS is applied for clustering. We have evaluated the performance of the proposed algorithm by running it twenty times. The confusion matrix for the JAFFE data set provided by the EPGDFS is shown in Fig. 7. Table IV illustrates

![Fig. 7. Confusion matrix obtained by EPGDFS for the facial expression recognition task.](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>$\hat{M}$</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPGDFS</td>
<td>6.85 (0.12)</td>
<td>87.71 (1.07)</td>
</tr>
<tr>
<td>EPGD</td>
<td>6.80 (0.09)</td>
<td>85.97 (1.15)</td>
</tr>
<tr>
<td>EPGauFS</td>
<td>6.75 (0.14)</td>
<td>84.55 (1.34)</td>
</tr>
<tr>
<td>EMGDFS</td>
<td>6.75 (0.17)</td>
<td>83.86 (1.58)</td>
</tr>
</tbody>
</table>

The average face expression recognition accuracy and the number of components ($\hat{M}$) computed over 20 runs by different algorithms. The numbers in parenthesis are the standard deviations of the corresponding quantities.

More complicated techniques for locating eyebrows and mouth may be used for registering the face in this preprocessing step. This is, however, clearly out of the scope of this work.

The average recognition accuracy and the average number of components obtained by each algorithm. As we can observe, our algorithm outperforms the other three algorithms in terms of the highest classification accuracy (87.71%) and the most accurately detected number of categories (6.85). In addition, we have tested the performance when using SVM (85.83%), $k$-NN (81.36%) and naive Bayes (82.54%). It is clear that EPGDFS outperforms all of them. In addition, the features saliencies calculated by EPGDFS show that the 2478 features have different degrees of importance. More specifically, 358 features have saliencies lower than 0.5, and thus provide less contribution for recognizing facial expression.
V. CONCLUSION

In this paper, we have proposed an EP framework for learning finite GD mixture models with feature selection. Within this framework, in which prior knowledge is incorporated naturally, features relevancies and the number of clusters can be determined simultaneously which allows to avoid over-fitting. Extensive experiments have been conducted and have involved synthetic data and real applications namely action videos categorization and facial expression recognition. 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 (with a stick-breaking representation, for instance) or the development of an online learning approach to tackle the crucial problem of dynamic data clustering.

REFERENCES