Bayesian Feature and Model Selection for Gaussian Mixture Models

Constantinos Constantinopoulos,
Michalis K. Titsias, and
Aristidis Likas, *Senior Member*, *IEEE*

Abstract—We present a Bayesian method for mixture model training that simultaneously treats the feature selection and the model selection problem. The method is based on the integration of a mixture model formulation that takes into account the saliency of the features and a Bayesian approach to mixture learning that can be used to estimate the number of mixture components. The proposed learning algorithm follows the variational framework and can simultaneously optimize over the number of components, the saliency of the features, and the parameters of the mixture model. Experimental results using high-dimensional artificial and real data illustrate the effectiveness of the method.

Index Terms—Mixture models, feature selection, model selection, Bayesian approach, variational training.

1 Introduction

MIXTURE models constitute a widely used approach for unsupervised learning problems. Fitting a mixture model to the distribution of the data can be interpreted as identifying clusters with the mixture components. The estimation of the parameters of mixture models with a predefined number of components is usually achieved through likelihood maximization using the EM algorithm or several variants [1]. Apart from the selection of the number of components, a problem that naturally arises, especially in high-dimensional data, deals with the detection of the salient features. Intuitively, salient features are those that facilitate the modeling task and produce reasonable results. Regarding mixtures with Gaussian components, the salient features describe data with multimodal distribution and modes that can be sufficiently represented with Gaussian components. On the other hand, uniform or unimodal features are irrelevant to clustering. Moreover, they may confuse inference by increasing the complexity of the model, for examples see [2], [3]. Notice that choosing the features and finding the number of components are strongly dependent problems. Clearly, for different feature subsets, we might get different estimations for the number of clusters, see [4] for a discussion. It is common sense that using more features may lead to more complex structures in the data space and, consequently, more clusters. This suggests that choosing the features and selecting the number of clusters should be addressed simultaneously.

To address both feature and model selection, we present a Bayesian variational framework for training a two-level mixture model that maximizes a lower bound of the marginal likelihood. We employ the model proposed in [3], i.e., a Gaussian mixture model that incorporates a feature saliency determination process, where each feature is useful up to a probability. So, when this probability obtains a close to zero value the feature is effectively removed from consideration. This approach is attractive since it does not require an explicit search over the possible subsets of the features which is generally an infeasible task. According to the Bayesian framework, we place prior distributions over the parameters of the model and maximize the marginal likelihood given the mixing coefficients and the feature saliencies. For optimization, we use variational methods to derive an EM-like algorithm [5], following the approach proposed in [6], [7].

In Section 2, we briefly present related work from the literature. In Section 3, we describe the proposed model, the Bayesian framework for feature and model selection, and the variational learning for parameter estimation. Comparative experiments are described in Section 4 and conclusions in Section 5.

2 Related Work on Unsupervised Feature Selection

The feature selection problem, although extensively studied along in the classification framework, is only recently considered for clustering. Two major approaches have been proposed; in the wrapper approach, a feature subset selection algorithm exists as a wrapper around the clustering algorithm. The feature selection algorithm conducts a search for a good subset using the clustering algorithm as a part of the function that evaluates the candidate feature subsets. The second approach treats clustering and feature selection simultaneously, defining a proper objective function. Optimization of the objective function yields a feature subset and the clustering solution in the corresponding feature space. In the remainder of this section, we describe briefly representative methods.

Dy and Brodley [4] use a wrapper approach for feature selection. They search the space of feature subsets and evaluate each candidate subset by first clustering using the corresponding features and then evaluating the result using appropriate measures. To search the feature space, they use sequential forward search starting with zero features and, sequentially, adding one feature at a time. To identify the best feature subset, the scatter separability and the maximum-likelihood criteria are utilized. For data clustering, they employ Gaussian mixture models trained using EM. To estimate the number of components, they merge clusters one at a time and use the BIC criterion to select the best model.

Law et al. [3] follow the second approach and define feature saliency as a probability. They use Gaussian mixture models for clustering and assume independent features given a mixture component. Given a feature, observations are considered independent of the components up to a probability and follow a common distribution. The complement of this probability is the measure of feature saliency. To estimate the mixture models, the MML criterion is employed and a component-wise version of the EM algorithm that enforces a pruning behavior over the components of the model. As stated in [3], the method can be viewed as a MAP approach with improper priors on mixture weights and feature saliencies.

Carbonetto et al. [2] propose a Bayesian shrinkage model. They use Gaussian mixture models for clustering and define conjugate priors over all mixture parameters. Moreover, they place hyperpriors over the parameters of the priors of means and mixing weights. Using a shrinkage prior above the prior of the means, they intend to discover the irrelevant features and concentrate the corresponding estimates of the means around common values across components. For parameter estimation, they resort to the MAP approach.

Liu et al. [8] conduct a principal component analysis or correspondence analysis for data reduction and then fit a Gaussian mixture model to the data having been projected to the several major factors resulting from the analysis. To select a subset of the factors, they assume that a datum has its first k features follow a mixture model and the remaining features follow a simple Gaussian distribution. They treat k as a random variable and

- *C. Constantinopoulos and A. Likas are with the Department of Computer Science, University of Ioannina, Ioannina, GR 45110, Greece.
E-mail: {ccostas, arly}@cs.uoi.gr.*
- *M.K. Titsias is with the School of Informatics, University of Edinburgh, Edinburgh EH1 2QL, UK. E-mail: M.Titsias@sms.ed.ac.uk.*

Manuscript received 4 May 2005; revised 14 Dec. 2005; accepted 16 Jan. 2006; published online 13 Apr. 2006.
Recommended for acceptance by B.J. Frey.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0229-0505.

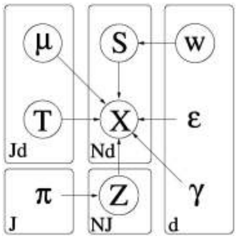

Fig. 1. Graphical model for the generation of the observed data assuming a Bayesian mixture density model and allowing noisy features. Symbols in circles denote random variables; otherwise, they denote model parameters. Plates denote repetitions and the number of repetitions for the variables in a plate is depicted in the bottom-left corner.

propose a Bayesian formulation and Markov Chain Monte Carlo strategies to tackle the problem.

The method we propose engages the same model as in [3] to describe the relevance of features, but integrates model and feature selection under a Bayesian framework. The MML approach used in [3] is based on a statistical criterion and is obtained after several assumptions and simplifications. Using the Bayesian framework, our method is expected to be more robust, especially for sparse data sets. Evidence from the experiments we conducted supports our effort.

It must be noted that our approach to feature selection assumes a weighting of the features and the weight of each feature is the same for all the clusters. A different approach is subspace clustering that assumes separate feature weights for each cluster. Thus, each cluster is differentiated from the rest in a particular subspace, for methods following this approach, see [9], [10].

3 A Bayesian Mixture Model with Feature Saliency

In this section, we present a Bayesian method for learning mixture models that automatically determines the number of components and the saliencies of the features. In Section 2.1, we define the Bayesian mixture model with feature saliency and, in Section 2.2, we present a variational training method for this model.

3.1 Bayesian Framework

Assume a set of data $X = \{x^n | n = 1, \dots, N\}$, where each x^n is a real feature vector in a d-dimensional space. We wish to model these data by training a mixture model. We further assume that each component density of the mixture is factorized over the features so that the features are considered to be independent given a component. Some of the features might be irrelevant for modeling while others may be more useful. Instead of assuming that there is a deterministic separation between useful and noisy features, we assume that a feature is useful up to a probability. Thus, given some component, we assume that a feature of x is drawn from a mixture of two univariate subcomponents, as proposed in [3]. The first subcomponent that is different for each mixture component generates "useful" data, while the second subcomponent that is common to all mixture components generates "noisy" data.

In this work, the above model for feature saliency is integrated in the Bayesian framework suggested in [7] for estimating the number of components in mixture models. We assume that data set X has been generated from the graphical model illustrated in Fig. 1. A maximum number J of Gaussian components is initially supposed and the density corresponding to the two-level mixture model previously explained is given by:

$$f(x) = \sum_{j=1}^{J} \pi_j \prod_{i=1}^{d} \varphi(x_i), \tag{1}$$

$$\varphi(x_i) = w_i \mathcal{N}(x_i; \mu_{ji}, \tau_{ji}) + (1 - w_i)\mathcal{N}(x_i; \varepsilon_i, \gamma_i). \tag{2}$$

This graphical model implies a dependence of the observed variable x^n on the jth mixture component through the hidden variables z_j^n, where $z_j^n \in \{0, 1\}$ and $\sum_j z_j^n = 1$. If x^n is generated from the jth component, then the value of z_j^n is one; otherwise, it is zero. The saliency of features is expressed through the hidden variables s_i^n, where $s_i^n \in \{0, 1\}$. If the value of s_i^n is one, then the ith feature of x^n has been generated from the "useful" subcomponent; otherwise, it has been generated from the "noisy" subcomponent.

Given the sets of hidden variables $Z = \{z_j^n\}$ and $S = \{s_i^n\}$, the data is assumed to be independently drawn from a Gaussian distribution

$$\begin{aligned} p(X|Z, \mu, T, S, \varepsilon, \gamma) = \prod_{n=1}^{N} \prod_{j=1}^{J} \Bigg[\prod_{i=1}^{d} \mathcal{N}(x_i^n; \mu_{ji}, \tau_{ji})^{s_i^n} \\ \times \mathcal{N}(x_i^n; \varepsilon_i, \gamma_i)^{1-s_i^n} \Bigg]^{z_j^n}. \end{aligned} \tag{3}$$

The sets $\mu = \{\mu_{ji}\}$ and $T = \{\tau_{ji}\}$ accumulate the means and the inverse variances (precisions) of the "useful" subcomponents. Correspondingly, $\varepsilon = \{\varepsilon_i\}$ and $\gamma = \{\gamma_i\}$ are the sets of parameters for the "noisy" subcomponent. The distribution of the hidden variables Z given the mixing probabilities $\pi = \{\pi_j\}$ and of the hidden variables S given the probabilities $w = \{w_i\}$ (feature saliencies) are given by

$$p(Z|\pi) = \prod_{n=1}^{N} \prod_{j=1}^{J} \pi_j^{z_j^n}, \tag{4}$$

$$p(S|w) = \prod_{n=1}^{N} \prod_{i=1}^{d} w_i^{s_i^n} (1 - w_i)^{1-s_i^n}. \tag{5}$$

The likelihood of the observed data given the parameters is obtained by marginalizing out the hidden variables Z and S from $p(X, Z, S|\pi, \mu, T, w, \varepsilon, \gamma)$

$$p(X|\pi, \mu, T, w, \varepsilon, \gamma) = \prod_{n=1}^{N} \sum_{j=1}^{J} \pi_j \prod_{i=1}^{d} \varphi(x_i^n). \tag{6}$$

This is the usual quantity that the maximum-likelihood framework maximizes over the parameters. However, this objective function cannot be used for selecting the number of components. Thus, it is not useful, in our case, since we wish to estimate the number of components. In [3], this problem is addressed by applying the MML criterion and a component-wise version of the EM algorithm that enforces a pruning behavior over the components of the model. In our method, a Bayesian approach for model selection is adopted [7]. In particular, we introduce Gaussian and Gamma priors for μ and T, respectively,

$$p(\mu) = \prod_{j=1}^{J} \prod_{i=1}^{d} \mathcal{N}(\mu_{ji}; m_i, c), \tag{7}$$

$$p(T) = \prod_{j=1}^{J} \prod_{i=1}^{d} \mathcal{G}(\tau_{ji}; \alpha, \beta), \tag{8}$$

and integrate them out to obtain the marginal likelihood. The hyperparameters m, c, α, and β control the prior distributions and are fixed at values that form broad and uninformative priors. More specifically, m is set to the mean of all data, while $c = \alpha = \beta = 10^{-16}$, which is a very small number near the machine precision. The method does not exhibit sensitivity for hyperparameter values on this near zero scale. Notice that a prior has not been imposed on the mixing weights and the feature saliencies which are considered as model parameters. Setting some mixing weights equal to zero allows for elimination of the corresponding components from the model. The learning approach we followed for the proposed model is described next.

3.2 Variational Learning

To simplify notation, we define $\theta = \{Z, \mu, T, S\}$ the set of random variables and $\vartheta = \{\pi, w, \varepsilon, \gamma\}$ the set of parameters. The learning method we propose estimates the parameters ϑ of the model through maximization of the marginal likelihood $p(X|\vartheta)$:

$$p(X|\vartheta) = \sum_{Z,S} \int p(X, \theta|\vartheta) \mathrm{d}\mu \, \mathrm{d}T, \tag{9}$$

with respect to the mixing probabilities π, feature saliencies w and the parameters of the noise component. Note that, by assuming suitable prior distributions on the component parameters and marginalizing them out, we expect to smooth the likelihood surface (6) and obtain a marginal likelihood that is more robust to over fitting. This methodology was proposed in [7] to optimize over the mixing probabilities π and infer the number of components in a typical mixture model with remarkable results.

Since the integration in (9) is intractable, the *variational* approach is employed, which suggests the maximization of a lower bound $\mathcal{L}$ of the logarithm of the marginal likelihood:

$$\begin{aligned} \mathcal{L}[Q, \vartheta] &= \sum_{Z,S} \int Q(\theta) \log \frac{p(X, \theta|\vartheta)}{Q(\theta)} \mathrm{d}\mu \, \mathrm{d}T & (10) \\ &\leq \log p(X|\vartheta). & (11) \end{aligned}$$

The bound $\mathcal{L}$ is a functional of an arbitrary distribution $Q(\theta)$ that approximates the posterior distribution $p(\theta|X, \vartheta)$. In order to maximize $\mathcal{L}$, an iterative procedure is adopted that consists of two steps at each iteration: first, maximization of the bound with respect to Q and, subsequently, with respect to ϑ.

According to the *mean field* approximation, we do not assume any specific form for Q, except that it is constrained to be a product of the form $Q(\theta) = Q_Z(Z)Q_\mu(\mu)Q_T(T)Q_S(S)$. Maximizing $\mathcal{L}$ with respect to the functional form of Q_Z, Q_μ, Q_T, and Q_S, the standard variational approach provides the following general form of the solutions:

$$Q(\theta_i) = \frac{\exp\langle P(X, \theta|\vartheta)\rangle_{k \neq i}}{\int \exp\langle P(X, \theta|\vartheta)\rangle_{k \neq i} \theta_i}, \tag{12}$$

where $\langle \cdot \rangle_{k \neq i}$ denotes an expectation with respect to the distributions $Q_k(\theta_k)$ for all $k \neq i$. For our Bayesian model, (12) yields:

$$Q_Z(Z) = \prod_{n=1}^{N} \prod_{j=1}^{J} r_{jn}^{z_j^n}, \tag{13}$$

$$Q_\mu(\mu) = \prod_{j=1}^{J} \prod_{i=1}^{d} \mathcal{N}(\mu_{ji}; m_{ji}^v, c_{ji}^v), \tag{14}$$

$$Q_T(T) = \prod_{j=1}^{J} \prod_{i=1}^{d} \mathcal{G}(\tau_{ji}; \alpha_{ji}^v, \beta_{ji}^v), \tag{15}$$

$$Q_S(S) = \prod_{n=1}^{N} \prod_{i=1}^{d} \rho_{in}^{s_i^n} (1 - \rho_{in})^{1 - s_i^n}. \tag{16}$$

The *variational parameters* $r_{jn}, m_{ji}^v, c_{ji}^v, \alpha_{ji}^v, \beta_{ji}^v$, and ρ_{in} emerge from the maximization and determine the densities involved in Q. The variational parameters themselves are defined using the expected values of z_j^n, μ_{ji}, τ_{ji}, s_i^n, and functions of them. Using the functional forms of Q_Z, Q_μ, Q_T, and Q_S, we can derive the corresponding expectations and use them in the definitions of the variational parameters. After some algebra, the following equations are obtained:

$$r_{jn} = \frac{\pi_j \tilde{r}_{jn}}{\sum_{j=1}^{J} \pi_j \tilde{r}_{jn}}, \tag{17}$$

$$\begin{aligned} \tilde{r}_{jn} = \exp\Bigg\{ &\frac{1}{2} \sum_{i=1}^{d} \rho_{in} \Big[\psi(\alpha_{ji}^v) - \log \beta_{ji}^v\Big] \\ &- \frac{1}{2} \sum_{i=1}^{d} \rho_{in} \frac{\alpha_{ji}^v}{\beta_{ji}^v} \left[(x_i^n - m_{ji}^v)^2 + \frac{1}{c_{ji}^v}\right] \Bigg\}, \end{aligned} \tag{18}$$

$$m_{ji}^v = \frac{c \, m_i + (\alpha_{ji}^v / \beta_{ji}^v) \sum_{n=1}^{N} r_{jn} \rho_{in} x_i^n}{c + (\alpha_{ji}^v / \beta_{ji}^v) \sum_{n=1}^{N} r_{jn} \rho_{in}}, \tag{19}$$

$$c_{ji}^v = c + \frac{\alpha_{ji}^v}{\beta_{ji}^v} \sum_{n=1}^{N} r_{jn} \rho_{in}, \tag{20}$$

$$\alpha_{ji}^v = \alpha + \frac{1}{2} \sum_{n=1}^{N} r_{jn} \rho_{in}, \tag{21}$$

$$\beta_{ji}^v = \beta + \frac{1}{2} \sum_{n=1}^{N} r_{jn} \rho_{in} \left[(x_i^n - m_{ji}^v)^2 + \frac{1}{c_{ji}^v}\right], \tag{22}$$

$$\rho_{in} = \frac{w_i \tilde{\rho}_{in}}{w_i \tilde{\rho}_{in} + (1 - w_i) \xi_{in}}, \tag{23}$$

$$\begin{aligned} \tilde{\rho}_{in} = \exp\Bigg\{ &\frac{1}{2} \sum_{j=1}^{J} r_{jn} \Big[\psi(\alpha_{ji}^v) - \log \beta_{ji}^v\Big] \\ &- \frac{1}{2} \sum_{j=1}^{J} r_{jn} \frac{\alpha_{ji}^v}{\beta_{ji}^v} \left[(x_i^n - m_{ji}^v)^2 + \frac{1}{c_{ji}^v}\right] \Bigg\}, \end{aligned} \tag{24}$$

$$\xi_{in} = \exp\left\{ -\frac{1}{2} \gamma_i (x_i^n - \varepsilon_i)^2 + \frac{1}{2} \log \gamma_i \right\}, \tag{25}$$

where $\psi(x) = \mathrm{d} \log \Gamma(x) / \mathrm{d}x$. The maximization of $\mathcal{L}$ with respect to Q aims to find a tight bound of the log marginal likelihood. Although an exact maximization of $\mathcal{L}$ with respect to the variational parameters is impossible, as they are coupled together in a nonlinear way, we can still improve the bound by iteratively updating the parameters using (17) to (24). An analogous approach is taken in [7].

After the maximization of $\mathcal{L}$ with respect to Q, the second step of the method requires maximization of $\mathcal{L}$ with respect to π_j, w_i, ε_i, and γ_i. Setting the derivative of $\mathcal{L}$ with respect to the parameters equal to zero, we get the following update rules:

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} r_{jn}, \tag{26}$$

$$w_i = \frac{1}{N} \sum_{n=1}^{N} \rho_{in}, \tag{27}$$

$$\varepsilon_i = \frac{\sum_{n=1}^{N} \rho_{in} x_i^n}{\sum_{n=1}^{N} \rho_{in}}, \tag{28}$$

$$\frac{1}{\gamma_i} = \frac{\sum_{n=1}^{N} \rho_{in} (x_i^n - \varepsilon_i)^2}{\sum_{n=1}^{N} \rho_{in}}. \tag{29}$$

The above two-step procedure is repeated until convergence. Convergence can be monitored through inspection of the variational bound. The above algorithm has the property that it does not allow for Gaussians with similar parameters to fit the same cluster. Consequently, one of them dominates and the others are removed. Starting with a large number of components, the competition among components finally yields a model where the redundant components have been eliminated. Simultaneously, the update of the parameters w_i enables the determination of the feature saliencies.

4 Experimental Results

We compared our method (*varFnMS*) with the method of Law et al. [3] (*FnMs*) for clustering high-dimensional artificial and real data. We also conducted the same experiments using the method in [7] (*varMS*). The first series of experiments was for clustering artificially generated shapes. More specifically, we created 9×9 gray-scale

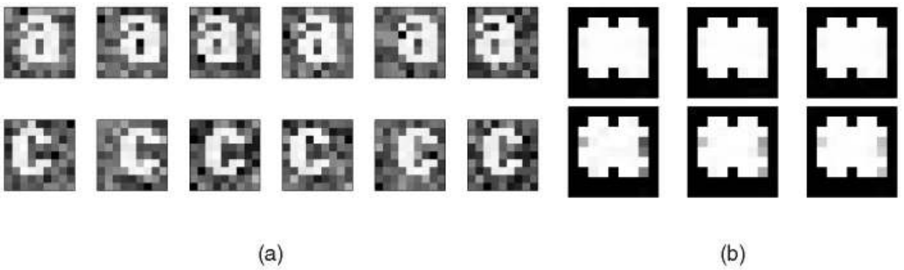

Fig. 2. (a) A sample of the artificially created images. (b) Saliencies are illustrated on the top row using varFnMS and in the bottom row using FnMS. From left to right, results with data sets having 180, 240, and 300 images, respectively, are provided.

images, each one illustrating the shape of the character "a" or "c." The shape in each image has been placed in one of three different positions so that 41 pixels across the image border were always background. The intensities of the background pixels were drawn from a Gaussian $\mathcal{N}(0.4, 12 \cdot 10^{-3})$ and the foreground pixels from a Gaussian $\mathcal{N}(0.85, 0.4 \cdot 10^{-3})$, then all intensities were normalized in $[0, 1]$. Fig. 2a illustrates some of the images used. It is clear that six clusters exist and at least the 41 pixels are irrelevant. We applied the three methods on data sets with various numbers of images. The same number of images per cluster was used in each run. For each data set, we run 10 trials initially using 30 components. For data sets with 180, 240, and 300 images, our method identified correctly the six clusters 4, 10, and 10 times, respectively. The FnMS method identified the six clusters 0, 5, and 10 times, respectively, strongly affected from the reduction in the size of the data set. Fig. 2b provides a visual illustration of the expected saliencies estimated by varFnMS and FnMS. The varMS method with 30 initial components never identified the correct number of components, providing on average 12 components for all three data sets.

For experiments with real data we used the "multiple feature database" used in [11], which is available from the UCI repository [12]. It consists of features of handwritten numerals ("0"-"9") extracted from a collection of Dutch utility maps. From each class, 200 patterns have been digitized to produce a total of 2,000 images. Digits are represented in terms of various feature sets. We used three data sets, the first describing the digits using Zernike moments (47 features), the second using Fourier coefficients (76 features), and the third profile correlations (216 features). The clustering performance was evaluated on test data using the "classification" error. To compute the "classification" error given a clustering of the training data, we assign to each cluster the class of the majority of its data. Then, we classify each test pattern to the class of the cluster it has been assigned and compute the classification error given ground truth. To estimate the expected classification error and the number of components we carried out 20 trials, splitting the data in half to create the train and test sets preserving class ratio. The results are aggregated in Tables 1 and 2, initially using 30 and 50 components, respectively.

Our method always gives better error, but uses more components compared to FnMS. Both methods converge to a similar number of components independently of the initial number, thus their clustering solutions are different but consistent. On the other hand, varMS is affected from the initial number of components and

TABLE 1
Expected Error and Number of Components Using varFnMS, varMS, and FnMS, with 30 Initial Components

		Zernike	*Fourier*	*Profile*
varFnMS	*error*	0.39 (0.07)	0.35 (0.06)	0.13 (0.01)
	comp.	26.8 (6.4)	24.6 (7.2)	26.3 (3.7)
varMS	*error*	0.37 (0.02)	0.34 (0.02)	0.14 (0.01)
	comp.	29.5 (0.5)	27.5 (1.2)	27.6 (1.5)
FnMS	*error*	0.53 (0.02)	0.50 (0.07)	0.77 (0.04)
	comp.	10.7 (1.2)	6.1 (0.9)	2.3 (0.7)

In parentheses, the corresponding standard deviations.

TABLE 2
Expected Error and Number of Components Using varFnMS, varMS, and FnMS, with 50 Initial Components

		Zernike	*Fourier*	*Profile*
varFnMS	*error*	0.37 (0.03)	0.32 (0.02)	0.12 (0.01)
	comp.	28.1 (2.3)	25.0 (1.6)	29.6 (2.4)
varMS	*error*	0.35 (0.02)	0.31 (0.02)	0.11 (0.01)
	comp.	44.6 (1.9)	37.6 (2.9)	41.3 (2.1)
FnMS	*error*	0.53 (0.01)	0.52 (0.03)	0.76 (0.04)
	comp.	10.8 (1.1)	5.7 (0.7)	2.3 (0.4)

In parentheses, the corresponding standard deviations.

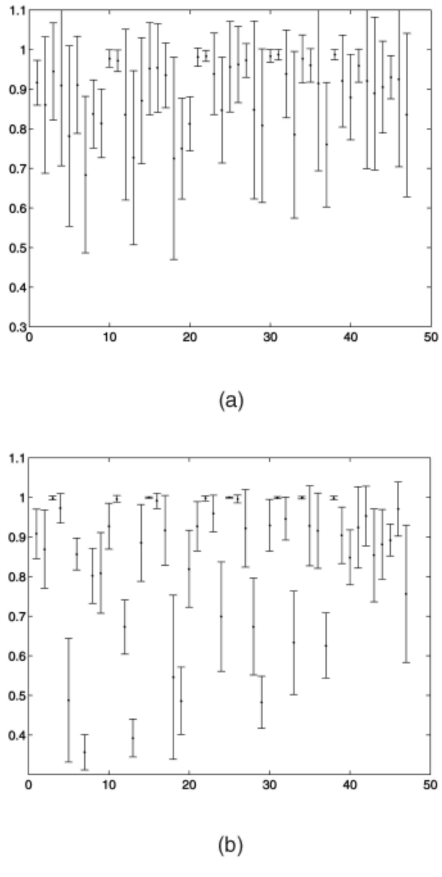

Fig. 3. Saliencies of Zernike features using (a) *varFnMS* and (b) *FnMS*.

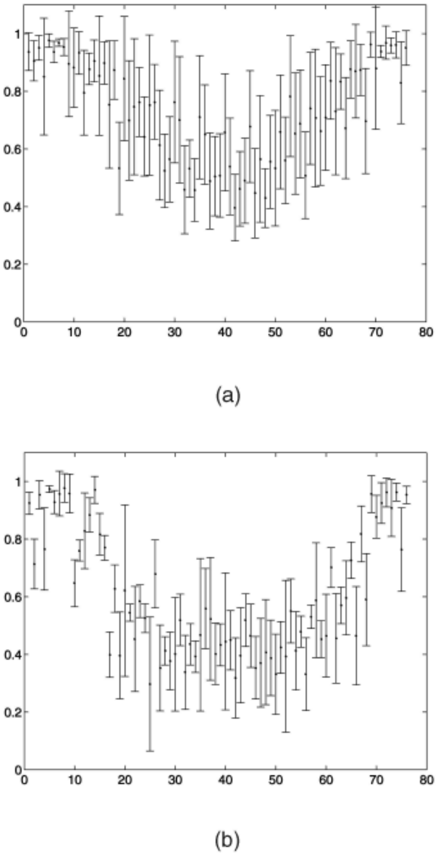

Fig. 4. Saliencies of Fourier features using (a) *varFnMS* and (b) *FnMS*.

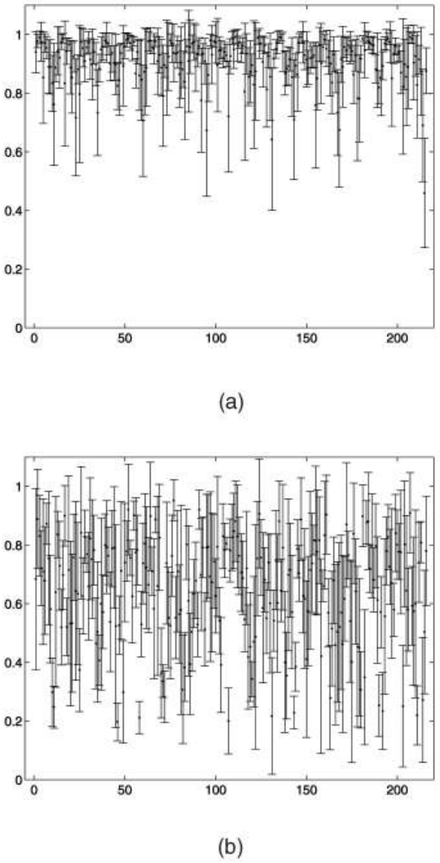

Fig. 5. Saliencies of profile correlation features using (a) *varFnMS* and (b) *FnMS*.

tends to keep most of them. The same behavior was also noticed for experiments with 60 initial components. Also of interest is that, for varFnMS, as the number of features in the data set increases, the number of components varies slightly, but the error drops significantly. Apparently, the method exploits a larger number of features to improve its solution and is not affected from the sparsity of data. Regarding the estimated saliency of features, we present error-bar plots in Figs. 3, 4, and 5 for models initialized with 30 components. Note that, in Fig. 4, the Fourier coefficients tend to be irrelevant to clustering as we approach the middle band and, in Fig. 6, the expected saliency using varFnMS has a local minimum every 12 features. As a general comment, the FnMS method provides smaller values for the saliencies, while varFnMS is more conservative.

5 Conclusions

We have presented a variational Bayesian approach for mixture learning that can automatically determine the number of components and the saliency of features. Our experiments show that this algorithm outperforms the MML-based approach [3] in the presence of sparse data and this illustrates the importance of the Bayesian framework we adopted. As expected, the MML criterion used in [3] requires more data to fully exploit the underlying model of feature saliency. Also, our approach exhibits more consistent behavior than the method in [7], regarding the number of components used. This is to be expected as the later approach does not use feature selection and in high dimensions this hinders model selection.

The main restriction of the proposed method is that the features are assumed to be conditionally independent given the component. We plan to elaborate further on this issue and generalize our method so that the full covariances of the useful features can be used and simultaneously the feature saliencies can be estimated.

Acknowledgments

The authors would like to thank Professor M. Figueiredo for providing the Matlab code for the experiments with the *FnMS* method. This research was cofunded by the European Union in the framework of the program "Heraklitos" of the "Operational Program for Education and Initial Vocational Training" of the third Community Support Framework of the Hellenic Ministry of Education, funded 25 percent from national sources and 75 percent from the European Social Fund (ESF).

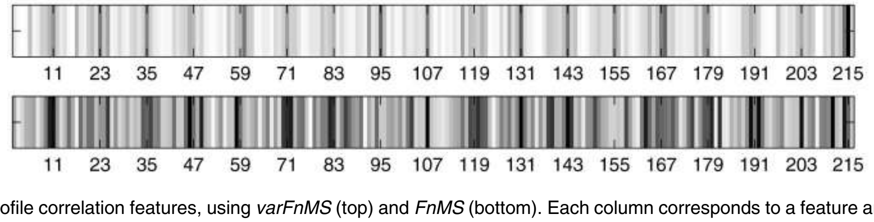

Fig. 6. Expected saliency of profile correlation features, using *varFnMS* (top) and *FnMS* (bottom). Each column corresponds to a feature and the intensities are scaled so that black corresponds to the minimum expected saliency and white to the maximum.

REFERENCES

[1] B.G. McLachlan and D. Peel, *Finite Mixture Models.* Wiley, 2000.

[2] P. Carbonetto, N. de Freitas, P. Gustafson, and N. Thompson, "Bayesian Feature Weighting for Unsupervised Learning, with Application to Object Recognition," *Proc. Ninth Int'l Conf. Artificial Intelligence and Statistics,* 2003.

[3] M.H. Law, M.A.T. Figueiredo, and A.K. Jain, "Simultaneous Feature Selection and Clustering Using a Mixture Model," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1154-1166, Sept. 2004.

[4] J. Dy and C. Brodley, "Feature Selection for Unsupervised Learning," *J. Machine Learning Research,* vol. 5, pp. 845-889, 2004.

[5] R.M. Neal and G.E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphical Models,* M.I. Jordan, ed., pp. 355-368, Kluwer, 1998.

[6] H. Attias, "A Variational Bayesian Framework for Graphical Models," *Advances in Neural Information Processing Systems 12,* MIT Press, 2000.

[7] A. Corduneanu and C.M. Bishop, "Variational Bayesian Model Selection for Mixture Distributions," *Proc. Eighth Int'l Conf. Artificial Intelligence and Statistics,* T. Richardson and T. Jaakkola, eds., pp. 27-34, Morgan Kaufmann, 2001.

[8] J.S. Liu, J.L. Zhang, M.J. Palumbo, and C.E. Lawrence, "Bayesian Clustering with Variable and Transformation Selections," *Bayesian Statistics,* vol. 7, pp. 249-276, 2003.

[9] J.H. Friedman and J.J. Meulman, "Clustering Objects on Subsets of Attributes," *J. Royal Statistical Soc.,* vol. 66, no. 4, pp. 815-849, 2004.

[10] P.D. Hoff, "Model-Based Subspace Clustering," *Bayesian Analysis,* vol. 1, no. 2, pp. 321-344, 2006.

[11] A.K. Jain, R. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-38, Jan. 2000.

[12] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases," 1998, http://www.ics.uci.edu/mlearn/MLRepository.html.