Class Conditional Density Estimation Using Mixtures with Constrained Component Sharing

Michalis K. Titsias and
Aristidis Likas, *Member*, *IEEE*

Abstract—We propose a generative mixture model classifier that allows for the class conditional densities to be represented by mixtures having certain subsets of their components shared or common among classes. We argue that, when the total number of mixture components is kept fixed, the most efficient classification model is obtained by appropriately determining the sharing of components among class conditional densities. In order to discover such an efficient model, a training method is derived based on the EM algorithm that automatically adjusts component sharing. We provide experimental results with good classification performance.

Index Terms—Mixture models, classification, density estimation, EM algorithm, component sharing.

1 Introduction

In this paper, we consider classification methods based on mixture models. In the usual *generative* approach, training is based on partitioning the data according to class labels and then estimating each class conditional density $p(x|C_k)$ (maximizing the likelihood) using the data of class C_k. The above approach has been widely used [7], [8], [13], [16], and one of its great advantages is that training can be easily performed using the EM algorithm [5]. An alternative approach is *discriminative* training, where mixture models are suitably normalized in order to provide a representation of the posterior probability $P(C_k|x)$ and training is based on the maximization of the conditional likelihood. Discriminative training [2], [9], [15] essentially takes advantage of the flexibility of mixture models to represent the decision boundaries and must be considered different in principle from the generative approach where an explanation (the distribution) of the data is provided. It must be pointed out that the model used in this paper is a *generative* mixture model classifier, so our training approach is based on estimating class conditional densities.

Consider a classification problem with K classes. We model each class conditional density by the following mixture model:

$$p(x|C_k;\pi_k,\theta)=\sum_{j=1}^{M}\pi_{jk}p(x|j;\theta_j)\quad k=1,\ldots,K, \tag{1}$$

where π_{jk} is the mixture coefficient representing the probability $P(j|C_k)$, θ_j the parameter vector of component j and $\theta=(\theta_1,\ldots,\theta_M)$. Also, we denote with π_k the vector of all mixing coefficients π_{jk} associated with class C_k. The mixing coefficients cannot be negative and satisfy

$$\sum_{j=1}^{M}\pi_{jk}=1,\quad k=1,\ldots,K. \tag{2}$$

This model has been studied in [7], [13], [16], and in the sequence it will be called the *common components model*, since all the component densities are shared among classes. From a generative point of view, the above model suggests that differently labeled data that are similarly distributed in some input subspaces can be represented by common density models. An alternative approach is to assume independent or separate mixtures to represent each class data [1], [12]. A theoretical study of that model for the case of Gaussian components can be found in [8]. Next, we refer to that model as the *separate mixtures model*. From a generative point of view, the later model assumes a priori that there exist no common properties of data coming from different classes (for example common clusters). If the total number of component density models is M, we can consider the separate mixtures as a constrained version of the common components model also having M components [16].

Both methods have advantages and disadvantages depending on the classification problem at hand. More specifically, an advantage of the common components model is that training and selection of the total number of components M, is carried out using simultaneously data from all classes. Also, by using common components we can explain data of many classes by the same density model, thus reducing the required number of model parameters. Ideally, we wish after training the model, a component j that remains common (for at least two different classes the parameter π_{jk} is not zero) to represent data that highly overlap, which means that the underlying distribution of the differently labeled data is indeed locally similar. However, we have observed [16] that, if we allow the components to represent data of any class, we can end up with a maximum-likelihood solution where some components represent data of different classes that slightly overlap. In such cases, those components are allocated above the true decision boundary, causing classification inefficiency. Regarding the separate mixtures, a disadvantage of this method is that learning is carried out by partitioning the data according to the class labels and dealing separately with each mixture model. Thus, the model ignores any common characteristics between data of different classes so we cannot reduce the totally used number of density models or equivalently the number of parameters that must be learned from the data. However, in many cases, the separate mixtures model trained through likelihood maximization provides more discriminative representation of the data than the common components model.

In this paper, we consider a general mixture model classifier that encompasses the above two methods as special cases. The model assumes that each component is constrained to possibly represent data of only a *subset* of the classes. We refer to this model as the Z-model, where Z is an indicator matrix specifying these constrains. By fixing the Z matrix to certain values, we can obtain several cases such as the common components and separate mixtures model. However, in general, the Z values are part of the unknown parameters and we wish to discover a choice of Z that leads to improved discrimination. In the next section, we provide an example where, for a fixed total number of components, the best classifier is obtained for an appropriate choice of matrix Z. In order to specify the Z matrix, we propose a method based on the maximization of a suitably defined objective function using the EM algorithm [5]. After convergence, this algorithm provides an appropriate Z matrix specification and, additionally, effective initial values of all the other model parameters. Then, by applying again, the EM algorithm to maximize the likelihood for fixed Z values, the final solution is obtained. Section 2 describes the proposed mixture model classifier (Z-model) and provides an example illustrating the usefulness of constrained component sharing. In Section 3, a training algorithm is presented based on the EM algorithm that simultaneously adjusts constraints and parameter values. Experimental results using several classification data sets are presented in Section 4. Finally, Section 5 provides conclusions and directions for future research.

• *The authors are with the Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece. E-mail: {mtitsias, arly}@cs.uoi.gr.*

Manuscript received 4 May 2001; revised 29 Apr. 2002; accepted 23 July 2002. Recommended for acceptance by C. Brodley.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 114096.

2 Class Mixture Densities with Constrained Component Sharing

The class conditional density model (1) can be modified to allow for a *subset* of the total mixture components M to be used by each class conditional model. To achieve this, we introduce an $M \times K$ matrix Z of indicator variables z_{jk} defined as follows:

$$z_{jk} = \begin{cases} 1 & \text{if component } j \text{ can represent data of class } C_k \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In order to avoid situations where some mixture components are not used by any class density model or a class density contains no components, we assume that every row and column of a valid Z matrix contains at least one unit element. A way to introduce the constraints z_{jk} to model (1) is by imposing constraints to the mixing coefficients, i.e., by setting the parameter π_{jk} constantly equal to zero in the case where $z_{jk} = 0$. In such a case, the conditional density of a class C_k can still be considered that it is described by (1), but with the original parameter space confined to a subspace specified by the constraints indicated by the Z matrix, that is

$$p(x|C_k; z_k, \pi_k, \theta) = \sum_{j=1}^{M} \pi_{jk} p(x|j; \theta_j) = \sum_{j:z_{jk}=1} \pi_{jk} p(x|j; \theta_j), \quad (4)$$

where z_k denotes the kth column of Z and $\{j : z_{jk} = 1\}$ denotes the set of values of j for which $z_{jk} = 1$. Clearly, the common components model is a special Z-model with $z_{jk} = 1$ for all j, k, while the separate mixtures model is a special Z-model with exactly one unit element in each row of the Z matrix.

Consider now, a training set X of labeled data that are partitioned according to their class labels into K independent subsets X_k, $k = 1, \ldots, K$. If Θ denotes the set of all parameters excluding the Z values, then training can be performed by maximizing the log likelihood

$$L(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log p(x|C_k; z_k, \pi_k, \theta). \quad (5)$$

using the EM algorithm (Appendix A) and for fixed values of the Z matrix.

Let us examine now the usefulness of the above model compared with the common components and separate mixtures model. Since the common components model is the most broad Z-model, it is expected to provide the highest log likelihood value (5) and, in some sense, better data representation from the density estimation viewpoint. However, this is not always the best model if our interest is to obtain efficient classifiers. As mentioned in the introduction, the common components model can benefit from possible common characteristics of differently labeled data and lead to a reduction in the number of model parameters [16]. In the case of Gaussian components, this usually happens when common components represent subspaces with *high overlap* among differently labeled data and the obtained representation is efficient from the classification viewpoint. Nevertheless, there are problems where a common component represents differently labeled data in a less efficient way from a classification perspective as, for example, when a common Gaussian component is placed on boundary between two weakly overlapped regions with data of different classes. In this case, the separate mixtures model provides a more discriminative representation of the data. Consequently, we wish to choose the Z values so that common components are employed only for highly overlapped subspaces. Fig. 1 displays a data set where, for a fixed number of components M, a certain Z model leads to a superior classifier compared to the common components and separate mixtures case.

Consequently, a method is needed to estimate an appropriate Z matrix for a given dataset and total number of components M. Once the Z matrix has been specified, we can proceed to obtain a maximum-likelihood solution using the EM algorithm for fixed values of Z. Such a method is presented in the following section.

3 Training and Z Matrix Estimation

It is computationally intractable to investigate all possible Z-models in order to find an efficient classifier in the case of large values of M and K. Our training method initially assumes that the class conditional densities follow the broad model (1) and iteratively adjusts a soft form of the Z matrix. In particular, we maximize an objective function which is a regularized form of the log likelihood corresponding to the common components model.

We define the constraint parameters r_{jk}, where $0 \leq r_{jk} \leq 1$, and for each j satisfy:

$$\sum_{k=1}^{K} r_{jk} = 1. \quad (6)$$

The role of each parameter r_{jk} is analogous to z_{jk}: they specify the degree at which the component j is allowed to be used for modeling data of class C_k. However, unlike the mixing coefficients (2), these parameters sum to unity for each j.

The r_{jk} parameters are used to define the following functions:

$$\varphi(x; C_k, r_k, \pi_k, \theta) = \sum_{j=1}^{M} r_{jk} \pi_{jk} p(x|j; \theta_j) \quad k = 1, \ldots, K. \quad (7)$$

Equation (7) is an extension of (1) with special constraint parameters r_{jk} incorporated in the linear sum. As it will become clear shortly, for each j the parameters r_{jk} express the competition between classes concerning the allocation of the mixture component j.

If for a constraint parameter holds $r_{jk} = 0$, then, by definition we set the corresponding prior $\pi_{jk} = 0$. However, in order for the constraint (2) to be satisfied, there must be at least one $r_{jk} > 0$ for each k. The functions φ in general do not constitute densities with respect to x due to the fact that $\int \varphi(x; C_k, r_k, \pi_k, \theta) dx \leq 1$, unless the constraints r_{jk} are assigned zero-one values (in this special case each function $\varphi(x; C_k, r_k, \pi_k, \theta)$ is identical to the corresponding $p(x|C_k; \pi_k, \theta)$) in which case they coincide with z_{jk} constraints. However, it generally holds that $\varphi(x; C_k, r_k, \pi_k, \theta) \geq 0$ and $\int \varphi(x; C_k, r_k, \pi_k, \theta) dx > 0$.

Using the above function, we introduce an objective function analogous to the log-likelihood function as follows:

$$L(\Theta, r) = \sum_{k=1}^{K} \sum_{x \in X_k} \log \varphi(x; C_k, r_k, \pi_k, \theta), \quad (8)$$

where r denotes all the r_{jk} parameters. Through the maximization of the above function, we adjust the values of the r_{jk} variables (actually the degree of component sharing) and this automatically influences the solution for the class density models parameter vector Θ. The EM algorithm [5] can be used for maximizing the objective function (8). Note that the above objective function is not a log likelihood, however, it can be considered as the logarithm of an unnormalized likelihood.

At this point, it would be useful to write the update equations (provided in Appendix B) for the priors π_{jk} and the constraints r_{jk} in order to provide insight on the way the algorithm operates:

$$\pi_{jk}^{(t+1)} = \frac{1}{|X_k|} \sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)}), \quad (9)$$

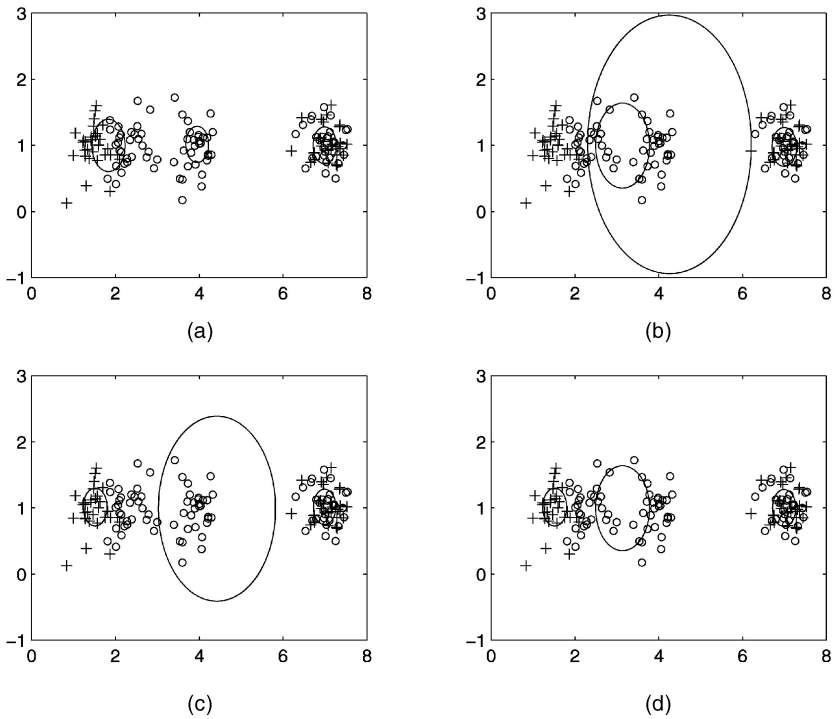

Fig. 1. The data of each class was drawn according to $p(x|C_1) = 0.33N([2.3\ 1]^T, 0.08) + 0.33N([4\ 1]^T, 0.08) + 0.33N([7\ 1]^T, 0.08)$ and $p(x|C_2) = 0.5N([1.5\ 1]^T, 0.08) + 0.5N([7\ 1]^T, 0.08)$, while the prior class probabilities were $P(C_1) = P(C_2) = 0.5$. Two data sets were generated; one for training and one for testing, and for each model we found the maximum-likelihood estimate. The generalization error e and the final log-likelihood value L computed for the four models are: (a) Common components: $e = 33.33$ percent and $L = -1754.51$, (b) separate mixtures (two components for C_1 and one for C_2): $e = 24.33$ percent $L = -2683.25$, (c) separate mixtures (one component for C_1 and two for C_2): $e = 34$ percent and $L = -3748.42$, and (d) one component common and the other two separate (one per class): $e = 21.67$ percent and $L = -1822.53$.

and

$$r_{jk}^{(t+1)} = \frac{\sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)})}{\sum_{i=1}^{K} \sum_{x \in X_i} \Phi_j(x; C_i, r_i^{(t)}, \pi_i^{(t)}, \theta^{(t)})}, \tag{10}$$

where

$$\Phi_j(x; C_k, r_k, \pi_k, \theta) = \frac{r_{jk}\pi_{jk}p(x|j;\theta_j)}{\sum_{i=1}^{M} r_{ik}\pi_{ik}p(x|i;\theta_i)}. \tag{11}$$

Using (9), (10) can be written as

$$r_{jk}^{(t+1)} = \frac{\pi_{jk}^{(t+1)}|X_k|}{\sum_{i=1}^{K} \pi_{ji}^{(t+1)}|X_i|}. \tag{12}$$

The above equation illustrates how the r_{jk} variables are adjusted at each EM iteration with respect to the newly estimated prior values π_{jk}. If we assume that the classes have nearly the same number of available training points, then, during training each class C_k is constrained to use a component j to a degree specified by the ratio of the corresponding prior value π_{jk} over the sum of the rest of the priors associated with the same component j. In this way, the more a component j represents data of class C_k, i.e., the higher the value of π_{jk}, the greater the new value of r_{jk}, which causes in the next iteration, the value of π_{jk} to become even higher (due to (9) and (11)) etc. This explains how the competition among classes for component allocation is realized through the adjustment of the constraints r_{jk}. According to this competition, it is less likely for a component to be placed at some decision boundary since in such a case the class with more data in this region will attract the component toward its side. On the other hand, the method does not seem to significantly influence the advantage of the common components model in highly overlapping regions. This can be explained from (11). In a region with high class overlap represented by a component j, the density $p(x|j;\theta_j)$ will essentially provide high values for data of all involved classes. Therefore, despite the fact that the constraint parameters might be higher for some classes, the Φ_j value (11) will still be high for data of all involved classes.

To apply the EM algorithm, the component parameters are initialized randomly from all data (ignoring class labels) and the constrains r_{jk} are initialized to $1/K$. The EM algorithm performs iterations until convergence to some locally optimal parameter point (Θ^*, r^*). Then we use the r_{jk}^* values for determine the Z matrix values:

$$z_{jk}^* = \begin{cases} 1 & \text{if } r_{jk}^* > 0 \\ 0 & \text{if } r_{jk}^* = 0. \end{cases} \tag{13}$$

The choice of Z^* is based on the argument that if $r_{jk}^* > 0$ the component j contributes to modeling of class C_k (since $\pi_{jk}^* > 0$) and, consequently, j must be incorporated in the mixture model representing C_k, the opposite holding when $r_{jk}^* = 0$. Once the Z values have been specified, then, we maximize the log-likelihood (5) applying EM and starting from parameter vector Θ^*. The final parameters Θ_f will be the estimates for the class conditional densities (4).

The above method was applied to the problem described in Section 3 (Fig. 1). The obtained solution Θ_f was exactly the one presented in Fig. 1d, where the appropriate selection for the Z-model was made in advance. Remarkably, we found that $|\Theta_f - \Theta^*| = 0.03$ with the only difference being in the values of the prior parameters of the component representing the region where the classes exhibit high overlap.

4 Experimental Results

We have conducted a series of experiments using Gaussian components and compare the common components model, the separate mixtures model, and the proposed Z-model. We tested

TABLE 1
Generalization Error and Standard Deviation Values for All Tested Algorithms and Data Sets

Clouds						
	6 components		8 components		10 components	
	error	std	error	std	error	std
Z-model	12.4	0.93	11.42	0.51	10.82	0.85
Common components	**11.12**	0.84	**11.32**	0.89	**10.42**	0.89
Separate mixtures	20.44	4.45	11.86	0.85	11.36	0.98
Satimage						
	12 components		18 components		24 components	
	error	std	error	std	error	std
Z-model	12.33	0.5	11.4	0.74	11.1	0.75
Common Components	13.23	0.56	12.28	0.79	11.52	0.75
Separate mixtures	**12.05**	0.53	**11.21**	0.75	**10.98**	0.71
Phoneme						
	10 components		12 components		14 components	
	error	std	error	std	error	std
Z-model	17.96	1.14	**17.07**	1.01	**15.85**	1.19
Common components	20.62	0.75	20.03	0.75	20.98	1.04
Separate mixtures	**17.85**	1.4	17.37	0.75	16.88	1.15
Pima Indians						
	10 components		12 components		14 components	
	error	std	error	std	error	std
Z-model	27.08	2.6	26.92	3.26	**25.94**	2.27
Common components	29.95	3.06	28.12	2.21	28.25	1.97
Separate mixtures	**26.69**	3.58	**26.43**	1.34	27.08	2.22
Ionosphere						
	8 components		10 components		12 components	
	error	std	error	std	error	std
Z-model	**11.11**	2.3	**8.55**	2.4	**9.13**	3.92
Common component	15.11	3.85	9.41	3.35	9.27	3.21
Separate mixtures	11.82	1.89	12.24	3.77	9.39	3

several well-known classification data sets and also varied the total number of components. Some of the examined data sets (for example, the Clouds data set) exhibit regions with significant class overlap, some other data sets contain regions with small class overlap, while the rest of them contain regions of both types. We expect the Z-model to be adapted to the geometry of the data and use common components only for representing data subspaces with high overlap among classes.

We considered five well-known data sets, namely the Clouds, Satimage and Phoneme from the ELENA database, and the Pima Indians and Ionosphere from the UCI repository [3]. For each data set, number of components, and model type, we employed the 5-fold cross-validation method in order to obtain an estimate of the generalization error. In the case of separate mixtures, we considered an equal number of components M/K used for the density model of each class.

Table 1 displays performance results (average generalization error and its standard deviation) for the five data sets and several choices of the total number of components. The experimental results clearly indicate that: 1) depending on the geometry of the data set and the number of available components either the common components model or the separate mixtures model may outperform one another and 2) in all data sets the Z-model either outperforms the other models or exhibits performance that is very close to the performance of the best model. It must be noted that there was no case with the performance of the Z-model being inferior to both other models. This illustrates the capability of the proposed model and training algorithm to be adapted to the geometry of each problem.

5 Conclusions and Future Research

We have generalized mixture model classifiers by presenting a model that constrains component density models to be shared among subsets of the classes. For a given total number of mixture components that must represent the data learning, the above constraints leads to improved classification performance. The objective function that is optimized for learning the constraints can be considered as a regularized log likelihood. Clearly, the regularization scheme is not a statistical one, since we do not apply Bayesian regularization. The current training method works well in practice and, additionally, all the parameters that must be specified are easily learned through EM. However, in the future, we wish to examine the suitability of more principled methods such as the method proposed in [4]. Finally, it must be noted that, in this work, we do not address the problem of assessing the optimal number of components M and we consider it as our main future research direction. To address this issue, our method needs to be adapted and combined with several well-known methodologies and criteria for model selection [11], [12].

Appendix A

EM Algorithm for a Z-Model

The Q function of the EM algorithm is

$$Q(\Theta;\Theta^{(t)})=\sum_{k=1}^{K}\sum_{x\in X_k}\sum_{j:z_{jk}=1}P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})\log\{\pi_{jk}p(j|x;\theta_j)\}. \tag{14}$$

If we assume that the mixture components are Gaussians of the general form

$$p(x|j;\mu_j,\Sigma_j)=\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}}\exp\left\{-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right\}, \quad (15)$$

then, the maximization step gives the following update equations:

$$\mu_j^{(t+1)}=\frac{\sum_{k:z_{jk}=1}\sum_{x\in X_k}P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})x}{\sum_{k:z_{jk}=1}\sum_{x\in X_k}P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})}, \quad (16)$$

$$\Sigma_j^{(t+1)}=\frac{\sum_{k:z_{jk}=1}\sum_{x\in X_k}P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})(x-\mu_j^{(t+1)})(x-\mu_j^{(t+1)})^T}{\sum_{k:z_{jk}=1}\sum_{x\in X_k}P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})}, \quad (17)$$

for $j=1,\ldots,M$ and

$$\pi_{jk}^{(t+1)}=\frac{1}{|X_k|}\sum_{x\in X_k}P\left(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)}\right), \quad (18)$$

for all j and k such that $z_{jk}=1$.

APPENDIX B
EM ALGORITHM FOR LEARNING THE Z MATRIX

The objective function to be maximized is

$$L(\Theta,r)=\log P(X;\Theta,r)=\log\prod_{k=1}^{K}\prod_{x\in X_k}\sum_{j=1}^{M}r_{jk}\pi_{jk}p(x|j;\theta_j). \quad (19)$$

As noted in Section 4, since $P(X;\Theta,r)$ does not correspond to probability density (with respect to X), the objective function can be considered as an unnormalized "incomplete data log-likelihood." Now, the EM framework for maximizing (19) is completely analogous to the mixture model case. The expected complete data log likelihood is given by

$$Q\left(\Theta,r;\Theta^{(t)},r^{(t)}\right)=\sum_{k=1}^{K}\sum_{x\in X_k}\sum_{j=1}^{M}\Phi_j\left(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)}\right)\log\{r_{jk}\pi_{jk}p(x|j;\theta_j)\}. \quad (20)$$

In the M-step, the maximization of the above function provides the following update equations:

$$\mu_j^{(t+1)}=\frac{\sum_{k=1}^{K}\sum_{x\in X_k}\Phi_j(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)})x}{\sum_{k=1}^{K}\sum_{x\in X_k}\Phi_j(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)})}, \quad (21)$$

$$\Sigma_j^{(t+1)}=\frac{\sum_{k=1}^{K}\sum_{x\in X_k}\Phi_j(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)})(x-\mu_j^{(t+1)})(x-\mu_j^{(t+1)})^T}{\sum_{k=1}^{K}\sum_{x\in X_k}\Phi_j(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)})}, \quad (22)$$

$$\pi_{jk}^{(t+1)}=\frac{1}{|X_k|}\sum_{x\in X_k}\Phi_j(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)})\quad k=1,\ldots,K. \quad (23)$$

$$r_{jk}^{(t+1)}=\frac{\sum_{x\in X_k}\Phi_j(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)})}{\sum_{i=1}^{K}\sum_{x\in X_i}\Phi_j(x;C_i,r_i^{(t)},\pi_i^{(t)},\theta^{(t)})}, \quad (24)$$

where $j=1,\ldots,M$ and $k=1,\ldots,K$.

REFERENCES

[1] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford Univ. Press, 1995.
[2] L.R. Bahl, M. Padmanabhan, D. Nahamoo, and P.S. Gopalakrishnan, "Discriminative Training of Gaussian Mixture Models for Large Vocabulary Speech Recognition Systems," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, 1996.
[3] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases." Dept. of Computer and Information Sciences, Univ. of California, Irvine, 1998.
[4] M. Brand, "An Entropic Estimator for Structure Discovery," *Neural Information Processing Systems 11*, 1998.
[5] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B*, vol. 39, pp. 1-38, 1977.
[6] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[7] Z. Ghahramani and M.I. Jordan, "Supervised Learning From Incomplete Data Via an EM Approach," *Neural Information Processing Systems 7*, Mass.: MIT Press, D.J. Cowan, G. Tesauro, and J. Alspector, eds., vol. 6, pp. 120-127, 1994.
[8] T.J. Hastie and R.J. Tibshirani, "Discriminant Analysis by Gaussian Mixtures," *J. Royal Statistical Soc. B*, vol. 58, pp. 155-176, 1996.
[9] T. Jebara and A. Pentland, "Maximum Conditional Likelihood Via Bound Maximization and the CEM Algorithm," *Neural Information Processing Systems 11*, M. Kearns, S. Solla, and D. Cohn, eds., 1998.
[10] M.I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation*, vol. 6, pp. 181-214, 1994.
[11] P. Kontkanen, P. Myllymaki, and H. Tirri, "Comparing Presequential Model Selection Criteria in Supervised Learning of Mixture Models," *Proc. Eighth Int'l Workshop Artificial Intelligence and Statistics*, T. Jaakola and T. Richardson, eds., pp. 233-238, 2001.
[12] G.J. McLachlan and D. Peel, *Finite Mixture Models*. Wiley, 2000.
[13] D.J. Miller and H.S. Uyar, "A Mixture of Experts Classifier With Learning Based on Both Labeled and Unlabeled Data," *Neural Information Processing Systems 9*, Mass.: MIT Press, M.C. Mozer, M.I. Jordan, and T. Petsche, eds., 1996.
[14] R. Redner and H. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195-239, 1984.
[15] L.K. Saul and D.D. Lee, "Multiplicative Updates for Classification by Mixture Models," *Neural Information Processing Systems 14*, S. Becker, T. Dietterich, and Z. Ghahramani, eds., 2002.
[16] M. Titsias and A. Likas, "Shared Kernel Models for Class Conditional Density Estimation," *IEEE Trans. Neural Networks*, vol. 12, no. 5, pp. 987-997, Sept. 2001.
[17] D.M. Titterington, A.F. Smith, and U.E. Makov, *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.