LETTER **Communicated by Steven Nowlan**

Mixture of Experts Classification Using a Hierarchical Mixture Model

Michalis K. Titsias
mtitsias@cs.uoi.gr
Aristidis Likas
arly@cs.uoi.gr
Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece

A three-level hierarchical mixture model for classification is presented that models the following data generation process: (1) the data are generated by a finite number of sources (clusters), and (2) the generation mechanism of each source assumes the existence of individual internal class-labeled sources (subclusters of the external cluster). The model estimates the posterior probability of class membership similar to a mixture of experts classifier. In order to learn the parameters of the model, we have developed a general training approach based on maximum likelihood that results in two efficient training algorithms. Compared to other classification mixture models, the proposed hierarchical model exhibits several advantages and provides improved classification performance as indicated by the experimental results.

1 Introduction

A widely applied method for implementing the Bayes classifier is based on obtaining the posterior probabilities of class membership through the estimation of the class prior probabilities and the class conditional densities (Duda & Hart, 1973). This is a generative approach to classification since a model of the joint distribution of the input data and the class labels is provided. The computationally intensive part of the design of such classifiers concerns the estimation of the class conditional densities. The widely used way to obtain these estimates is independently to apply density estimation methods to each class-labeled data set. However, such an approach does not benefit from the existence of any common characteristics among data of different classes. For example, the data may arise from differently labeled clusters that are located in overlapping regions in the data space.

A very general assumption about data generation in a classification problem which can benefit from the existence of common characteristics among differently labeled data is the following: the data are drawn from a finite number of sources (clusters), and within each cluster, the data are generated by labeled sources that form subclusters of the parent cluster. These generation assumptions can be efficiently modeled by a three-level hierarchical mixture model (Bishop & Tipping, 1998). The first generation assumption is represented at the second level of the hierarchical mixture, and typically the number of components are unknown and must be inferred by the data. However, at the third level of the hierarchical mixture, where the second assumption is represented, each submixture (associated with a specific parent component) has precisely as many components as the number of classes. We refer to the above model as the hierarchical mixture classification model. In order to learn the parameters of the model, we derive a general training approach based on the maximum likelihood framework that results in two fast training algorithms.

The proposed model can be considered as a mixture of experts classifier. Mixtures of experts (Jacobs, Jordan, Nowlan, & Hinton, 1991; Jordan & Jacobs, 1994) are general models for estimating conditional distributions. Typically, these models comprise a gating network that divides the problem into smaller problems and expert networks that solve each subproblem. In our case, both the gating network units and the specialized experts are suitably defined from the hierarchical mixture.

An additional feature of the hierarchical mixture classifier is that it provides class conditional density estimates as flat mixtures.[1] Consequently, it is possible to compare the method directly with two well-known class conditional density estimation techniques based on mixture models. The first is the well-known approach that employs a separate mixture (having its own components) for representing each class conditional density (McLachlan & Peel, 2000). This is the most widely used method and has been studied by Hastie and Tibshirani (1996). The second approach is to assume that the class conditional densities are modeled by mixtures having common mixture components (Ghahramani & Jordan, 1994; Miller & Uyar, 1996; Titsias & Likas, 2001). The last is actually similar to using a radial basis function (RBF) or an RBF-like neural network for solving classification problems. This is further investigated in Miller and Uyar (1998), where the Bayes decision function of a classifier that estimates the class conditional densities by mixtures with common components is shown to be equivalent to the decision function of an RBF classifier. In the following, we will refer to the first approach as the *separate mixtures model* and to the second as the *common components model*. The hierarchical mixture classifier can be thought of as being an extended and more flexible version of the common components model. In addition, the proposed model can be also considered as a constrained case of a separate mixtures model that employs a certain number of components. The proposed model exhibits several advantages over these methods as illustrated in sections 3.2 and 3.3.

[1] We use the term *flat* mixture to refer to the usual mixture density model of the form $p(x) = \sum_{j=1}^{M} \pi_j p(x \mid j, \theta_j)$, which does not exhibit any hierarchical structure.

Section 2 provides a unifying description of classification techniques based on mixture models. In Section 3, the proposed hierarchical mixture classification model is described along with a training approach based on maximum likelihood. In addition, we provide illustrative comparisons of the proposed method with the common components and the separate mixtures model. Conclusions drawn from these comparisons are also supported experimentally in section 4, where comparative performance results are presented for several well-known data sets. Finally, section 5 provides conclusions and future research directions.

2 Bayes Classification Based on Mixtures

Consider a classification problem with K classes C_k, $k = 1, \ldots, K$. The Bayes classifier decides about the class of a data point x by selecting the class label C_k with the highest posterior probability value $P(C_k \mid x)$. Using the Bayes rule, the posterior probability $P(C_k \mid x)$ is written as

$$P(C_k \mid x) = \frac{p(x \mid C_k) P(C_k)}{\sum_{\ell=1}^{K} p(x \mid C_\ell) P(C_\ell)}, \tag{2.1}$$

where $P(C_k)$ is the class prior probability and $p(x \mid C_k)$ the corresponding class conditional density. Each class conditional density $p(x \mid C_k)$ is estimated by applying density estimation methods using the available data. In the following, we provide a brief unifying description of some existing methods for estimating the class conditional densities using mixtures.

We assume that the data have been generated by M sources (or clusters), and these clusters can be modeled by the densities $p(x \mid j, \theta_j)$, $j = 1, \ldots, M$, with θ_j denoting the corresponding parameter vector. We further suppose that only some of the clusters can generate data of the class C_k; thus, only a subset T_k of the density models is responsible for generating the data of class C_k. Consequently, the C_k-class conditional density can be modeled as the following mixture,

$$p(x \mid C_k, \Theta_k) = \sum_{j \in T_k} \pi_{jk} p(x \mid j, \theta_j), \tag{2.2}$$

where the parameter π_{jk} represents the probability $P(j \mid C_k)$ and Θ_k is the total parameters corresponding to class C_k. We assume that any two different subsets T_k and T_ℓ (corresponding to classes C_k and C_ℓ) may contain common elements, that is, in general, $T_k \cap T_\ell \neq \emptyset$. The latter implies that the data of different classes may have been generated from some common data sources. According to equation 2.2, it is clear that once we know the component j from which a data point x has been drawn, then x is independent of class C_k, that is, $p(x \mid j) = p(x \mid j, C_k)$.

The above choice of the class conditional densities provides as special cases two well-known approaches. The first is the separate mixtures model, and its basic property is that the data of each class are a priori assumed to be generated by clusters that are not common with clusters corresponding to differently labeled data. This model results from equation 2.2 if the sets T_k, $k = 1, \ldots, K$ are such that $T_k \cap T_\ell = \emptyset$ for all $k \neq \ell$. The separate mixtures model constitutes a widely used method for designing a Bayes classifier, and it has been theoretically studied in Hastie and Tibshirani (1996) in the case of gaussian mixture components. An alternative approach, the common components model, assumes that all data may arise from any of the M clusters and results from equation 2.2 by assuming that $T_k = \{1, \ldots, M\}$ for each k (Ghahramani & Jordan, 1994; Miller & Uyar, 1996; Titsias & Likas, 2001). Clearly, the common components model exhibits generality over the separate mixtures and also over all possible models described by equation 2.2.

To classify a new data point x based on the Bayes formula 2.1, the class prior probabilities $P(C_k)$ are also needed, which are represented by introducing the parameters P_k. Training can be performed based on maximum likelihood. Assume that we have a set (X, Y) of labeled data where X is the set of data points and Y the corresponding class labels. The original data set X can be partitioned according to the class labels into K disjoint subsets X_k, $k = 1, \ldots, K$. Learning the whole parameter vector Θ can be performed by maximizing the following log likelihood $L(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log P_k \, p(x \mid C_k, \Theta_k)$:

$$\begin{aligned} L(\Theta) &= \sum_{k=1}^{K} |X_k| \log P_k + \sum_{k=1}^{K} \sum_{x \in X_k} \log \sum_{j \in T_k} \pi_{jk} p(x \mid j, \theta_j) \\ &= \sum_{k=1}^{K} |X_k| \log P_k + \sum_{k=1}^{K} L_k(\Theta_k), \end{aligned} \tag{2.3}$$

where L_k is the class log likelihood corresponding to the subset X_k. Maximization of the first term in equation 2.3 gives $P_k = \frac{|X_k|}{|X|}$, while maximization of the second term would provide estimates of the class conditional densities. Note that the latter maximization in the case of the separate mixtures approach splits into K independent problems, each one involving a class log likelihood L_k. Clearly, the same does not hold for the common components approach since the parameters of all components appear in each L_k.

Let F_j be the subset of all classes C_k for which the data can arise from the component j ($j \in T_k$). To find out which is the generation process for a pair (x, C_k), we need to express the joint distribution of x and C_k. It holds that $p(x, C_k \mid \Theta) = P_k \sum_{j=1}^{M} \pi_{jk} p(x \mid j, \theta_j)$ (where for all $j \notin T_k$, we assume $\pi_{jk} = 0$) and since $P_k \pi_{jk} = P(j \mid \Theta) P(C_k \mid j, \Theta)$ (where $P(j \mid \Theta) = \sum_{k \in F_j} \pi_{jk} P_k$ and

$P(C_k \mid j, \Theta) = \frac{P_k \pi_{jk}}{P(j|\Theta)}$), we obtain:

$$p(x, C_k \mid \Theta) = \sum_{j=1}^{M} P(j \mid \Theta) P(C_k \mid j, \Theta) p(x \mid j, \theta_j). \tag{2.4}$$

Based on this expression we may assume that the labeled data are generated as follows:

- Select a component j from the set $\{1, \ldots, M\}$ with probability $P(j \mid \Theta)$.
- Select a class label C_k, where $k \in F_j$, with probability $P(C_k \mid j, \Theta)$, and draw x from density $p(x \mid j, \theta_j)$.

The generative model for the separate mixtures and common components model is obtained as a special case. More specifically, in the separate mixtures case, the selection of a component j automatically specifies the class of x since in this case the set F_j contains only one element. On the contrary in the common components case, each F_j contains all classes, and the class label is selected among by all possible values. According to the second point above, once the component j has been selected, the label C_k and the data point x are independently specified. Actually, x and C_k are conditionally independent given the component variable j.

Finally, if we are interested in the unconditional density of x, this is given by $p(x \mid \Theta) = \sum_{j=1}^{M} P(j \mid \Theta) p(x \mid j, \theta_j)$, which clearly is a flat mixture. In the next section, we present a classification model that estimates the unconditional density of x by a hierarchical mixture.

3 The Hierarchical Mixture Classification Model

We wish to define a generative model realizing the following two assumptions: (1) the data are generated by M clusters and (2) within each cluster, the data are generated by class-labeled sources that form subclusters of the larger cluster. If a subcluster corresponding to class C_k can be modeled by the density $p(x \mid C_k, j, \theta_{kj})$ (where θ_{kj} are the corresponding parameters), then the unconditional density of x can be given by the following three-level hierarchical mixture model (Bishop & Tipping, 1998) illustrated in Figure 1,

$$p(x \mid \Theta) = \sum_{j=1}^{M} \pi_j \sum_{k=1}^{K} P_{kj} p(x \mid C_k, j, \theta_{kj}), \tag{3.1}$$

where the parameter π_j represents the probability $P(j)$, P_{kj} the probability $P(C_k \mid j)$, and Θ denotes the whole set of model parameters.

Clearly the second level of the hierarchical mixture (see Figure 1) provides information on how the data are generated by the M components ignoring the class labels. In this level, each component density is obtained

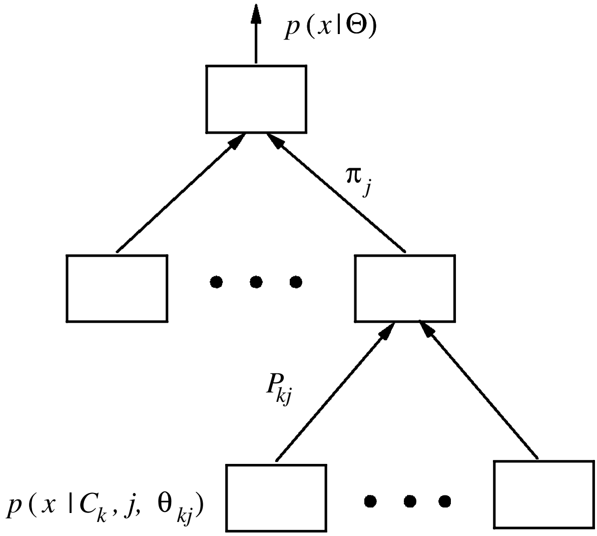

Figure 1: Estimation of the unconditional density of x by the hierarchical mixture classification model.

by marginalizing out the class labels—$p(x \mid j, \Theta) = \sum_{k=1}^{K} P_{kj} p(x \mid C_k, j, \theta_{kj})$. At the third level of the hierarchy, information is provided about the data along with their class labels. Note that since we have K classes, K subcomponents correspond to each component j of the second level.

We are particularly interested in exploiting the use of this model for solving classification problems. Therefore, the posterior probabilities of class membership $P(C_k \mid x)$ must be computed:

$$P(C_k \mid x, \Theta) = \frac{\sum_{j=1}^{M} \pi_j P_{kj} P(x \mid C_k, j, \theta_{kj})}{p(x \mid \Theta)}. \tag{3.2}$$

Although the above expression results directly by the model, an equivalent and more useful expression is

$$P(C_k \mid x, \Theta) = \sum_{j=1}^{M} P(j \mid x, \Theta) P(C_k \mid x, j, \Theta), \tag{3.3}$$

where

$$P(j \mid x, \Theta) = \frac{\pi_j p(x \mid j, \Theta)}{p(x \mid \Theta)} \tag{3.4}$$

and

$$P(C_k \mid x, j, \Theta) = \frac{P_{kj} p(x \mid C_k, j, \theta_{kj})}{p(x \mid j, \Theta)}. \tag{3.5}$$

Expression 3.3 explicitly denotes that the model estimates the posterior $P(C_k \mid x)$ as a mixture of experts model. The mixture of experts network was originally introduced in Jacobs et al. (1991) and extended to a hierarchical structure in Jordan and Jacobs (1994). A mixture of experts network consists of several expert models that estimate the input-dependent distribution of the output in different regions of the input space. The output of the model is computed using an input-dependent gating network that probabilistically combines the estimates of the experts. In our case, the gating network units correspond to $P(j \mid x, \Theta)$ provided by equation 3.4, while the estimates of the experts correspond to the locally computed posterior probabilities of class membership $P(C_k \mid x, j, \Theta)$ provided by equation 3.5.

Several useful quantities such as the class prior probability and the class conditional density can be easily expressed as

$$P(C_k \mid \Theta) = \sum_{j=1}^{M} P_{kj} \pi_j \tag{3.6}$$

and

$$p(x \mid C_k, \Theta) = \sum_{j=1}^{M} P(j \mid C_k, \Theta) p(x \mid C_k, j, \theta_{kj}), \tag{3.7}$$

respectively, where

$$P(j \mid C_k, \Theta) = \frac{P_{kj} \pi_j}{P(C_k \mid \Theta)}. \tag{3.8}$$

According to the hierarchical mixture classification model, the generation of a data pair (x, C_k) proceeds as follows:

- Select a component from the set $\{1, \ldots, M\}$ with probability π_j.
- Select a class label C_k, where $k \in \{1, \ldots, K\}$, with probability P_{kj}, and then draw x according to the probability density $p(x \mid C_k, j, \theta_{kj})$.

Note that according to equation 3.7, each class-conditional density exhibits a flat mixture form. This suggests that we can contrast the proposed model against the mixture model classifiers described in section 2. In the hierarchical mixture classifier case, the class label C_k and the data x are not conditionally independent given the component j, which yields the latter to be in principle different from the other classification mixture models.

In addition, the hierarchical mixture classification model with M components in its second level can be considered as an extension of the common components model that employs M components in total. Particularly, the common components model assumes that all data points generated by the component j, and possibly corresponding to different classes, are explained by the same density model $p(x \mid j, \theta_j)$. In contrast, the hierarchical mixture classification model assumes that the data generated by the component j are explained in a way that depends on their class labels (for each class C_k, a different probability model $p(x \mid C_k, j, \theta_{kj})$ is provided). In section 3.2 we explain how this additional flexibility resolves a serious data representation drawback of the common components model and improves classification performance significantly. Compared to the separate mixtures model, the proposed model can be derived by setting special constraints to a separate mixtures model with KM components in total in which each class conditional density is modeled by a mixture with M components. The imposed constraints are that M groups of K components (each belonging to different class conditional densities) must be formed, and the components of each group must explain a common input subspace as discussed in section 3.

3.1 Training the Hierarchical Mixture Classification Model.

In the following, we assume that all the probability models $p(x \mid C_k, j, \theta_{kj})$ follow the same parametric form taken from the exponential family. The log likelihood of the labeled data set (X, Y) is

$$L(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log \sum_{j=1}^{M} \pi_j P_{kj} p(x \mid C_k, j, \theta_{kj}). \tag{3.9}$$

It is possible to maximize the above quantity using the expectation-maximization (EM) algorithm. However, such a maximization would cause the whole model to collapse to one equivalent to a separate mixtures model (with M components employed by each class conditional density model), which means that hierarchy is lost. Therefore, in order to maintain the hierarchical nature of the model, we cannot rely on direct optimization of the above log likelihood.

According to the assumption of the hierarchical mixture classification model, the missing information is related to the way that the data points are generated by the components of the second level. On the other hand, there is no missing information in the third level of the hierarchy (where class labels are taken into account), and the probability model that generated a data point is explicitly indicated by its class label. In order to express the second-level missing information, we introduce for each x an M-dimensional binary vector $z(x)$ indicating the component that generated x. The resulting complete data log likelihood is

$$L_C(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j=1}^{M} z_j(x) \log \pi_j P_{kj} p(x \mid C_k, j, \theta_{kj}). \tag{3.10}$$

However, since each variable $z(x)$ is unknown, we should expect to employ only an approximation of $z_j(x)$ provided by its expected value. In our case, two methods exist to obtain the expected value of $z_j(x)$. In the first method, class labels are ignored, and the expected value of $z_j(x)$ is equal to the probability $P(j \mid x)$. The second type of expectation takes into account the class label C_k of x and corresponds to the probability $P(j \mid x, C_k)$.[2] If $h_j(x)$ denotes either $P(j \mid x)$ or $P(j \mid x, C_k)$, then $\sum_{j=1}^{M} h_j(x) = 1$, and the expected value of the complete data log likelihood L_C is

$$Q(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j=1}^{M} h_j(x) \log \pi_j P_{kj} p(x \mid C_k, j, \theta_{kj}). \tag{3.11}$$

In analogy to the case of unsupervised hierarchical mixture training (Bishop & Tipping, 1998), we consider that $h_j(x)$ have been computed in a previous stage and remain constant. In this case, the maximization of Q with respect to the parameters Θ yields

$$\hat{\pi}_j = \frac{1}{|X|} \sum_{k=1}^{K} \sum_{x \in X_k} h_j(x) \tag{3.12}$$

$$\hat{P}_{kj} = \frac{\sum_{x \in X_k} h_j(x)}{\sum_{\ell=1}^{K} \sum_{x \in X_\ell} h_j(x)} \tag{3.13}$$

$$\hat{\theta}_{kj} = \arg\max_{\theta_{kj}} \sum_{x \in X_k} h_j(x) \log p(x \mid C_k, j, \theta_{kj}). \tag{3.14}$$

Since $p(x \mid C_k, j, \theta_{kj})$ is chosen from the exponential family, $\hat{\theta}_{jk}$ can be analytically obtained by solving the equation

$$\sum_{x \in X_k} h_j(x) \nabla_{\theta_{kj}} \log p(x \mid C_k, j, \theta_{kj}) = 0 \tag{3.15}$$

with respect to θ_{kj}.

[2] In the first case, the expected value is $E[z_j(x) \mid X] = P(z_j(x) = 1 \mid x) = P(j \mid x)$, while in the second case, it holds that $E[z_j(x) \mid X, Y] = P(z_j(x) = 1 \mid x, C_k) = P(j \mid x, C_k)$.

In the gaussian case, the solution of equation 3.15 can be analytically obtained. Assume that each probability model $p(x \mid C_k, j, \theta_{kj})$ is a gaussian of the general form

$$p(x \mid C_k, j, \theta_{kj}) = \frac{1}{(2\pi)^{d/2}|\Sigma_{kj}|^{1/2}} \times \exp\left\{-\frac{1}{2}(x-\mu_{kj})^T\Sigma_{kj}^{-1}(x-\mu_{kj})\right\}. \tag{3.16}$$

Then the solution for each parameter vector $\theta_{kj} = \{\mu_{kj}, \Sigma_{kj}\}$ takes the form

$$\hat{\mu}_{kj} = \frac{\sum_{x\in X_k} h_j(x)x}{\sum_{x\in X_k} h_j(x)} \tag{3.17}$$

$$\hat{\Sigma}_{kj} = \frac{\sum_{x\in X_k} h_j(x)(x-\hat{\mu}_{kj})(x-\hat{\mu}_{kj})^T}{\sum_{x\in X_k} h_j(x)}. \tag{3.18}$$

Note that these two estimates are provided only if $\hat{P}_{kj} > 0$, since otherwise the component j does not represent data of the class C_k.

Obviously, in order to obtain the parameter solution described by equations 3.12 through 3.14, we must first specify the values of $h_j(x)$, that is, estimate the probabilities $P(j \mid x)$ or $P(j \mid x, C_k)$. An approximation of $P(j \mid x)$ can be obtained by running a mixture model with M components using the data set X and ignoring class labels. Similarly, an approximation of $P(j \mid x, C_k)$ can be obtained by applying the common components model to the labeled data set (X, Y). Therefore, two different approaches can be applied for obtaining an estimate of $h_j(x)$:

- *Algorithm 1: Unsupervised case* ($h_j(x) = P(j \mid x)$). We introduce the mixture model $p(x \mid \Phi) = \sum_{j=1}^{M} \pi_j p(x \mid j, \phi_j)$ where $p(x \mid j, \phi_j)$ typically has the same parametric form as $p(x \mid C_k, j, \theta_{kj})$. We maximize the log likelihood considering the unlabeled data X using the EM algorithm and obtain the parameter solution $\hat{\Phi}$ (see section A.1). Then we replace $h_j(x)$ by

$$P(j \mid x, \hat{\Phi}) = \frac{\hat{\pi}_j p(x \mid j, \hat{\phi}_j)}{\sum_{i=1}^{M} \hat{\pi}_i p(x \mid i, \hat{\phi}_i)}. \tag{3.19}$$

- *Algorithm 2: Supervised case* ($h_j(x) = P(j \mid x, C_k)$). We introduce the common components model $p(x \mid C_k, \Phi_k) = \sum_{j=1}^{M} \pi_{jk} p(x \mid j, \phi_j)$ and obtain a parameter solution $\hat{\Phi}_k$ for each k by maximizing the log likelihood 2.3 using the EM algorithm (see section A.2). Then we replace $h_j(x)$ by

$$P(j \mid x, C_k, \hat{\Phi}_k) = \frac{\hat{\pi}_{jk} p(x \mid j, \hat{\phi}_j)}{\sum_{i=1}^{M} \hat{\pi}_{ik} p(x \mid i, \hat{\phi}_i)}. \tag{3.20}$$

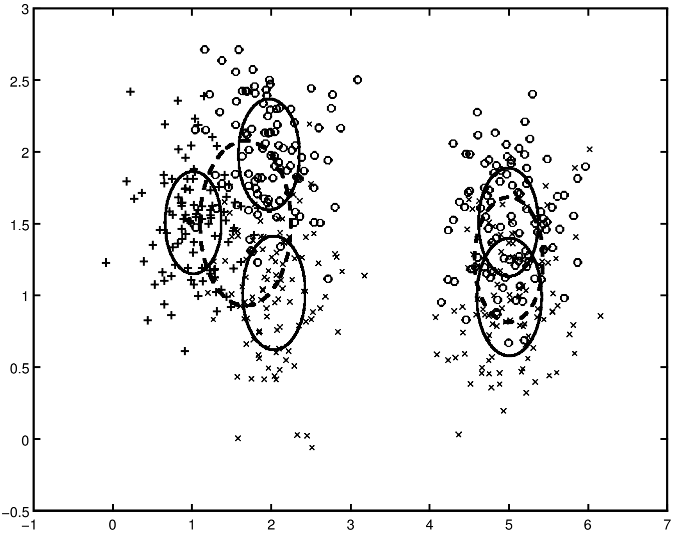

Figure 2: The two-dimensional data points of a three-class problem and the parameter solutions for the models $p(x \mid C_k, j, \theta_{kj})$, which were assumed to be gaussians (solid lines). The dashed lines represent the parameter solution obtained by the intermediate training stage (approximation of $h_j(x)$). Note that for the cluster on the right, there exist only two subclusters (solid lines). This is because this data region contains data from two classes only (+'s are missing); thus, the density model of the third class is automatically pruned.

Once we have obtained the parameter solution for the hierarchical mixture classification model, several useful quantities can be estimated. The class prior probability given by equation 3.6 would essentially be $P(C_k \mid \hat{\Theta}) = \frac{|X_k|}{|X|}$, where equations 3.12 and 3.13 are used. The class conditional density can be estimated using equation 3.7, where

$$P(j \mid C_k, \hat{\Theta}) = \frac{1}{|X_k|} \sum_{x \in X_k} h_j(x). \tag{3.21}$$

In Figure 2, a three-class data set is illustrated along with the parameter solution of the models $p(x \mid C_k, j, \theta_{kj})$ (solid lines), which were chosen to be gaussians. The model employs two components at the second level of the hierarchy. In this example, the same solution is obtained at the intermediate training stage (represented with dash lines) using either a mixture model or the common components model. Although the two algorithms provided the same parameter solutions in this example, this is not expected to hold in general. This can be explained by the fact that the application of a mixture

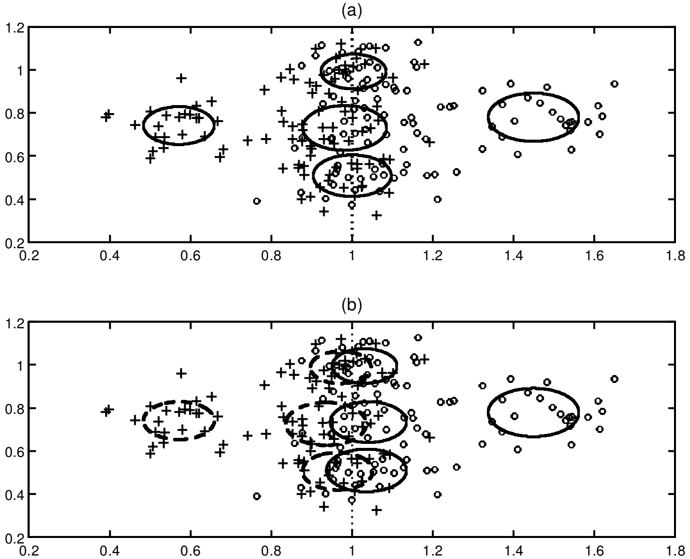

Figure 3: A two-class problem where the true decision boundary is linear (dotted line). (a) The solution found by the common components model with five components. (b) The solution of the hierarchical mixture classifier found based on the common components solution (a different line style is used for the components of each class).

model constitutes an unsupervised learning task, while the application of the common components model is a supervised task.

3.2 Comparison with the Common Components Model. The hierarchical mixture classification model is a more flexible model compared to the common components model. In this section, we investigate this issue and explain how the increased flexibility yields an improvement in data representation in regions close to the decision boundaries.

Figure 3 displays the data of an artificial classification problem of two classes. The problem is so constructed that the true decision boundary is linear corresponding to the vertical dot line in the figure. Figure 3a displays the parameter solution found by the common components model with $M = 5$. The common components model places three components on the decision boundary, thus leading to a decrease in classification performance. We also applied the hierarchical mixture classifier with $M = 5$ using algorithm 2; the solution is displayed in Figure 3b. More specifically, the hierarchical mixture classifier refines the solution found by the common components model by taking the probabilities $P(j \mid x, C_k, \hat{\Phi}_k)$ and approximating the local class-labeled clusters within the previous five clusters. As shown in Figure 3b, the final parameter solution of the model approximates quite efficiently the true decision boundary. If we compare the two solutions, we can see that they differ only in regions close to the decision boundary (i.e., in the subspace relevant to classification), while in subspaces irrelevant to classification, the representation is exactly the same.[3] Thus, the hierarchical mixture classifier (trained using algorithm 2) can be considered as taking the common components solution as input and improving it in subspaces that are critical for classification efficiency. In the following, we discuss more formally this property of the solutions found by the proposed model.

Once a hierarchical mixture classification model has been constructed using algorithm 2, it is natural to compare the two classification methods in terms of the values of the corresponding solution parameters. Assume that $\hat{\Theta}$ is the parameter solution for the hierarchical mixture classification model provided by equations 3.12 through 3.14, where $h_j(x)$ is computed as $P(j \mid x, C_k, \hat{\Phi}_k)$ obtained by the common components model with parameters $\hat{\Phi}_k$, $k = 1, \ldots, K$. We wish to compare the classifier provided by the hierarchical mixture classification model with parameters $\hat{\Theta}$ with the corresponding common components model with parameters $\hat{\Phi}_k$, $k = 1, \ldots, K$. To achieve this, it is sufficient to compare the class conditional density estimate $p(x \mid C_k, \hat{\Theta}) = \sum_{j=1}^{M} P(j \mid C_k, \hat{\Theta}) p(x \mid C_k, j, \hat{\theta}_{kj})$ with the corresponding $p(x \mid C_k, \hat{\Phi}_k) = \sum_{j=1}^{M} \hat{\pi}_{jk} p(x \mid j, \hat{\phi}_j)$.

It can be shown that the solution $p(x \mid C_k, \hat{\Theta})$ is better than $p(x \mid C_k, \hat{\Phi}_k)$ in terms of the corresponding log-likelihood values. More specifically, the following proposition holds:

Proposition 1. *Let $\hat{\Theta}$ be the parameter solution for the hierarchical mixture classification model provided by equations 3.12 through 3.14, where $h_j(x)$ is computed as $P(j \mid x, C_k, \hat{\Phi}_k)$, the solution provided by the common components model. Also assume that for each j, the density models $p(x \mid j, \phi_j)$ and $p(x \mid C_k, j, \theta_{kj})$, $k = 1, \ldots, K$ have the same parametric form, which is such that the maximum of equation 3.14 occurs for a unique value of the parameters.*

If for a class C_k it holds that $\nabla_{\Phi_k} L_k(\hat{\Phi}_k) \neq 0$, where L_k is the C_k-class log likelihood defined in equation 2.3, then the estimate $p(x \mid C_k, \hat{\Theta})$ provides higher class log-likelihood value than the estimate $p(x \mid C_k, \hat{\Phi}_k)$, that is,

$$\sum_{x \in X_k} \log \sum_{j=1}^{M} P(j \mid C_k, \hat{\Theta}) p(x \mid C_k, j, \hat{\theta}_{kj}) > \sum_{x \in X_k} \log \sum_{j=1}^{M} \hat{\pi}_{jk} p(x \mid j, \hat{\phi}_j). \tag{3.22}$$

[3] Subspaces relevant to classification are considered to be all the data clusters containing data from more than one class.

If for a class C_k it holds that $\nabla_{\Phi_k} L_k(\hat{\Phi}_k) = 0$, then the estimates $p(x \mid C_k, \hat{\Theta})$ and $p(x \mid C_k, \hat{\Phi}_k)$ are identical.[4]

The proof is given in appendix B.

Proposition 1 states that for each C_k, the estimate $p(x \mid C_k, \hat{\Theta})$ can be such that either the class log-likelihood value would be higher than the log likelihood computed using $p(x \mid C_k, \hat{\Phi}_k)$, or it would be identical to $p(x \mid C_k, \hat{\Phi}_k)$. The second case occurs when the parameter values $\hat{\Phi}_k$ locally maximize the class log-likelihood L_k, which means that the $p(x \mid C_k, \hat{\Phi}_k)$ is already a locally optimum estimate for the conditional density of class C_k. On the other hand, the assumption in the first case implies that $\hat{\Phi}_k$ does not constitute a local optimum of the class log-likelihood value L_k. The first case occurs frequently in practice. To explain this, consider that since each $\hat{\Phi}_k$ is obtained from the maximization of the log likelihood equation 2.3 (corresponding to the common components model case) using the EM algorithm (Dempster, Laird, & Rubin, 1977), we may assume that it constitutes a stationary point of the log likelihood. This implies that each $\hat{\pi}_{jk}$ and $\hat{\phi}_j$ satisfy

$$\hat{\pi}_{jk} = \frac{1}{|X_k|} \sum_{x \in X_k} P(j \mid x, C_k, \hat{\Phi}_k), \tag{3.23}$$

$$\sum_{k=1}^{K} \sum_{x \in X_k} P(j \mid x, C_k, \hat{\Phi}_k) \nabla_{\phi_j} \log p(x \mid j, \hat{\phi}_j) = 0 \tag{3.24}$$

or

$$\sum_{k=1}^{K} \nabla_{\phi_j} L_k(\hat{\Phi}_k) = 0. \tag{3.25}$$

Although $\hat{\pi}_{jk}$ will always correspond to a stationary point of the class log-likelihood L_k, equation 3.25 explicitly points out that $\hat{\phi}_j$ may not correspond to a stationary point of L_k for all k unless the component represents data of only one class. In order for $\hat{\phi}_j$ to be a stationary point of L_k, it must hold that

$$\nabla_{\varphi_j} L_k(\hat{\Phi}_k) = \sum_{x \in X_k} P(j \mid x, C_k, \hat{\Phi}_k) \nabla_{\phi_j} \log p(x \mid j, \hat{\phi}_j) = 0. \tag{3.26}$$

The situation where $\hat{\phi}_j$ satisfies equation 3.24 without satisfying equation 3.26 for every k occurs when the component j represents data of dif-

[4] We mean that $P(j \mid C_k, \hat{\Theta}) = \hat{\pi}_{jk}$ and $p(x \mid C_k, j, \hat{\theta}_{kj}) = p(x \mid j, \hat{\phi}_j)$ for each $j = 1, \ldots, M$ and x.

ferent classes that do not overlap significantly.[5] In real-world classification problems (with class overlapping), it is almost certain that there would be some ϕ_j for which the condition 3.26 will not be true for all k.[6] For such a $\hat{\phi}_j$, we expect the corresponding data subspace (represented by the component j of the common components model) to be close to the true decision boundaries. This is because the component j essentially represents data of more than one class that overlap or lie very close to each over. All these $\hat{\phi}_j$ will result in some $\nabla_{\Phi_k} L_k(\hat{\Phi}_k) \neq 0$, and the first case of proposition 1 would be applicable. Subsequently, the specific estimates $p(x \mid C_k, \hat{\Theta})$ will improve the corresponding $p(x \mid C_k, \hat{\Phi}_k)$, and this improvement will be observed in subspaces relevant to classification. The latter can be considered very beneficial from the classification point of view. An illustrative example is in Figure 3.

3.3 Comparison with the Separate Mixtures Model. As we previously noted, the hierarchical mixture classifier can be considered a constrained version of a separate mixtures model with a total of KM components (M components for each class conditional density). An important advantage over separate mixtures is the ability to adjust the number of active components automatically. More specifically, the hierarchical mixture model avoids overfitting because it is able to prune[7] class density models at the third level of hierarchy based on the distribution of the data. For this reason, the active (not pruned) number of density models can be any number in the range $[M, KM]$ and is learned from the data. For example, in the problem of Figure 2, six density models are assumed initially; however, after training, only five remain active (precisely as many as the problem requires). Similarly, in the data of Figure 3, only eight density models remain active, while originally 10 such models were assumed. These results indicate that the hierarchical mixture classifier is able to avoid overfitting by pruning density models during learning. This also makes the method to be robust with respect to the choice of M; even if M is overestimated, it is still possible to find an efficient solution. In order to verify this issue experimentally, we applied the hierarchical mixture classifier to the problem of Figure 2 for the

[5] An illustrative example is displayed in Figure 2, where each component of the common components model (dashed lines) represents simultaneously data clusters of different classes that clearly do not have the same means and variances. For this reason, the class log likelihoods are not maximum. Note that the class log likelihoods would all be simultaneously maximized if the class subclusters had precisely the same means and variances.

[6] Note that if for a specific $\hat{\phi}_j$ there exists a class C_k such that $\nabla_{\phi_j} L_k(\hat{\Phi}_k) \neq 0$, then in order for equation 3.24 to be satisfied, there must also be at least one different class C_ℓ for which $\nabla_{\phi_j} L_\ell(\hat{\Phi}_\ell) \neq 0$.

[7] If a parameter P_{kj} becomes zero (or very close to zero), the corresponding density model $p(x \mid C_k, j, \theta_{kj})$ is pruned.

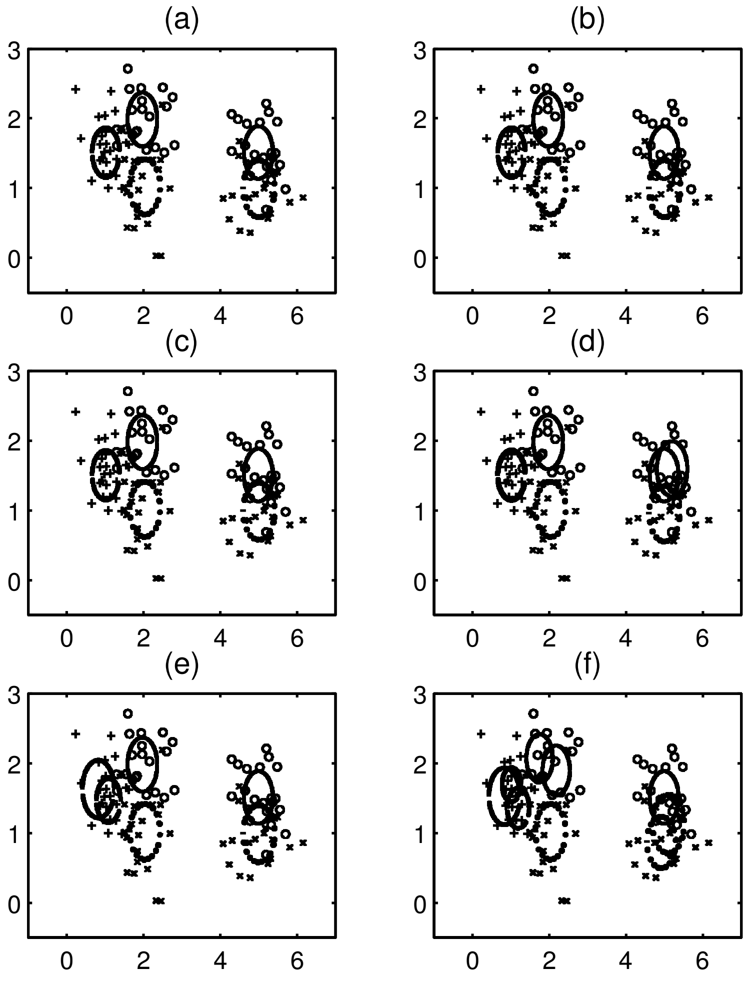

Figure 4: Displays the same data set as in Figure 2. (a–d) The solution of the hierarchical mixture classifier for $M = 2, 3, 4, 5$, respectively. (e, f) The solution obtained using the separate mixtures model with two and three components per class conditional density, respectively. A different line style is used for the density model of each class.

values of $M = 2, 3, 4, 5$. The obtained solutions are displayed in Figures 4a through 4d, where it is shown that the hierarchical mixture classification model finds exactly the same solution for $M = 2, 3, 4$ (with five active components), while for $M = 5$, six active components are used, but the solution is very similar to the previous cases. The higher the value of M becomes, the greater the number of pruned components; thus, the model is not prone to overfitting. We also measured the log-likelihood value achieved in a test data set; it was −757.8 for the first three cases and −758.12 for the last case. On the other hand, the separate mixtures model cannot avoid overfitting, because when we increase the number of components in a mixture model, the likelihood value always increases. In Figures 4e and 4f, we display the solutions of the separate mixtures model when each class conditional density is modeled using a mixture with two and three components, respectively. It can be observed that all components remain active, resulting in data overfitting. The log-likelihood value in test data was now −759.46 and −764.1, respectively. It must also be noted that analogous results are obtained if four or five components are used.

Another advantage of the hierarchical mixture classifier over separate mixtures is that it is more efficient in classification problems with small data sets. In order for the separate mixtures model to be efficient, we have to infer the number of mixture components corresponding to each class. For example, in the problem of Figure 2, in order to obtain the optimal solution, we have to assume two components for the two class conditional densities and one component for the third one. We can apply model selection techniques, such as cross validation, in order to determine the number of components; however, any model selection method is unreliable when few data are available. Moreover, in case of problems with many classes ($K > 10$) and few available data for each class (compared to the data dimensionality), the separate mixtures model is not applicable. On the contrary, this is not a problem for our approach, since the specification of M components at the second level of the hierarchy is performed based on all data from all classes, and, in addition, we are allowed to overestimate M.

4 Experiments

We conducted a series of experiments using gaussian components and compared the proposed model with the common components model and the separate mixtures model. We considered five well-known data sets: the Clouds, Satimage, and Phoneme from the ELENA database and the Pima Indians and Ionosphere from the UCI repository (Blake & Merz, 1998). Details of these data sets are provided in Table 1. We have performed experiments for several values of M, where M denotes the number of components at the second level of the hierarchical mixture classification model and also the total number of components of the common components model. In the case of separate mixtures, M denotes the number of components used by each class-conditional mixture density. To obtain average and standard deviation error values, we applied the five-fold cross-validation method. The results for all algorithms and all data sets are presented in Tables 2 and 3.

Table 1: Data Sets Used in the Experiments.

Data Set	Features	Classes	Number of Data
Satimage	5	6	6,435
Phoneme	5	2	5,404
Clouds	2	2	5,000
Pima Indians	8	2	768
Ionosphere	35	2	351

Table 2: Generalization Error and Standard Deviation Values for All Tested Algorithms Using the Satimage, Phoneme, and Clouds Data Sets.

Satimage

	M=6		M=12		M=18		M=24	
Algorithm	Error	S.D.	Error	S.D.	Error	S.D.	Error	S.D.
$h_j(x) = P(j \mid x)$	12.0	0.5	**10.7**	0.4	**10.7**	0.8	**10.4**	0.9
$h_j(x) = P(j \mid x, C_k)$	11.9	1.1	11.5	1.0	10.9	0.9	10.6	0.9
Common components model	17.1	0.4	12.9	0.2	12.2	0.3	11.4	0.5
Separate mixtures	**11.2**	0.5	11.2	1.0	11.7	0.9	12.5	0.5

Phoneme

	M=8		M=10		M=12		M=14	
Algorithm	Error	S.D.	Error	S.D.	Error	S.D.	Error	S.D.
$h_j(x) = P(j \mid x)$	**15.5**	1.1	15.2	1.2	15.4	0.8	14.9	1.2
$h_j(x) = P(j \mid x, C_k)$	15.8	1.1	**14.7**	1.0	**14.0**	0.9	**14.5**	1.0
Common components model	22.0	1.1	20.6	1.9	19.9	1.1	21.3	1.2
Separate mixtures	16.3	1.5	15.9	1.0	15.9	1.4	15.5	1.1

Clouds

	M=4		M=6		M=8		M=10	
Algorithm	Error	S.D.	Error	S.D.	Error	S.D.	Error	S.D.
$h_j(x) = P(j \mid x)$	16.7	2.5	12.9	0.9	12.6	1.0	12.5	0.9
$h_j(x) = P(j \mid x, C_k)$	13.1	0.9	**11.3**	1.0	10.9	0.9	10.8	0.9
Common components model	13.1	0.9	11.4	0.9	**10.9**	0.9	**10.8**	0.8
Separate mixtures	**12.2**	0.6	11.9	0.8	11.1	1.3	10.8	0.8

Note: Numbers in boldface type indicate best performance among the tested algorithms.

The hierarchical mixture classification model was trained using the two algorithms described in section 3.1. Experimental results are displayed in Tables 2 and 3, where bold numbers indicate best performance among the tested algorithms. The two algorithms are denoted as $h_j(x) = P(j \mid x)$ and $h_j(x) = P(j \mid x, C_k)$, respectively. Moreover, since the training of the hierar-

Table 3: Generalization Error and Standard Deviation Values for All Tested Algorithms Using the Pima Indians and Ionosphere Data Sets.

Pima Indians

	M=6		M=8		M=10		M=12	
Algorithm	Error	S.D.	Error	S.D.	Error	S.D.	Error	S.D.
$h_j(x) = P(j \mid x)$	26.0	1.1	**24.7**	2.5	24.8	2.7	25.0	2.5
$h_j(x) = P(j \mid x, C_k)$	**24.3**	1.8	24.8	1.7	**24.6**	2.5	**24.8**	2.8
Common components model	28.6	3.6	29.5	2.9	28.1	3.7	26.9	2.6
Separate mixtures	27.1	2.8	27.1	3.0	26.3	3.6	28.9	2.3

Ionosphere

	M=6		M=8		M=10		M=12	
Algorithm	Error	S.D.	Error	S.D.	Error	S.D.	Error	S.D.
$h_j(x) = P(j \mid x)$	13.7	3.0	**10.0**	3.1	9.4	2.6	7.4	3.6
$h_j(x) = P(j \mid x, C_k)$	12.6	4.0	12.0	3.6	**7.4**	3.2	**7.4**	1.3
Common components model	17.7	4.0	16.3	3.4	12.0	3.4	9.5	3.3
Separate mixtures	**12.1**	2.6	12.3	3.3	8.9	3.4	8.8	2.7

Note: Numbers in boldface type indicate best performance among the tested algorithms.

chical mixture classification model for the case $h_j(x) = P(j \mid x, C_k)$ requires the construction of the common components model, we have also obtained a solution for the common components model at no additional effort.

The experimental results indicate the following:

1. Both algorithms for training the hierarchical mixture classification model provide better generalization results than the separate mixtures and the common components model in most of the runs.

2. The algorithm that uses $h_j(x) = P(j \mid x, C_k)$ provides a classifier that significantly improves the corresponding common components classifier obtained at the intermediate training stage. For example, in the case of the phoneme data set, the improvement is impressive. This constitutes an experimental justification of the discussion in section 3.2. In the case of the Clouds data set, the two classifiers provide approximately the same class conditional density estimates (the second case of proposition 1 is applicable), and thus the two methods exhibit almost equal performance.

3. According to all results, the hierarchical mixture classifier is quite robust regarding the number of components M, and its classification performance is typically improved as we increase M. On the other hand, the performance of the separate mixtures model is very sensitive to the choice of M. For example, in the Satimage data set, this method clearly cannot avoid overfitting as we increase M.

5 Discussion

A hierarchical mixture classification model has been presented that exhibits a three-level structure. This structure provides at the higher level an unsupervised representation of the data and then at a lower level information about the classes having generated the data. The proposed model can be considered as a mixture of experts classifier, since the components at the second level of the hierarchy partition the data space into subspaces, while the probability models at the third level form the experts that solve the classification problem in each subspace.

The proposed hierarchical mixture classifier exhibits several attractive features compared to conventional mixture models for classification. Specifically, compared to the common components model, it improves data representation in subspaces relevant to classification. Also, the model exhibits robustness with respect to the number M of components at the second level, and this constitutes a great advantage over the separate mixtures model.

For future research, several interesting directions may be followed. Any advanced method for mixture density estimation (Ormeneit & Tresp, 1996; Ueda, Nakano, Ghahramani, & Hinton, 2000; Vlassis & Likas, 2002) can be incorporated at the first stage (computation of $h_j(x)$) of the proposed training algorithm of the hierarchical mixture classification model. Though such methods can be directly applied where $h_j(x) = P(j \mid x)$, slightly modified versions are needed for the case where $h_j(x) = P(j \mid x, C_k)$. Also in the proposed approach, the probability models $p(x \mid C_k, j, \theta_{kj})$ are assumed to be unimodal densities taken from the exponential family; however, other models may be used, such as factor analyzers (Everitt, 1984), or each $p(x \mid C_k, j, \theta_{kj})$ may itself be a mixture model.

Appendix A: Specification of $h_j(x)$ for Gaussian Components

A.1 Approximation of $h_j(x)$ by $P(j \mid x)$. We assume that the mixture model employed for determining the probabilities $P(j \mid x)$ has gaussian components of the form 3.16. We can obtain an estimation of $P(j \mid x)$ by iteratively applying until convergence the following update equations:

$$P(j \mid x, \Phi^{(t)}) = \frac{p(x \mid j, \mu_j^{(t)}, \Sigma_j^{(t)})\pi_j^{(t)}}{\sum_{i=1}^{M} p(x \mid i, \mu_i^{(t)}, \Sigma_i^{(t)})\pi_i^{(t)}} \tag{A.1}$$

$$\mu_j^{(t+1)} = \frac{\sum_{x \in X} P(j \mid x, \Phi^{(t)})x}{\sum_{x \in X} P(j \mid x, \Phi^{(t)})} \tag{A.2}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{x \in X} P(j \mid x, \Phi^{(t)})(x - \mu_j^{(t+1)})(x - \mu_j^{(t+1)})^T}{\sum_{x \in X} P(j \mid x, \Phi^{(t)})} \tag{A.3}$$

$$\pi_j^{(t+1)} = \frac{1}{|X|} \sum_{x \in X} P(j \mid x, \Phi^{(t)}), \tag{A.4}$$

where equation A.1 holds for each $x \in X$ and j and equations A.2 through A.4 for each j.

A.2 Approximation of $h_j(x)$ by $P(j \mid x, C_k)$. We assume that the common components model employed for determining the probability $P(j \mid x, C_k)$ employs gaussian components. The EM algorithm for maximizing the log likelihood 2.3 gives the following update equations (Titsias & Likas, 2001):

$$P(j \mid x, C_k, \Phi_k^{(t)}) = \frac{\pi_{jk}^{(t)} p(x \mid j, \mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_{i=1}^{M} \pi_{ik}^{(t)} p(x \mid i, \mu_i^{(t)}, \Sigma_i^{(t)})} \tag{A.5}$$

$$\mu_j^{(t+1)} = \frac{\sum_{k=1}^{K} \sum_{x \in X_k} P(j \mid x, C_k, \Phi_k^{(t)}) x}{\sum_{k=1}^{K} \sum_{x \in X_k} P(j \mid x, C_k, \Phi_k^{(t)})} \tag{A.6}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{k=1}^{K} \sum_{x \in X_k} P(j \mid x, C_k, \Phi_k^{(t)})(x - \mu_j^{(t+1)})(x - \mu_j^{(t+1)})^T}{\sum_{k=1}^{K} \sum_{x \in X_k} P(j \mid x, C_k, \Phi_k^{(t)})} \tag{A.7}$$

$$\pi_{jk}^{(t+1)} = \frac{1}{|X_k|} \sum_{x \in X_k} P(j \mid x, C_k, \Phi_k^{(t)}) \tag{A.8}$$

where all equations holds for each j, while equation A.8 holds additionally for each k and equation A.5 for each k and $x \in X$.

Appendix B: Proof of Proposition 1

For the parameter solution $\hat{\Theta}$, the conditional density estimate of the class C_k is

$$p(x \mid C_k, \hat{\Theta}) = \sum_{j=1}^{M} P(j \mid C_k, \hat{\Theta}) p(x \mid C_k, j, \hat{\theta}_{kj}), \tag{B.1}$$

where according to equations 3.21 and 3.14,

$$P(j \mid C_k, \hat{\Theta}) = \frac{1}{|X_k|} \sum_{x \in X_k} P(j \mid x, C_k, \hat{\Phi}_k) \tag{B.2}$$

and

$$\hat{\theta}_{kj} = \underset{\theta_{kj}}{\operatorname{argmax}} \sum_{x \in X_k} P(j \mid x, C_k, \hat{\Phi}_k) \log p(x \mid C_k, j, \theta_{kj}), \tag{B.3}$$

respectively. Also, the corresponding class conditional estimate provided by the common components model is given by

$$p(x \mid C_k, \hat{\Phi}_k) = \sum_{j=1}^{M} \hat{\pi}_{jk} p(x \mid j, \hat{\phi}_j). \tag{B.4}$$

Assume the C_k-class log likelihood corresponding to the data set X_k:

$$L_k(\Phi_k) = \sum_{x \in X_k} \log \sum_{j=1}^{M} \pi_{jk} p(x \mid j, \phi_j). \tag{B.5}$$

If we apply one EM iteration to maximize the above log likelihood starting from $\Phi_k^{(0)} = \{\hat{\phi}_1, \ldots, \hat{\phi}_M, \hat{\pi}_{1k}, \ldots, \hat{\pi}_{Mk}\}$, the parameter value $\Phi_k^{(1)}$ is obtained by maximizing the function

$$Q(\Phi_k \mid \Phi_k^{(0)}) = \sum_{x \in X_k} \sum_{j=1}^{M} P(j \mid x, C_k, \Phi_k^{(0)}) \log \pi_{jk} p(x \mid j, \phi_j), \tag{B.6}$$

which yields

$$\pi_{jk}^{(1)} = \frac{1}{|X_k|} \sum_{x \in X_k} P(j \mid x, C_k, \Phi_k^{(0)}) \tag{B.7}$$

and

$$\phi_j^{(1)} = \underset{\phi_j}{\operatorname{argmax}} \sum_{x \in X_k} P(j \mid x, C_k, \Phi_k^{(0)}) \log p(x \mid j, \phi_j). \tag{B.8}$$

Now clearly from equations B.2 and B.7, it holds that $P(j \mid C_k, \hat{\Theta}) = \pi_{jk}^{(1)}$. Also since $p(x \mid j, \phi_j)$ has the same parametric form with $p(x \mid C_k, j, \theta_{kj})$, $\phi_j^{(1)}$ and $\hat{\theta}_{kj}$ are obtained by maximizing the same quantity (see equations B.3 and B.8). Thus, it holds that $\hat{\theta}_{kj} = \phi_j^{(1)}$ and the class conditional estimates $p(x \mid C_k, \Phi_k^{(1)})$ and $p(x \mid C_k, \hat{\Theta})$ are identical. Now, one of the following two cases holds:

1. $\nabla_{\Phi_k} L_k(\hat{\Phi}_k) \neq 0$: The convergence property of the EM algorithm implies that if for the log likelihood $L(\Theta)$ of interest it holds that $\nabla_{\Theta} L(\Theta^{(t)}) \neq 0$, then at the next EM iteration, it will hold that $L(\Theta^{(t+1)}) > L(\Theta^{(t)})$ (Wu, 1983; McLachlan & Krishnan, 1997). Thus, in our case, we find that

$$\begin{aligned} &\sum_{x \in X_k} \log \sum_{j=1}^{M} P(j \mid C_k, \Phi_k^{(1)}) p(x \mid j, \phi_j^{(1)}) \\ &\quad > \sum_{x \in X_k} \log \sum_{j=1}^{M} \pi_{jk}^{(0)} p(x \mid j, \phi_j^{(0)}) \end{aligned} \tag{B.9}$$

   which proves inequality 3.22.

2. $\nabla_{\Phi_k} L_k(\hat{\Phi}_k) = 0$: Since the EM algorithm converges to a stationary point (Wu, 1983), it holds that $\Phi_k^{(1)} = \Phi_k^{(0)}$. Consequently, since $p(x \mid C_k, \hat{\Theta})$ is identical to $p(x \mid C_k, \Phi_k^{(1)})$, it will also be identical to $p(x \mid C_k, \hat{\Phi}_k)$.

References

Bishop, C. M., & Tipping, M. E. (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine intelligence, 20*(3), 281–293.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. University of California, Irvine, Department of Computer and Information Sciences.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B, 39*, 1–38.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Everitt, B. S. (1984). *An introduction to the latent variable models*. London: Chapman and Hall.

Ghahramani, Z., & Jordan, M. I. (1994). Supervised learning from incomplete data via an EM approach. In D. J. Cowan, G. Tesauro, & J. Alspector (Eds.), *Neural information processing systems, 6* (pp. 120–127). Cambridge, MA: MIT Press.

Hastie, T. J., & Tibshirani, R. J. (1996). Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society B, 58*, 155–176.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*, 79–87.

Jordan, M. I., & Jacobs R. A. (1994). Hierarchical mixtures of experts and the EM algorithm, *Neural Computation, 6*, 181–214.

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.

Miller, D. J., & Uyar, H. S. (1996). A mixture of experts classifier with learning based on both labeled and unlabeled data. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Neural information processing systems, 9*. Cambridge, MA: MIT Press.

Miller, D. J., & Uyar, H. S. (1998). Combined learning and use for a mixture model equivalent to the RBF classifier. *Neural Computation, 10*, 281–293.

Ormoneit, D., & Tresp, V. (1996). Improved gaussian mixture density estimates using Bayesian penalty terms and network averaging. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems, 8* (pp. 542–548). Cambridge, MA: MIT Press.

Titsias, M. K., & Likas, A. (2001). Shared kernel models for class conditional density estimation. *IEEE Trans. on Neural Networks, 12*(5), 987–997.

Ueda, N., Nakano, R., Ghahramani, Z., & Hinton, G. E. (2000). SMEM algorithm for mixture models. *Neural Computation, 12*, 2109–2128.

Vlassis, N., & Likas, A. (2002). A greedy EM algorithm for gaussian mixture learning. *Neural Processing Letters, 15*, 77–87.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics, 11*, 95–103.

Received October 16, 2001; accepted April 1, 2002.