MIXTURE OF EXPERTS CLASSIFICATION USING A HIERARCHICAL MIXTURE MODEL

M.K. Titsias and A. Likas

25 – 2001

Preprint, no 25 – 01 / 2001

Department of Computer Science
University of Ioannina
45110 Ioannina, Greece

Mixture of Experts Classification Using a Hierarchical Mixture Model

Michalis K. Titsias and Aristidis Likas

Department of Computer Science
University of Ioannina
45110 Ioannina - GREECE
e-mail: mtitsias@cs.uoi.gr, arly@cs.uoi.gr

Abstract

A three-level hierarchical mixture model for classification is presented which models the following data generation process i) the data are generated by a finite number of sources (clusters) and ii) 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 as a mixture of experts classifier where both the gating network units and the specialised experts are suitably defined from the hierarchical mixture. In order to learn the parameters of the model we have developed a general training approach based on maximum likelihood which 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). The computationally intensive part of the design of such classifier concerns the estimation of the class conditional densities. The common approach 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 instance, 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 implies the existence of common characteristics among data with different class labels is the following:

- the data is drawn by a finite number of sources (clusters).
- within each cluster the data is generated by *labeled* sources which form subclusters of the parent cluster.

The above 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*.

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 which divides the problem into smaller problems and expert networks which solve each subproblem. In our case both the gating network units and the specialised experts are suitably defined from the hierarchical mixture.

In order to learn the parameters of the hierarchical mixture classification model we derive a general training approach based on the maximum likelihood framework which results in two fast training algorithms. We provide comparative results for the two training algorithms using well-known artificial and real data sets. Moreover, an additional feature of the hierarchical mixture classifier is that it provides class conditional density estimates as 'flat' mixtures. Consequently, it is possible to directly compare the method 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. This is the most widely used method and has been studied in (Hastie & 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 later is actually similar to using an RBF or an RBF-like neural network for solving classification problems. This is further investigated in (Miller & Uyar, 1998) where the Bayes decision function of a classifier which 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*. We claim that the hierarchical mixture classifier is an extension of the common components model and also compares favorably to the separate mixtures model.

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 theoretical justification that the proposed method is more efficient compared to the common components model. This justification is also verified 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 each data point x generated from a class 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 $P(C_k|x)$. Using the Bayes rule, the posterior probability $P(C_k|x)$ is written:

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

where $P(C_k)$ is the class prior probability and $p(x|C_k)$ the corresponding class conditional density. Each class conditional density $p(x|C_k)$ 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 $p(x|j, \theta_j)$, $j = 1, \ldots, M$, 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:

$$p(x|C_k, \Theta_k) = \sum_{j \in T_k} \pi_{jk} p(x|j, \theta_j) \tag{2}$$

where the parameter π_{jk} represents the probability $P(j|C_k)$ 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 later implies that the data of different classes may have been generated from some common data sources. According to (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, ie. $p(x|j) = p(x|j, C_k)$.

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 is a priori assumed to be generated by clusters which are not common with clusters corresponding to differently labeled data. This model results from (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 & 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 (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 (2).

To classify a new data point x based on the Bayes formula (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, $k = 1, ..., K$. Learning the whole parameter vector Θ can be performed by maximizing the following log likelihood $L(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log P_k p(x|C_k, \Theta_k)$:

$$\begin{aligned} L(\Theta) &= \sum_{k=1}^{K} |X_k| \log P_k + \sum_{k=1}^{K} \sum_{x \in X_k} \log \sum_{j \in T_k} \pi_{jk} p(x|j, \theta_j) \\ &= \sum_{k=1}^{K} |X_k| \log P_k + \sum_{k=1}^{K} L_k(\Theta_k) \end{aligned} \tag{3}$$

where L_k is the class log likelihood corresponding to the subset X_k. Maximization of the first term in (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 later 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 $p(x, C_k|\Theta) = P_k \sum_{j \in T_k} \pi_{jk} p(x|j, \theta_j)$ and since $P_k \pi_{jk} = P(j|\Theta) P(C_k|j, \Theta)$ (where $P(j|\Theta) = \sum_{k \in F_j} \pi_{jk} P_k$ and $P(C_k|j, \Theta) = \frac{P_k \pi_{jk}}{P(j|\Theta)}$), we obtain:

$$p(x, C_k|\Theta) = \sum_{j=1}^{M} P(j|\Theta) P(C_k|j, \Theta) p(x|j, \theta_j). \tag{4}$$

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 $P(j|\Theta)$.

- Select a class label C_k, where $k \in F_j$, with probability $P(C_k|j, \Theta)$ and draw x from density $p(x|j, \theta_j)$.

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, both x and C_k are independent from one another provided that the component variable j has been observed.

This can be explained by considering that according to (2) x is independent from C_k given j, while the opposite results from the fact that $P(C_k|x, j, \Theta) = \frac{p(x|j,\theta_j)\pi_{jk}P_k}{p(x|j,\theta_j)\sum_{\ell\in F_j} P_\ell \pi_{j\ell}} = P(C_k|j, \Theta)$.

Finally if we are interested in the unconditional density of x, this is given by $p(x|\Theta) = \sum_{j=1}^{M} P(j|\Theta)p(x|j, \theta_j)$ which clearly is a 'flat' mixture. In the next section we present a classification model which 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: i) the data is generated by M clusters and ii) within each cluster the data is generated by class-labeled sources which form subclusters of the larger cluster. If a subcluster corresponding to class C_k can be modeled by the density $p(x|C_k, j, \theta_{kj})$ (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 Fig. 1:

$$p(x|\Theta) = \sum_{j=1}^{M} \pi_j \sum_{k=1}^{K} P_{kj} p(x|C_k, j, \theta_{kj}) \tag{5}$$

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

Clearly the second level of the hierarchical mixture (Fig. 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 by marginalizing out the class labels, ie. $p(x|j, \Theta) = \sum_{k=1}^{K} P_{kj}p(x|C_k, j, \theta_{kj})$. 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 $P(C_k|x)$ must be computed:

$$P(C_k|x, \Theta) = \frac{\sum_{j=1}^{M} \pi_j P_{kj} P(x|C_k, j, \theta_{kj})}{p(x|\Theta)}. \tag{6}$$

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

$$P(C_k|x, \Theta) = \sum_{j=1}^{M} P(j|x, \Theta) P(C_k|x, j, \Theta) \tag{7}$$

where

$$P(j|x, \Theta) = \frac{\pi_j p(x|j, \Theta)}{p(x|\Theta)} \tag{8}$$

and

$$P(C_k|x, j, \Theta) = \frac{P_{kj} p(x|C_k, j, \theta_{kj})}{p(x|j, \Theta)}. \tag{9}$$

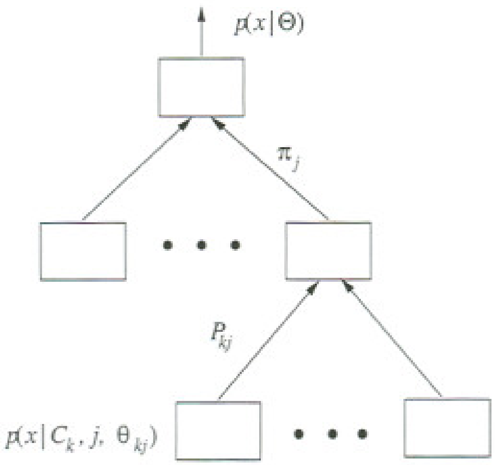

Figure 1: Estimation of the unconditional density of x by the hierarchical mixture classification model.

Expression (7) explicitly denotes that the model estimates the posterior $P(C_k|x)$ as a mixture of experts model. The mixture of experts network was originally introduced in (Jacobs, Jordan, Nowlan, & Hinton, 1991) and further extended to a hierarchical structure in (Jordan & Jacobs, 1994). A mixture of experts network consist of several experts models which 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 $P(j|x, \Theta)$ provided by (8), while the estimates of the experts correspond to the locally computed posterior probabilities of class membership $P(C_k|x, j, \Theta)$ provided by (9).

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

$$P(C_k|\Theta) = \sum_{j=1}^{M} P_{kj}\pi_j \tag{10}$$

and

$$p(x|C_k, \Theta) = \sum_{j=1}^{M} P(j|C_k, \Theta)p(x|C_k, j, \theta_{kj}) \tag{11}$$

respectively, where

$$P(j|C_k, \Theta) = \frac{P_{kj}\pi_j}{P(C_k|\Theta)}. \tag{12}$$

Note that according to (11) each class conditional density exhibits a 'flat' mixture form. 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 $\{1, \dots, M\}$ 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 $p(x|C_k, j, \theta_{kj})$.

In contrast to the models described in Section 2, in this case a class label C_k and the corresponding data value x are not independently selected given the component j. Clearly in this case x and C_k are dependent one another given that the component variable j is observed (see equations (5) and (9)) and this yields the hierarchical mixture classification model to be in principle different from any mixture model classifier described in Section 2.

In order to gain better understanding of the characteristics of the proposed model it is useful to compare it with the common components model. It is clear that the data generation mechanisms of the two methods actually differ in the way that a data point x is selected. More specifically, 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 $p(x|j, \theta_j)$. In contrast, the hierarchical mixture classification model assumes that the data generated by the component j are explained in a way that depends on the their class labels (for each class C_k a different probability model $p(x|C_k, j, \theta_{kj})$ is provided). In this sense the hierarchical mixture classification model can be considered as an extension of the common components model. We further elaborate on this issue in subsection 3.2, where we provide a quantitative comparison of the two methods based on the class conditional density estimates.

3.1 Training the hierarchical mixture classification model

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

$$L(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log \sum_{j=1}^{M} \pi_j P_{kj} p(x|C_k, j, \theta_{kj}). \tag{13}$$

It is possible to directly maximize the above quantity using the EM algorithm. However such a maximization would yield 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 with 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 resulted complete data log likelihood is

$$L_C(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j=1}^{M} z_j(x) \log \pi_j P_{kj} p(x|C_k, j, \theta_{kj}). \tag{14}$$

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 $P(j|x)$. The second type of expectation takes into account the class label C_k of x and corresponds to the probability $P(j|x, C_k)$[1]. If $h_j(x)$ denotes either $P(j|x)$ or $P(j|x, C_k)$, then $\sum_{j=1}^{M} h_j(x) = 1$ and the expected value of the complete data log likelihood L_C is

$$Q(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j=1}^{M} h_j(x) \log \pi_j P_{kj} p(x|C_k, j, \theta_{kj}). \tag{15}$$

Now, in analogy to the case of the 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

$$\hat{\pi}_j = \frac{1}{|X|} \sum_{k=1}^{K} \sum_{x \in X_k} h_j(x) \tag{16}$$

$$\hat{P}_{kj} = \frac{\sum_{x \in X_k} h_j(x)}{\sum_{\ell=1}^{K} \sum_{x \in X_\ell} h_j(x)} \tag{17}$$

$$\hat{\theta}_{kj} = \arg\max_{\theta_{kj}} \sum_{x \in X_k} h_j(x) \log p(x|C_k, j, \theta_{kj}). \tag{18}$$

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

$$\sum_{x \in X_k} h_j(x) \nabla_{\theta_{kj}} p(x|C_k, j, \theta_{kj}) = 0 \tag{19}$$

with respect to θ_{kj}.

In the Gaussian case the solution of (19) can be analytically obtained. Assume that each probability model $p(x|C_k, j, \theta_{kj})$ is a Gaussian of the general form

$$p(x|C_k, j, \theta_{kj}) = \frac{1}{(2\pi)^{d/2} |\Sigma_{kj}|^{1/2}} \exp\left\{ -\frac{1}{2}(x - \mu_{kj})^T \Sigma_{kj}^{-1} (x - \mu_{kj}) \right\}. \tag{20}$$

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

$$\hat{\mu}_{kj} = \frac{\sum_{x \in X_k} h_j(x) x}{\sum_{x \in X_k} h_j(x)} \tag{21}$$

$$\hat{\Sigma}_{kj} = \frac{\sum_{x \in X_k} h_j(x)(x - \hat{\mu}_{kj})(x - \hat{\mu}_{kj})^T}{\sum_{x \in X_k} h_j(x)}. \tag{22}$$

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

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 (16-18), we must first specify the values of $h_j(x)$, ie. to estimate the probabilities $P(j|x)$ or $P(j|x, C_k)$. An approximation of $P(j|x)$ can be obtained by running a mixture model with M components using the data set X and ignoring class labels. Similarly an approximation of $P(j|x, C_k)$ 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 $h_j(x)$ which are summarized next:

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

$$P(j|x, \hat{\Phi}) = \frac{\hat{\pi}_j p(x|j, \hat{\varphi}_j)}{\sum_{i=1}^{M} \hat{\pi}_i p(x|i, \hat{\varphi}_i)}. \tag{23}$$

- *Algorithm 2: Supervised case* ($h_j(x) = P(j|x, C_k)$): We introduce the common components model $p(x|C_k, \Phi_k) = \sum_{j=1}^{M} \pi_{jk} p(x|j, \varphi_j)$ and obtain a parameter solution $\hat{\Phi}_k$ for each k by maximizing the log likelihood (3) using the EM algorithm (Appendix A.2). Once the quantities $P(j|x, C_k, \hat{\Phi}_k)$ have been specified, we replace $h_j(x)$ by:

$$P(j|x, C_k, \hat{\Phi}_k) = \frac{\hat{\pi}_{jk} p(x|j, \hat{\varphi}_j)}{\sum_{i=1}^{M} \hat{\pi}_{ik} p(x|i, \hat{\varphi}_i)}. \tag{24}$$

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 (10) would essentially be $P(C_k|\hat{\Theta}) = \frac{|X_k|}{|X|}$, where (16) and (17) are used. The class conditional density can be estimated using (11) where

$$P(j|C_k, \hat{\Theta}) = \frac{1}{|X_k|} \sum_{x \in X_k} h_j(x). \tag{25}$$

In Fig. 2 a three-class data set is illustrated along with the parameter solution of the models $p(x|C_k, j, \theta_{kj})$ (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 model constitutes an unsupervised learning task, while the application of the common components model is a supervised task.

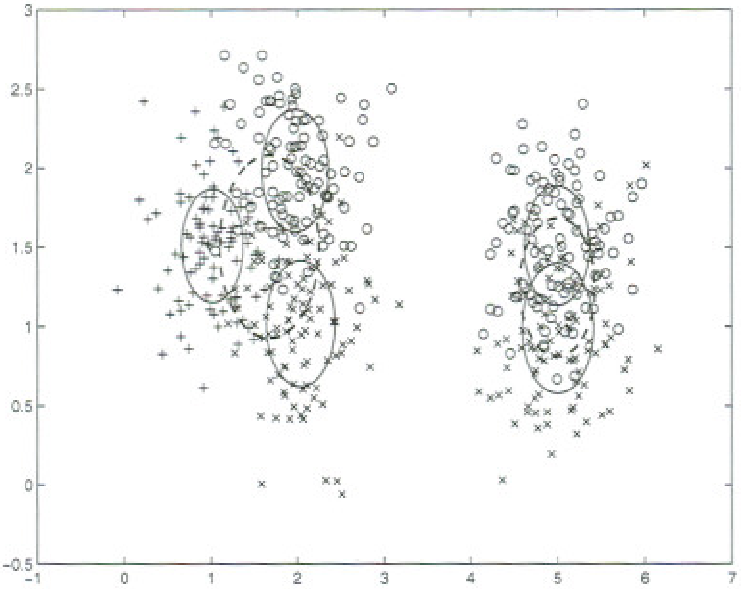

Figure 2: Illustrates the two-dimensional data points of a three-class problem and the parameter solutions for the models $p(x|C_k, j, \theta_{kj})$ which were assumed to be Gaussians (solid lines). The model employs at the second level of the hierarchy two components. The same solution at level 2 has been obtained using either unsupervised or supervised learning. The dash lines represent the parameter solution obtained for the two Gaussian components at the second level. 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 model $p(x|C_k, j, \theta_{kj})$ of the third class is automatically discarded (the corresponding parameter P_{kj} takes zero value).

3.2 Comparison with the common components model

As we have pointed out the hierarchical mixture classification model is more extended model compared to the common components model. In this section we further investigate this issue. 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 the equations (16-18), where $h_j(x)$ is computed as $P(j|x, C_k, \hat{\Phi}_k)$ 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 of the 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 $p(x|C_k, \hat{\Theta}) = \sum_{j=1}^{M} P(j|C_k, \hat{\Theta})p(x|C_k, j, \hat{\theta}_{jk})$ with the corresponding $p(x|C_k, \hat{\Phi}_k) = \sum_{j=1}^{M} \hat{\pi}_{jk} p(x|j, \hat{\varphi}_j)$.

It can be shown that the solution $p(x|C_k, \hat{\Theta})$ is better than $p(x|C_k, \hat{\Phi}_k)$ 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 the equations (16-18), where $h_j(x)$ is computed as $P(j|x, C_k, \hat{\Phi}_k)$ which is the solution provided by the common components model. Also assume that for each j the density models $p(x|j, \varphi_j)$ and $p(x|C_k, j, \theta_{kj})$, $k = 1, \ldots, K$ have the same parametric form which is such that the maximum of (18) occurs for a unique value of the parameters.*

1. *If for a class C_k it holds that $\nabla_{\Phi_k} L_k(\hat{\Phi}_k) \neq 0$, where L_k is C_k-class log likelihood defined in (3), then the estimate $p(x|C_k, \hat{\Theta})$ provides higher class log likelihood value than the estimate $p(x|C_k, \hat{\Phi}_k)$, that is*

$$\sum_{x \in X_k} \log \sum_{j=1}^{M} P(j|C_k, \hat{\Theta}) p(x|C_k, j, \hat{\theta}_{kj}) > \sum_{x \in X_k} \log \sum_{j=1}^{M} \hat{\pi}_{jk} p(x|j, \hat{\varphi}_j). \qquad (26)$$

2. *If for a class C_k it holds that $\nabla_{\Phi_k} L_k(\hat{\Phi}_k) = 0$, then the estimates $p(x|C_k, \hat{\Theta})$ and $p(x|C_k, \hat{\Phi}_k)$ are identical*[2].

The proof is given in Appendix B.

Proposition 1 states that for each C_k the estimate $p(x|C_k, \hat{\Theta})$ can be such that either the class log likelihood value would be higher than the log likelihood computed using $p(x|C_k, \hat{\Phi}_k)$ or it would be identical to $p(x|C_k, \hat{\Phi}_k)$. The second case occurs when the parameter values $\hat{\Phi}_k$ locally maximize the class log likelihood L_k which means that the $p(x|C_k, \hat{\Phi}_k)$ is already a locally optimum

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

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 (3) (corresponding to the common components model case) using the EM algorithm, we may assume that it constitutes a stationary point of the log likelihood. This implies that each $\hat{\pi}_{jk}$ and $\hat{\varphi}_j$ satisfy

$$\hat{\pi}_{jk} = \frac{1}{|X_k|} \sum_{x \in X_k} P(j|x, C_k, \hat{\Phi}_k) \tag{27}$$

$$\sum_{k=1}^{K} \sum_{x \in X_k} P(j|x, C_k, \hat{\Phi}_k) \nabla_{\varphi_j} \log p(x|j, \hat{\varphi}_j) = 0 \tag{28}$$

or

$$\sum_{k=1}^{K} \nabla_{\varphi_j} L_k(\hat{\Phi}_k) = 0. \tag{29}$$

Although $\hat{\pi}_{jk}$ will always correspond to a stationary point of the class log likelihood L_k, equation (29) explicitly points out that $\hat{\varphi}_j$ may not correspond to a stationary point of L_k for all k. In order $\hat{\varphi}_j$ to be stationary point of L_k it must hold that

$$\nabla_{\varphi_j} L_k(\hat{\Phi}_k) = \sum_{x \in X_k} P(j|x, C_k, \hat{\Phi}_k) \nabla_{\varphi_j} \log p(x|j, \hat{\varphi}_j) = 0. \tag{30}$$

The situation where $\hat{\varphi}_j$ satisfies (28) without satisfying (30) for every k occurs when the component j represents data of different classes which do not overlap significantly[3]. In real-world classification problems (with class overlapping) it is almost certain that there would be some φ_j for which the condition (30) will not be true for all k[4]. Those $\hat{\varphi}_j$ 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 $p(x|C_k, \hat{\Theta})$ will improve the corresponding $p(x|C_k, \hat{\Phi}_k)$ and this improvement will be observed in data regions with class overlap. The later can be considered very beneficial from the classification point of view since improvement of the class conditional density estimates at class overlapping data regions can naturally improve discrimination, which means that the obtained decision boundaries approximate more accurately the true decision boundaries.

4 Experiments

To assess the performance of the hierarchical mixture classification model, we have conducted a series of experiments using Gaussian components and compared the proposed model with the

[3] An illustrative example is displayed in Fig. 2. In this figure each component of the common components model (dash lines) represents simultaneously data clusters of different classes which clearly do not have the same means and variances. This yields the class log likelihoods not to be maximized. Note that these quantities would all be simultaneously maximized if the class-subclusters had precisely the same means and variances.

[4] Note that if for a specific $\hat{\varphi}_j$ there exists a class C_k such that $\nabla_{\varphi_j} L_k(\hat{\Phi}_k) \neq 0$, then in order for the equation (28) to be satisfied there must also be at least one different class C_ℓ for which $\nabla_{\varphi_j} L_\ell(\hat{\Phi}_\ell) \neq 0$.

Data set	Features	Classes	Number of data
Satimage	5	6	6435
Phoneme	5	2	5404
Clouds	2	2	5000
Pima indians	8	2	768
Ionosphere	35	2	351

Table 1: Description of the datasets used in the experiments.

common components model and the separate mixtures model. 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. Details of these datasets are provided in Table 1. We have performed experiments for several number of components M, where M denotes the number of components appearing at the second level of the hierarchical mixture classification model and also to the total number of components employed either by the separate mixtures or the common components model. In the case of separate mixtures we considered that an equal number of components is used in the mixture model of each class. To obtain average and standard deviation error values, we applied the 5-fold cross-validation method. The results for all algorithms and all datasets are displayed in Table 2.

The hierarchical mixture classification model was trained using the two algorithms presented in Section 3.1. Experimental results are displayed in Table 2, where bold numbers indicate best performance among the tested algorithms. The two algorithms are denoted in Table 2 as $h_j(x) = P(j|x)$ and $h_j(x) = P(j|x, C_k)$ respectively. Moreover, since the training of the hierarchical mixture classification model for the case $h_j(x) = P(j|x, C_k)$ 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: i) Both algorithms for training the hierarchical mixture classification model provide better generalization results than the separate mixtures and the common components model (except for the Clouds data set where the algorithm that computes $h_j(x)$ as $P(j|x)$ provided the worst performance). ii) The algorithm that uses $h_j(x) = P(j|x, C_k)$ provides a classifier which significantly improves the corresponding common components classifier obtained at the intermediate training stage. This constitutes an experimental justification of the discussion in subsection 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. iii) Both algorithms for training the hierarchical mixture classification model are efficient and can be proved competitive in real-world applications.

Satimage								
	M=6		M=12		M=18		M=24	
Algorithm	error	std	error	std	error	std	error	std
$h_j(x) = P(j \mid x)$	12.04	0.55	**10.75**	0.45	**10.73**	0.81	**10.39**	0.87
$h_j(x) = P(j \mid x, C_k)$	**11.90**	1.1	11.50	0.98	10.89	0.87	10.58	0.95
Comm. comp. model	17.09	0.39	12.91	0.25	12.20	0.32	11.42	0.48
Separate. mixtures	13.68	0.77	12.05	0.53	11.21	0.75	10.98	0.71
Phoneme								
	M=8		M=10		M=12		M=14	
Algorithm	error	std	error	std	error	std	error	std
$h_j(x) = P(j \mid x)$	**15.50**	1.07	15.19	1.16	15.44	0.84	14.85	1.22
$h_j(x) = P(j \mid x, C_k)$	15.76	1.12	**14.74**	1.03	**14.02**	0.91	**14.50**	1.00
Comm. comp. model	22.04	1.11	20.61	1.94	19.87	1.10	21.26	1.16
Separate mixtures	17.8	1.0	17.85	1.4	17.37	0.75	16.88	1.15
Clouds								
	M=4		M=6		M=8		M=10	
Algorithm	error	std	error	std	error	std	error	std
$h_j(x) = P(j \mid x)$	16.68	2.53	12.88	0.94	12.62	1.04	12.48	0.87
$h_j(x) = P(j \mid x, C_k)$	**13.06**	0.90	**11.34**	0.96	10.94	0.94	10.84	0.93
Comm. comp. model	**13.06**	0.90	11.38	0.95	**10.88**	0.91	**10.76**	0.82
Separate mixtures	24.24	2.03	20.44	4.45	11.86	0.85	11.36	0.98
Pima Indians								
	M=6		M=8		M=10		M=12	
Algorithm	error	std	error	std	error	std	error	std
$h_j(x) = P(j \mid x)$	26.01	1.07	**24.71**	2.51	24.84	2.69	24.97	2.51
$h_j(x) = P(j \mid x, C_k)$	**24.31**	1.81	24.84	1.73	**24.58**	2.47	**24.71**	2.79
Comm. comp. model	28.63	3.56	29.54	2.86	28.10	3.70	26.93	2.59
Separate mixtures	27.11	2.2	26.67	3.44	26.69	3.58	26.43	1.34
Ionosphere								
	M=6		M=8		M=10		M=12	
Algorithm	error	std	error	std	error	std	error	std
$h_j(x) = P(j \mid x)$	13.66	3.05	**9.98**	3.07	9.40	2.58	7.41	3.56
$h_j(x) = P(j \mid x, C_k)$	**12.56**	3.97	11.96	3.64	**7.41**	3.22	**7.39**	1.29
Comm. comp. model	17.69	4.01	16.26	3.39	11.98	3.38	9.5	3.31
Separate mixtures	15.09	3.8	11.82	1.89	12.24	3.77	9.39	3

Table 2: Generalization error and standard deviation values for all tested algorithms and datasets.

5 Discussion

A hierarchical mixture classification model has been presented which exhibits a three-level structure. This structure provides at the higher level an unsupervised representation of the data and then at a lower level provides 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 which solve the classification problem in each subspace.

The hierarchical mixture classifier exhibits several attractive features compared to conventional mixture models for classification. More specifically, the data generation assumption behind the proposed model is more general than that of the common components model classifier and this leads to improved performance results as it has been shown both theoretically and experimentally. Also, due to the structure of the model, the class conditional densities are estimated by taking into account data from all classes and this constitutes a computational advantage compared to the separate mixtures model. For instance, if we have a problem with many classes ($K > 10$) and few data is available for each class, then a method which separately estimates the class conditional densities may not be applicable. On the contrary this is not a problem for our approach and the training method described in Section 3.1 is mainly based on training using all available data for the specification of $h_j(x)$.

In what concerns future research, several interesting directions may be followed: Any advanced method for mixture density estimation such as the (Ormeneit & Tresp, 1996; Ueda, Nakano, Ghahramani, & Hinton, 2000) 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 in case where $h_j(x) = P(j|x)$, slightly modified versions are needed for the case $h_j(x) = P(j|x, C_k)$. Also in the proposed approach the probability models $p(x|C_k, j, \theta_{kj})$ 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 $p(x|C_k, j, \theta_k)$ may itself be a mixture model. Finally another important research direction is to develop a Bayesian approach for learning the parameters of the model.

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, Dept. 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: Chaoman 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., & Krishnan, T. (1997). The EM algorithm and extensions. Marcel Dekker.

McLachlan, G. J., & Peel, D. (2000). Finite mixture models. 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, 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.

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

A Specification of $h_j(x)$ for Gaussian components

A.1 Approximation of $h_j(x)$ by $P(j|x)$

We assume that the mixture model employed for determining the probabilities $P(j|x)$ has Gaussian components of the form (20). We can obtain an estimation of $P(j|x)$ by iteratively applying until convergence the following update equations:

$$P(j|x,\Phi^{(t)}) = \frac{p(x|j,\mu_j^{(t)},\Sigma_j^{(t)})\pi_j^{(t)}}{\sum_{i=1}^{M} p(x|i,\mu_i^{(t)},\Sigma_i^{(t)})\pi_i^{(t)}} \tag{31}$$

$$\mu_j^{(t+1)} = \frac{\sum_{x\in X} P(j|x,\Phi^{(t)})x}{\sum_{x\in X} P(j|x,\Phi^{(t)})} \tag{32}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{x\in X} P(j|x,\Phi^{(t)})(x-\mu_j^{(t+1)})(x-\mu_j^{(t+1)})^T}{\sum_{x\in X} P(j|x,\Phi^{(t)})} \tag{33}$$

$$\pi_j^{(t+1)} = \frac{1}{|X|}\sum_{x\in X} P(j|x,\Phi^{(t)}) \tag{34}$$

where (31) holds for each $x \in X$ and j and (32-34) for each j.

A.2 Approximation of $h_j(x)$ by $P(j|x, C_k)$

We assume that the common components model employed for determining the probability $P(j|x, C_k)$ employs Gaussian components. The EM algorithm for maximizing the log likelihood (3) gives the following update equations (Titsias & Likas, 2001):

$$P(j|x,C_k,\Phi^{(t)}) = \frac{\pi_{jk}^{(t)} p(x|j,\mu_j^{(t)},\Sigma_j^{(t)})}{\sum_{i=1}^{M} \pi_{ik}^{(t)} p(x|i,\mu_i^{(t)},\Sigma_i^{(t)})} \tag{35}$$

$$\mu_j^{(t+1)} = \frac{\sum_{k=1}^{K}\sum_{x\in X_k} P(j|x,C_k,\Phi^{(t)})x}{\sum_{k=1}^{K}\sum_{x\in X_k} P(j|x,C_k,\Phi^{(t)})} \tag{36}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{k=1}^{K}\sum_{x\in X_k} P(j|x,C_k,\Phi^{(t)})(x-\mu_j^{(t+1)})(x-\mu_j^{(t+1)})^T}{\sum_{k=1}^{K}\sum_{x\in X_k} P(j|x,\Phi^{(t)})} \tag{37}$$

$$\pi_{jk}^{(t+1)} = \frac{1}{|X_k|}\sum_{x\in X_k} P(j|x,C_k,\Phi^{(t)}) \tag{38}$$

where all equations holds for each j, while (38) holds additionally for each k and (35) for each k and $x \in X$.

B Proof of Proposition 1

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

$$p(x|C_k,\hat{\Theta}) = \sum_{j=1}^{M} P(j|C_k,\hat{\Theta})p(x|C_k,j,\hat{\theta}_{kj}) \tag{39}$$

where according to (25) and (18)

$$P(j|C_k, \hat{\Theta}) = \frac{1}{|X_k|} \sum_{x \in X_k} P(j|x, C_k, \hat{\Phi}_k) \tag{40}$$

and

$$\hat{\theta}_{kj} = \max_{\theta_{kj}} \sum_{x \in X_k} P(j|x, C_k, \hat{\Phi}_k) \log p(x|C_k, j, \theta_{kj}) \tag{41}$$

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

$$p(x|C_k, \hat{\Phi}_k) = \sum_{j=1}^{M} \hat{\pi}_{jk} p(x|j, \hat{\varphi}_j). \tag{42}$$

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

$$L_k(\Phi_k) = \sum_{x \in X_k} \log \sum_{j=1}^{M} \pi_{jk} p(x|j, \varphi_j). \tag{43}$$

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

$$Q(\Phi_k|\Phi_k^{(0)}) = \sum_{x \in X_k} \sum_{j=1}^{M} P(j|x, C_k, \Phi_k^{(0)}) \log \pi_{jk} p(x|j, \varphi_j) \tag{44}$$

which yields

$$\pi_{jk}^{(1)} = \frac{1}{|X_k|} \sum_{x \in X_k} P(j|x, C_k, \Phi_k^{(0)}) \tag{45}$$

and

$$\varphi_j^{(1)} = \max_{\varphi_j} \sum_{x \in X_k} P(j|x, C_k, \Phi_k^{(0)}) \log p(x|j, \varphi_j). \tag{46}$$

Now clearly from (40) and (45) it holds that $P(j|C_k, \hat{\Theta}) = \pi_{jk}^{(1)}$. Also since $p(x|j, \varphi_j)$ has the same parametric form with $p(x|C_k, j, \theta_{kj})$, $\varphi_j^{(1)}$ and $\hat{\theta}_{kj}$ are obtained by maximizing the same quantity (equations (41) and (46)). Thus, it holds that $\hat{\theta}_{kj} = \varphi_j^{(1)}$ and the class conditional estimates $p(x|C_k, \Phi_k^{(1)})$ and $p(x|C_k, \hat{\Theta})$ 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

$$\sum_{x \in X_k} \log \sum_{j=1}^{M} P(j|C_k, \Phi_k^{(1)}) p(x|j, \varphi_j^{(1)}) > \sum_{x \in X_k} \log \sum_{j=1}^{M} \pi_{jk}^{(0)} p(x|j, \varphi_j^{(0)}) \tag{47}$$

   which proves inequality (26).

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 $p(x|C_k, \hat{\Theta})$ is identical to $p(x|C_k, \Phi_k^{(1)})$, it will also be identical to $p(x|C_k, \hat{\Phi}_k)$.