CLASS CONDITIONAL DENSITY ESTIMATION USING MIXTURES WITH CONSTRAINED COMPONENT SHARING

M.K. Titsias and A. Likas

8 – 2001

Preprint, no 8 – 01 / 2001

Department of Computer Science
University of Ioannina
45110 Ioannina, Greece

Class Conditional Density Estimation using Mixtures with Constrained Component Sharing

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

The typical approach to classification using mixture models for class densities assumes that each class density is modeled independently using only data of the specific class. Consequently, the components of each mixture are exclusively used to represent the corresponding class data. Following an alternative approach, in previous work we considered the problem of class conditional density modeling using mixture models whose components are shared by all class models. In this paper, we propose an intermediate mixture modeling formulation that allows for models where each mixture component is utilized in a subset of the class densities. We provide an analysis suggesting that in many cases the most efficient classification model is obtained by appropriately determining the sharing of components among class densities. In order to discover such a efficient model, a training method is proposed based on the EM algorithm that automatically adjusts component sharing and provides solutions with good classification performance.

Index terms: mixture models, classification, density estimation, EM algorithm, component sharing.

1 Introduction

Class conditional density estimation is the computationally intensive part of designing a classifier based on statistical theory and Bayes decision theory [3]. In the typical approach, this estimation proceeds by first partitioning the data set into subsets based on class membership and subsequently computing each class density using only the data of the corresponding subset [1, 3]. If we restrict our attention to density estimation methods based on parametric models, the typical approach assumes that the corresponding class conditional models are functionally independent in the sense that they do not have common parameters [3]. In [8, 9] we considered the case where the class conditional densities are modeled using mixtures with common components. Because of the functional analogy of this model with the classical RBF network, we also referred

to this model as the Probabilistic RBF network (PRBF) [8]. Apparently, under this model assumption, the class densities are functionally dependent since the component parameters are common.

Consider a classification problem with K classes and a model with total number of components M. Let θ_j denote the parameter vector of component j and θ the vector of all mixture component parameters $\theta = (\theta_1, \ldots, \theta_M)$. The class conditional densities are given by

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

where π_{jk} is the mixture coefficient representing the prior probability of component j conditioned on the class C_k. Also we denote with π_k the vector of all priors π_{jk} associated with class C_k and with Θ the vector of all model parameters (both priors and component parameters). The priors cannot be negative and for each k satisfy the constraint

$$\sum_{j=1}^{M} \pi_{jk} = 1. \tag{2}$$

This model which will be called the *common component model* is in contrast to the typical class conditional density approach, called the *separate mixtures model*, where each component j contributes to the density mixture model of only one class C_k. The separate mixtures model can be derived as a special case of the common component model by setting for each component j assigned to class C_k the constraint $\pi_{jl} = 0$ for $l \neq k$. In [8, 9] an EM algorithm is presented for training the common component model.

The basic argument concerning the usefulness of modeling with common components is that *it can be beneficial in cases of highly overlapped classes* since the components can represent simultaneously data of different classes [9]. However, in real world problems, we are not aware of the kind of overlapping among the classes and by applying the above models we might arrive at unreliable data representations from the classification perspective. The main problem is that, based on the maximum likelihood principle for training, it is possible to obtain mixture component parameters that simultaneously represent differently labeled data, even in cases where there exists no significant local overlap among classes. This may happen because the maximization of the likelihood typically favors unsupervised solutions. On the other hand, if we had a sufficient

knowledge about a classification problem, we could choose an appropriate mixture modeling scheme where some components would be shared among specific classes, while some others would be assigned to a single class. In this paper, we propose an extension of the common component model (1) with the aim to allow for the representation of cases where each component is used by a *subset* of the class conditional models. We define the corresponding model formulation, called the Z-model, where Z is an indicator matrix specifying the utilization of the components from class densities. We provide examples where for fixed total number of components the maximum likelihood solution exhibiting best classification performance corresponds to a Z-model for an appropriate selection of matrix Z. In addition, we show that any Z-model can be considered as being derived from the common component model (1) by confining the original parameter space to a suitable subspace defined by constraining some priors to be zero. In order to find an effective classifier (Z-model) for a predefined number of mixture components, a learning scheme is proposed that is based on the maximization of a suitably defined objective function. This objective function is very similar to log likelihood and an Expectation - Maximization (EM) algorithm [2] has been developed to perform maximization. The proposed EM algorithm iteratively updates both the component parameters and the priors as well as the degree of sharing of each component among classes. The method has been tested on several artificial and real classification data sets with very promising results.

In Section 2 we present the advantages and drawbacks of modeling using mixtures with common components from the classification point of view. Section 3 describes the generalized mixture modeling formulation (Z-model) and provides examples illustrating the usefulness of constrained component sharing. In Section 4 a training algorithm is presented based on the EM procedure that simultaneously adjusts constraints and parameter values. Experimental results using several classification data sets are presented in Section 5. Finally Section 6 provides conclusions and directions for future research.

2 Comparison of common and separate mixture modeling

In this section a comparative discussion is provided concerning the advantages and drawbacks of separate and common mixture modeling. The following two

issues are considered:

- specification of the number of components needed for efficient data representation
- specification of cases where component sharing is beneficial and of cases where sharing leads to inferior classification performance.

In what concerns first issue, in the common component model only the total number M of components needs to be specified. On the contrary, the application of the separate mixtures model requires the specification of a partitioning $M_1, \ldots, M_K$ of the M components among the class mixtures. We consider this issue as an advantage of common mixture modeling over separate mixtures, since it is difficult to define an effective partitioning[1] and this significantly affects the performance of the approach based on separate mixtures.

In what concerns the second of the above issues, it is obvious that in a common component model some components may represent differently labeled data, that is a component may contribute to the density estimation of more than one classes. This is explicitly prohibited in the case of separate mixtures. Consequently, one may draw the conclusion that the common mixture model provides a way of reducing the required total number of components. Nevertheless, we have to examine if component sharing is also beneficial from the classification perspective. It will be shown that there are cases where component sharing can be meaningful, while in some others the resulting classifier generalizes poorly.

To illustrate the two opposite cases, we describe two examples of classification problems with two classes whose data are displayed in Fig. 1 and 2 respectively. In the first example (Fig. 1), the first class data form two clusters, while the second class data form a cluster which significantly overlaps with one of the first class clusters. In this case, the common component model with two components can represent quite adequately both classes[2] (Fig. 1a). For the same example, the separate mixtures model requires three components (two for the first class model and one for the second) to provide a similar representation and, obviously, the use of only two components (one for each class) leads to poor representation (Fig. 1b).

[1] In the absence of any prior information, the components are distributed equally among the classes.

[2] In both examples the solutions illustrated in figures are maximum likelihood estimates using Gaussian mixture components.

The second example (Fig. 2) is analogous to the previous one, with the difference that the degree of overlap of the differently labeled clusters has been significantly reduced (slight overlap). The common component model (with two components) provides a solution with one of the components placed at the decision boundary (Fig. 2a). Therefore, the corresponding classifier exhibits increased classification error. On the other hand, the employment of the separate mixtures model (with one component per class) provides a solution which, although not so adequate from the density estimation perspective, it approximates the true decision boundary quite sufficiently.

The possibility that a common mixture model may place one component at the decision boundary can be explained if we elaborate on the conditions that hold at the stationary points (local maxima) of the log likelihood with respect to component parameters. Suppose that any available information about the classification problem is expressed through a training set X of known data. The original data set X can be partitioned into K disjoint subsets X_k, $k = 1, ..., K$ each one containing only the data of the corresponding class C_k. If we assume that the points of each subset X_k are independently drawn from density $p(x|C_k; \pi_k, \theta)$, then the log likelihood of X is

$$L(\Theta) = \log P(X|\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log p(x|C_k; \pi_k, \theta) = \sum_{k=1}^{K} L_k(\pi_k, \theta), \qquad (3)$$

where L_k is the class log likelihood corresponding to subset X_k. At a stationary point $\hat{\Theta}$ of L, for each component parameters θ_j holds

$$\sum_{k=1}^{K} \sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) = 0, \qquad (4)$$

and for each prior π_{jk}

$$\hat{\pi}_{jk} = \frac{1}{|X_k|} \sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}). \qquad (5)$$

In the above equations $P(j|x, C_k; \pi_k, \theta)$ denotes the posterior probability that a data point x, belonging to class C_k, has been generated from mixture component j and is given from Bayes' theorem

$$P(j|x, C_k; \pi_k, \theta) = \frac{\pi_{jk} p(x|j; \theta_j)}{\sum_{i=1}^{M} \pi_{ik} p(x|i; \theta_i)}. \qquad (6)$$

Based on equation (4) two types of solutions can be identified concerning the parameters of component j. The first type, called *type 1* solution, concerns

solutions $\hat{\theta}_j$ which are stationary points for every class log likelihood L_k, ie. they satisfy the following equation for each k:

$$\sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) = 0. \quad (7)$$

This type of solution is obtained either in the case where the component is located in a region with high overlap among classes or in the case where a component is placed at a region containing data of a single class only. The second type of solutions, called *type 2*, refers to those satisfying equation (4) without satisfying equation (7) for each k.

The previous definitions described in the context of the common component model are also valid in the case of any class density estimation method using mixture models as described in the following definition.

Assume that each class density is modeled using a mixture $p(x|C_k, \pi_k, \theta_k)$, where θ_k denotes the parameters of the mixture components utilized in the density model of class C_k. A stationary point $\hat{\Theta}$ of the log likelihood (defined similarly to (3)) will be called *type 1* solution, if for each k the parameter vector $\hat{\theta}_k$ is a stationary point of the class log likelihood L_k. Otherwise $\hat{\Theta}$ will be called *type 2* solution.

Obviously, the separate mixtures model always provides type 1 solutions, since in this case the maximization of the log likelihood can be decomposed into K independent maximizations of the class log likelihoods.

Considering again the examples presented above, we observe that in the first example the common component model (Fig. 1a) provides a solution that is approximately type 1. In the second example (Fig. 2a) the solution is not type 1 since the component on the right side of the figure is located at the region with slight class overlap and therefore it does not satisfy the stationary condition for each class log likelihood. In addition, in the first example where both approaches give type 1 solutions, the best method which is the common component model also corresponds to higher log likelihood value.

Based on the above observations and assuming fixed total number of components the following two conclusions can be intuitively drawn:

- type 2 solutions do not correspond to effective component placement from the classification point of view
- the solutions of type 1 which should be considered superior from the classification perspective are those with higher likelihood value.

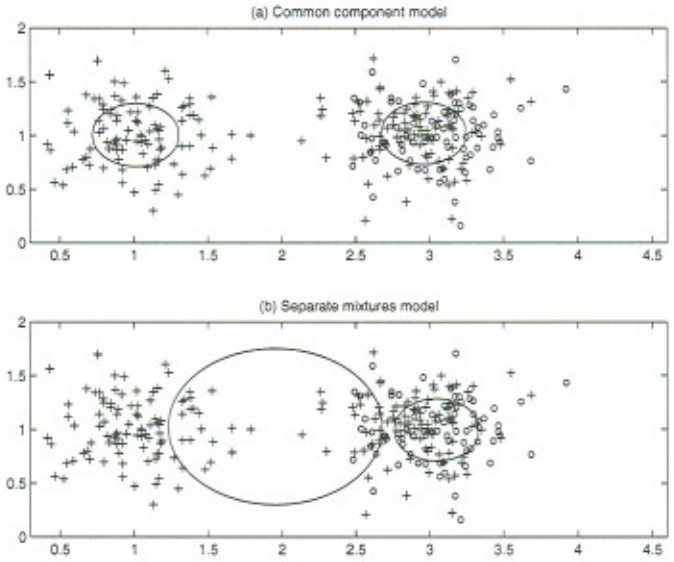

Figure 1: An example problem where the common component model leads to better classification. The data of each class was drawn according to $p(x|C_1) = 0.5N([1\ 1]^T, 0.08) + 0.5N([2.9\ 1]^T, 0.08)$ and $p(x|C_2) = N([3\ 1]^T, 0.08)$, while the prior class probabilities were $P(C_1) = 0.7$ and $P(C_2) = 0.3$. Note that since the clusters are spherical Gaussians we use the notation $N(\mu, \sigma^2)$, where μ is the mean vector and σ^2 the common variance value. Two data sets were generated one for training and one for testing. The common component model (a) provided generalization error 27% and log likelihood value $L = -238.62$. The corresponding generalization error and log likelihood value of the separate mixtures model (b) were 32.2% and $L = -465.97$.

Finally if a problem contains regions with high class overlap and also regions with slight class overlap the most efficient model is the one where some components would be common and while some others will represent data of only one class. This type of model is presented in the next section.

3 Class mixture densities with constrained component sharing

The class conditional density model (1) can be extended by allowing only a *subset* of the total mixture components M to be used by each class conditional model. In order to formulate this model, we introduce an $M \times K$ matrix Z of

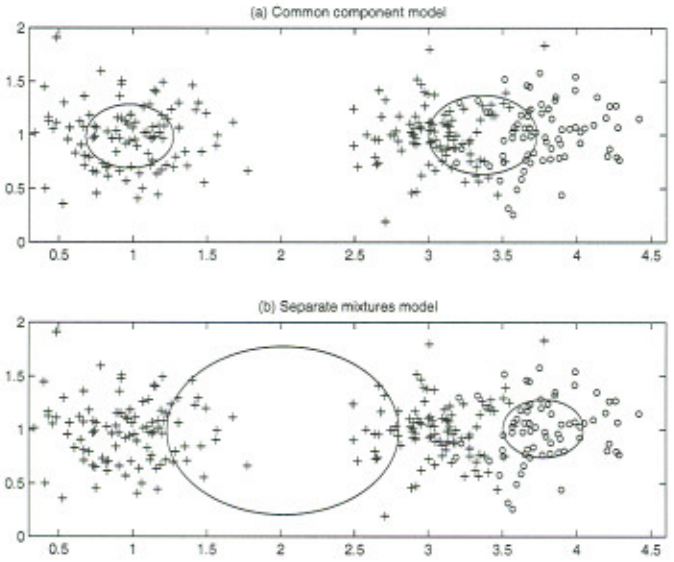

Figure 2: An example problem where the separate mixtures model leads to better classification. The data of each class was drawn according to $p(x|C_1) = 0.5N([1\ 1]^T, 0.08) + 0.5N([3\ 1]^T, 0.08)$ and $p(x|C_2) = N([3.8\ 1]^T, 0.08)$, while the prior class probabilities were $P(C_1) = 0.7$ and $P(C_2) = 0.3$. Two data sets were generated one for training and one for testing. The common mixture model (a) provided generalization error 26.1% and log likelihood value $L = -326.23$. The corresponding generalization error and log likelihood value of the separate mixtures model (b) were 7% and $L = -489.18$.

indicator variables z_{jk} defined as follows:

$$z_{jk} = \begin{cases} 1 & \text{if component } j \text{ contributes to the density model of class } C_k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

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

$$p(x|C_k; z_k, \pi_k, \theta) = \sum_{j=1}^{M} \pi_{jk} p(x|j; \theta_j) = \sum_{j:z_{jk}=1} \pi_{jk} p(x|j; \theta_j), \tag{9}$$

where z_k denotes the k^{th} column of Z and $\{j : z_{jk} = 1\}$ denotes the set of values of j for which $z_{jk} = 1$. Equation (2) for the priors π_{jk} remains valid, however what actually holds for each k is

$$\sum_{j:z_{jk}=1} \pi_{jk} = 1. \tag{10}$$

It is obvious that the common component model is a special Z-model with $z_{jk} = 1$ for all j, k, while the separate mixtures model is a special Z-model with exactly one unit element in each row of the Z matrix.

Consider now a training set X as defined in Section 2. If we assume that the data of each subset X_k are independently drawn according to $p(x|C_k; z_k, \pi_k, \theta)$, the likelihood of the data set X can be defined as

$$P(X|\Theta) = \prod_{k=1}^{K} \prod_{x \in X_k} p(x|C_k; z_k, \pi_k, \theta), \tag{11}$$

It must also be noted that in this formulation the matrix Z is specified in advance and remains fixed, ie. the elements z_{jk} do not constitute adjustable parameters. According to the maximum likelihood principle we wish to find the parameter values which maximize the above function or equivalently its logarithm:

$$L(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \log p(x|C_k; z_k, \pi_k, \theta). \tag{12}$$

Training a Z-model can be performed using the EM algorithm in a manner analogous to case of the common component model [9]. Details and update equations are provided in Appendix A.

In a manner analogous to the previous section, it can be shown that the necessary condition for the component parameters θ_j at stationary points of the log likelihood is

$$\sum_{k:z_{jk}=1} \sum_{x \in X_k} P(j|x, C_k; z_k, \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) = 0 \tag{13}$$

and for the priors π_{jk} is

$$\hat{\pi}_{jk} = \frac{1}{|X_k|} \sum_{x \in X_k} P(j|x, C_k; z_k, \hat{\pi}_k, \hat{\theta}), \tag{14}$$

where (13) holds for every j, while (14) for every j and k such that $z_{jk} = 1$. Also the posterior $P(j|x, C_k; z_k, \pi_k, \theta)$ is given similarly to (6) by

$$P(j|x, C_k; z_k, \pi_k, \theta) = \frac{\pi_{jk} p(x|j; \theta_j)}{\sum_{i:z_{ik}=1} \pi_{ik} p(x|i; \theta_i)}. \tag{15}$$

For the stationary points of a Z-model the following proposition holds.

Proposition 1. *Every stationary point $\hat{\Theta}$ of the log likelihood (12) of a Z-model (for arbitrary Z) is also a stationary point of the log likelihood (3) of the common component model with the same total number of components.*

Proof: A parameter vector value is stationary point of the common component log likelihood (3), if condition (4) holds for $j = 1, \ldots, M$ and condition (5) holds for each $j = 1, \ldots, M$ and $k = 1, \ldots, K$.

Since $\hat{\Theta}$ is stationary point of a Z-model, equation (13) holds for every j. For all $k : z_{jk} = 1$ and $x \in X_k$ holds $P(j|x, C_k; z_k, \hat{\pi}_k, \hat{\theta}) = P(j|x, C_k; \hat{\pi}_k, \hat{\theta})$ according to (6), (9) and (15), so (13) can be written as

$$\sum_{k:z_{jk}=1} \sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) = 0. \tag{16}$$

Moreover, since for all k such that $z_{jk} = 0$ it holds that $\pi_{jk} = 0$, any associated posterior probability $P(j|x, C_k; \hat{\pi}_k, \hat{\theta})$ will also be zero due to (6)). Therefore it holds that $\sum_{k:z_{jk}=0} \sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) = 0$, so (16) can be written

$$\begin{aligned} &\sum_{k:z_{jk}=1} \sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) + \\ &\sum_{k:z_{jk}=0} \sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) = 0, \end{aligned} \tag{17}$$

or

$$\sum_{k=1}^{K} \sum_{x \in X_k} P(j|x, C_k; \hat{\pi}_k, \hat{\theta}) \nabla_{\theta_j} \log p(x|j; \hat{\theta}_j) = 0, \tag{18}$$

which is essentially the condition (4) and holds for any j. In what concerns the priors, condition (5) holds for all $\hat{\pi}_{jk}$ with $z_{jk} = 1$. Obviously, condition (5) also holds for any prior with $z_{jk} = 0$ since in such a case both sides are equal to zero (as shown previously all associated posteriors $P(j|x, C_k; \hat{\pi}_k, \hat{\theta})$ are equal to zero). ∎

According to the discussion of Section 2 the Z-models which could provide with efficient classifiers are those that can approximately provide type 1 solutions with high log likelihood value. Finally, in order to demonstrate that an appropriate Z-model can provide with better classification performance compared to common components and separate mixtures we present an illustrative artificial example (Fig. 3).

This example problem is actually a combination of the problems presented in Section 2. The first class data form three clusters, while the second class data form two clusters. The problem is constructed in such a way that there exists a pair of differently labeled clusters with significant overlap and a pair with slight overlap. We assume that the total number of components is three.

The solution found by the common component model (Fig. 3a) represents sufficiently the region with high class overlap and insufficiently the region with slight class overlap (places a component at the decision boundary). In what concerns the separate mixtures models two possibilities exist: i) using two components for the first class density and one for the second class (Fig. 3b) and ii) using one component for the first class density and two for the second class(Fig. 3c). The best classification model is provided by a Z-model with one component common to both classes and the remaining two components allocated one component per class (Fig. 3d).

4 A method that selects a Z-model

From the above discussion it is clear that the problem we have to overcome is related with the identification of the modeling case (Z matrix) which for fixed number of components provides good generalization performance. One approach is to exhaustively explore the binary space of z_{jk} variables: for each specification of Z, construct the appropriate classification model using the EM

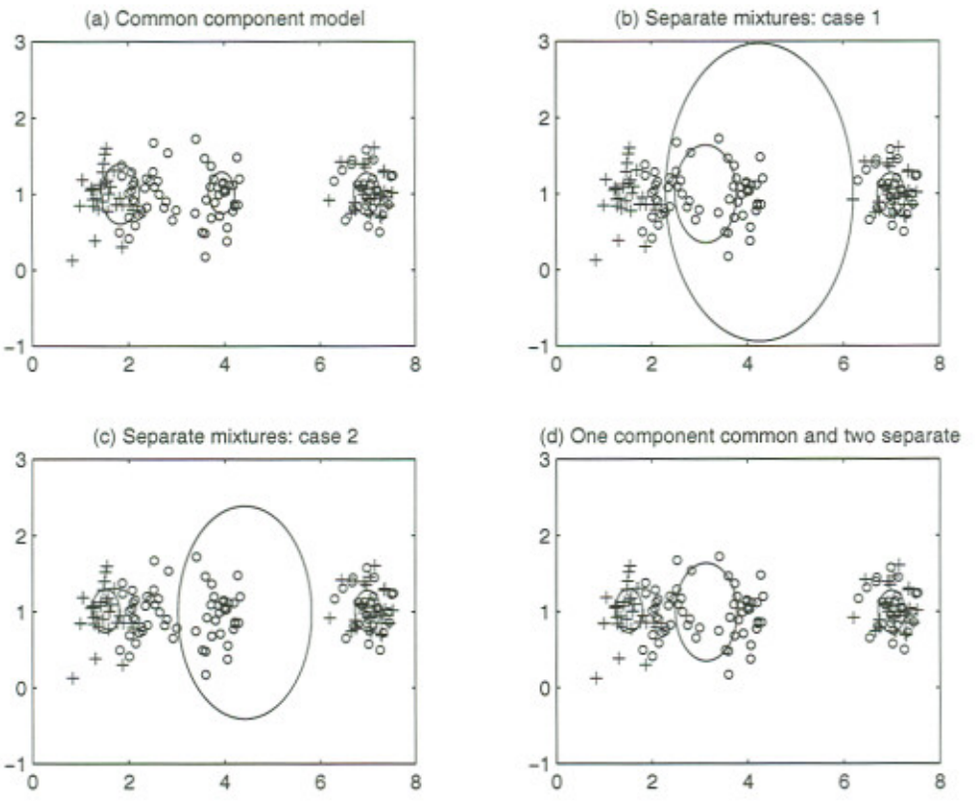

Figure 3: An example problem where an appropriate Z-model model leads to better classification. The data of each class was drawn according to $p(x|C_1) = 0.33N([2.3\ 1]^T, 0.08) + 0.33N([4\ 1]^T, 0.08) + 0.33N([7\ 1]^T, 0.08)$ and $p(x|C_2) = 0.5N([1.5\ 1]^T, 0.08) + 0.5N([7\ 1]^T, 0.08)$, while the prior class probabilities were $P(C_1) = P(C_2) = 0.5$. Two data sets were generated one for training and one for testing and for each model we found the maximum likelihood estimate. The generalization error e and the final log likelihood value L computed for the four models are: a) Common components: $e = 33.33\%$ and $L = -1754.51$ b) Separate mixtures (two components for C_1 and one for C_2): $e = 24.33\%$ $L = -2683.25$, c) Separate mixtures (one component for C_1 and two for C_2): $e = 34\%$ and $L = -3748.42$ and d) One component common and the other two separate (one per class): $e = 21.67\%$ and $L = -1822.53$.

algorithm, and rank each model (using for example, cross-validation) to select the best one.

However, it is computationally intractable to investigate all possible Z-models in the case of large values of M and K. In addition, this search approach also suffers from a practical drawback concerning component parameter initialization: if a mixture component is chosen to be common for some models, then it is reasonable that, at the start point of the EM training algorithm, this component should be placed at the region where class overlap occurs. This is not trivial since in general we are not aware of the type of overlapping among classes.

To deal with these problems, we have developed a computationally efficient training method that simultaneously adjusts both the model specification (model constraints) and the model parameters. To achieve this we relied on the consideration that the z_{jk} values actually constrain the parameter vector Θ to lie on a corresponding subspace of the broader parameter space corresponding to $z_{jk} = 1$ for all j, k. Therefore, we can develop a training method that initially assumes that the class conditional densities follow the broad model (1) and iteratively adjusts the constraints in the course of training to progressively define a subspace containing efficient solutions from the classification perspective. As noted in the previous sections, the basic guideline to obtain a good classification model is that a mixture component must not represent data from different classes unless it is placed at a region with significant class overlap. Our method attempts to make the training competitive between the class density models concerning component allocation. This can solve the problem related to common component in cases of slight overlap.

In order to define a training method that automatically adjusts the constraints during training, we define the constraint parameters r_{jk} where $0 \leq r_{jk} \leq 1$ and for each j satisfy:

$$\sum_{k=1}^{K} r_{jk} = 1. \tag{19}$$

The role of each parameter r_{jk} is analogous to z_{jk}: they specify the degree at which the kernel j is allowed to be used for modeling class C_k, that is to represent data of this class.

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

analogous to class conditional density models[3].

$$\varphi(x; C_k, r_k, \pi_k, \theta) = \sum_{j=1}^{M} r_{jk}\pi_{jk}p(x|j;\theta_j) \quad k = 1, \ldots, K. \tag{20}$$

Equation (20) is an extension of (1) with special constraint parameters r_{jk} incorporated in the linear sum. As it will become clear later, the parameters r_{jk} express the competition between classes concerning the allocation of each mixture component j.

The values of the priors π_{jk} satisfy by definition

$$\sum_{j=1}^{M} \pi_{jk} = 1 \;\; and \; if \; r_{jk} = 0 \; then \; \pi_{jk} = 0, \tag{21}$$

for each k. Therefore for every class C_k there must be at least one $r_{jk} > 0$. Thus, we exclude the case where a function φ and, consequently, the corresponding class conditional density is zero.

The functions φ in general do not constitute densities with respect to x due to the fact that $\int \varphi(x; C_k, r_k, \pi_k, \theta)dx \leq 1$, unless the constraints r_{jk} are assigned zero-one values[4] in which case they coincide with z_{jk} constraints. However, it generally holds that $\varphi(x; C_k, r_k, \pi_k, \theta) \geq 0$ and $\int \varphi(x; C_k, r_k, \pi_k, \theta)dx > 0$ which is due to (21).

In order to exploit the φ functions for learning the model parameters (Θ, r) (where r is the vector of all r_{jk}) it is necessary to treat them as if they were class conditional probability densities. In this spirit, we introduce an objective function analogous to the log likelihood function as follows:

$$L(\Theta, r) = \sum_{k=1}^{K} \sum_{x \in X_k} \log \varphi(x; C_k, r_k, \pi_k, \theta), \tag{22}$$

Through the maximization of the above function we adjust the values of the r_{jk} variables (actually the degree of component sharing) and this automatically influences the solution for the class density models parameter vector Θ.

The EM algorithm [2] can be used for maximizing the objective function (22). Although the EM algorithm is employed for log likelihood or log posterior maximization, the fact that our objective function in its general form does

[3]These functions do not represent the class density models to be constructed, but are used in order to find suitable parameters for the actual class density models (which are given by (1)).

[4]In this special case each function $\varphi(x; C_k, r_k, \pi_k, \theta)$ is identical to the corresponding $p(x|C_k; \pi_k, \theta)$.

not correspond to any of the above cases does not constitute a problem. In Appendix C it is shown that the basic EM property concerning the guaranteed monotone increase of the objective function still holds for the objective function $L(\Theta, r)$.

The Q function evaluated at the E-step of the $t+1$ EM iteration is:

$$Q(\Theta, r; \Theta^{(t)}, r^{(t)}) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j=1}^{M} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)}) \log\{r_{jk} \pi_{jk} p(x|j; \theta_j)\}, \tag{23}$$

where

$$\Phi_j(x; C_k, r_k, \pi_k, \theta) = \frac{r_{jk} \pi_{jk} p(x|j; \theta_j)}{\sum_{i=1}^{M} r_{ik} \pi_{ik} p(x|i; \theta_i)}. \tag{24}$$

The algorithm at the M-step maximizes the above function with respect the parameter vector (Θ, r). In Appendix B we present how the Q function (23) is derived and also the EM update equations for the case of Gaussian components.

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

$$\pi_{jk}^{(t+1)} = \frac{1}{|X_k|} \sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)}), \tag{25}$$

for each prior parameter π_{jk} and

$$r_{jk}^{(t+1)} = \frac{\sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)})}{\sum_{i=1}^{K} \sum_{x \in X_i} \Phi_j(x; C_i, r_i^{(t)}, \pi_i^{(t)}, \theta^{(t)})}, \tag{26}$$

for each constraint r_{jk}. Using the equation (25), equation (26) can be written as

$$r_{jk}^{(t+1)} = \frac{\pi_{jk}^{(t+1)} |X_k|}{\sum_{i=1}^{K} \pi_{ji}^{(t+1)} |X_i|}. \tag{27}$$

The above equation illustrates how the r_{jk} variables are adjusted at each EM iteration with respect to the newly estimated prior values π_{jk}. If we assume that the classes have nearly the same number of available training points, then during training each class C_k is constrained to use a component j to a degree specified by the ratio of the corresponding prior value π_{jk} over the sum of the rest of the priors associated with the same component j. In this way, the more a component j contributes to density of class C_k ie., the higher the value of π_{jk}, the greater the new value of r_{jk}, which causes in the next iteration the value of π_{jk} to become even higher (due to (25) and (24)) and so on. This explains how

the competition among classes for component allocation is realized through the adjustment of the constraints r_{jk}. According to this competition it is less likely for a component to be placed at some decision boundary since in such a case the class with more data in this region will attract the component towards its side. On the other hand, the method does not seem to significantly influence the advantage of the common component model in highly overlapped regions This can be explained from equation (24). In a region with high class overlap represented by a component j, the density $p(x|j;\theta_j)$ will essentially provide high values for data of all involved classes. Therefore, despite the fact that the constraint parameters might be higher for some classes, the Φ_j value (24) will still be high for data of all involved classes.

The EM algorithm performs iterations until convergence to some locally optimal parameter point (Θ^*, r^*). Then we use the r^*_{jk} values for Z-model selection, ie., to specify the z^*_{jk} values. A sensible choice is the following

$$z^*_{jk} = \begin{cases} 1 & \text{if } r^*_{jk} > 0 \\ 0 & \text{if } r^*_{jk} = 0 \end{cases} \tag{28}$$

The above specification of matrix Z^* is based on the argument that if $r^*_{jk} > 0$ the component j contributes to modeling of class C_k (since $\pi^*_{jk} > 0$) and, consequently, j must be incorporated in the mixture model representing C_k, the opposite holding when $r^*_{jk} = 0$. Once the Z^*-model has been specified the EM algorithm is run until convergence starting from parameter vector Θ^* to provide a finer tuning of the model parameters and provide the final parameter vector Θ_f that will be used for estimating the class densities using equation (1).

The above method was applied to the problem described in Section 3 (Fig. 3). The obtained solution Θ_f was exactly the one presented in Fig. 3d, where an appropriate selection for the Z-model was made. Remarkably, we found that $|\Theta_f - \Theta^*| = 0.03$ with the only difference being in the values of the prior parameters of the component representing the highly overlapped region[5].

4.1 Models with fixed r constraints

A special case of the algorithm presented above is obtained when the constraints r_{jk} are appropriately specified at the beginning and then remain fixed, ie. they

[5]We observed that in many problems Θ_f is very close to Θ^*, which means that maximizing (22) we provide a solution that is approximately a local maximum of (3). The essential condition for $\Theta_f = \Theta^*$ is that the parameters r^*_{jk} should have zero-one values. In such a case the obtained model corresponds to a separate mixtures model where the components have been dynamically partitioned among classes using the proposed training algorithm.

are not updated at each EM iteration. Such a model, called r-model can be considered analogous to the Z-model, where the Z matrix remains fixed during training. The distinction lies in the different assumptions imposed for the z_{jk} and r_{jk} constraints. For example any separate mixtures model with M total number of components can be described as special case of an r-model by properly defining the constraints r_{jk}, which in this case will obtain zero-one values.

Another model that results as a interesting special case of an r-model is the λPRBF model presented in [9]. Assume that the M available components are partitioned into K disjoint sets (groups) T_k, $k = 1, \ldots, K$, with each group T_k corresponding to class C_k and $|T_1| + \ldots + |T_K| = M$. The degree of component sharing is regulated by a parameter $\lambda \in [0, 1]$ in the following sense: we consider that the conditional density of class C_k utilizes fully the components of the group T_k, while the rest of the components are used at some degree λ. It must be noted that the parameter λ is specified in advance and remains fixed during EM training.

The λPRBF model can be considered as a special r-model obtained by specifying the constraints r as follows:

$$r_{jk} = \begin{cases} \frac{1}{1+\lambda(K-1)} & j \in T_k \\ \frac{\lambda}{1+\lambda(K-1)} & j \notin T_k \end{cases} \tag{29}$$

where with the expression $j \notin T_k$ we consider all components of the set $\bigcup_{k' \neq k} T_{k'}$. Under this assumption the objective function (22) takes the form

$$L(\Theta; \lambda) = \sum_{k=1}^{K} \sum_{x \in X_k} \log \varphi(x; C_k, \lambda, \pi_k, \theta) - |X| \log\{1 + \lambda(K-1)\}, \tag{30}$$

where

$$\varphi(x|C_k, \lambda, \pi_k, \theta) = \sum_{j \in T_k} \pi_{jk} p(x|j; \theta_j) + \lambda \sum_{j \notin T_k} \pi_{jk} p(x|j; \theta_j). \tag{31}$$

Since λ is fixed, the maximization of (30) concerns only the first term of the sum. The above equations describe the λPRBF model (31) and the objective function (30) maximized using an EM procedure [9].

It must be also noted that if $\lambda = 0$, (22) is a log likelihood corresponding to a separate mixtures model (the mixture density of class C_k uses the components of group T_k). If $\lambda = 1$, (22) corresponds to (3), that is to the log likelihood of the common mixture model.

The λPRBF model and training method is described in detail in [9] where an EM algorithm was derived for adjusting the model parameters Θ. In order to overcome the problem of appropriately specifying the value λ, an averaging approach was proposed over models obtained for different λ. It is obvious that method for Z^*-model selection presented in this work is more general and overcomes the problem of constraint specification through the update of constraints r_{jk} at each EM iteration.

5 Experimental results

To assess the performance of the proposed method for Z-model selection, we have conducted a series of experiment using Gaussian components and compare the common component model, the separate mixtures model and the Z^*-model. We tested several well-known classification data sets and also varied the total number of components. Some of the examined data sets (for example the Clouds data set) exhibit regions with significant class overlap, some other data sets contain regions with small class overlap, while the rest of them contain regions of both types. Following the discussion in sections 2 and 3, if the number of components is sufficient for data representation, then in the first type of problems the common component model is expected to provide better classification results, in the second type the separate mixtures model, while in the last case the Z^*-model is expected to behave better. In addition, due to its capability for adaptation to particular problem characteristics, the Z^*-model is expected to provide very good performance (either best or close to the best) for all types of problems.

We considered five well-known data sets namely the Clouds, Satimage and Phoneme from the ELENA database [4] and the Pima Indians and Ionosphere from the UCI repository [11]. For each dataset, number of components and model type, we employed the 5-fold cross-validation method in order to obtain an estimate of the generalization error. In the case of separate mixtures we considered an equal number of components used for the density model of each class. Also each Z^*-model was created starting from the same value for all constraints $r_{jk} = 1/K$. The use of EM procedures for the construction of all models resulted in training algorithms that are very fast and easy to implement.

Tables 1-5 display performance results (average generalization error and its standard deviation using 5-fold cross-validation) for the five data sets and

	4 components		6 components		8 components		10 components	
	error	std	error	std	error	std	error	std
Z^*-model	18.82	4.94	12.4	0.93	11.42	0.51	10.82	0.85
Common components	**13.06**	0.8	**11.12**	0.84	**11.32**	0.89	**10.42**	0.89
Separate mixtures	24.24	2.03	20.44	4.45	11.86	0.85	11.36	0.98

Table 1: Generalization error for the Clouds data set

	12 components		18 components		24 components	
	error	std	error	std	error	std
Z^*-model	12.33	0.5	11.4	0.74	11.1	0.75
Common Components	13.23	0.56	12.28	0.79	11.52	0.75
Separate mixtures	**12.05**	0.53	**11.21**	0.75	**10.98**	0.71

Table 2: Generalization error for the Satimage data set

	10 components		12 components		14 components	
	error	std	error	std	error	std
Z^*-model	17.96	1.14	**17.07**	1.01	**15.85**	1.19
Common components	20.62	0.75	20.03	0.75	20.98	1.04
Separate mixtures	**17.85**	1.4	17.37	0.75	16.88	1.15

Table 3: Generalization error for the Phoneme data set

	10 components		12 components		14 components	
	error	std	error	std	error	std
Z^*-model	27.08	2.6	26.92	3.26	**25.94**	2.27
Common components	29.95	3.06	28.12	2.21	28.25	1.97
Separate mixtures	**26.69**	3.58	**26.43**	1.34	27.08	2.22

Table 4: Generalization error for the Pima Indians data set.

	8 components		10 components		12 components	
	error	std	error	std	error	std
Z^*-model	**11.11**	2.3	**8.55**	2.4	**9.13**	3.92
Common component	15.11	3.85	9.41	3.35	9.27	3.21
Separate mixtures	11.82	1.89	12.24	3.77	9.39	3

Table 5: Generalization error for the Ionosphere data set

several choices on the total number of components. The experimental results clearly indicate that:

- Depending on the geometry of the data set and the number of available components either the common component model or the separate mixtures model may outperform one another.
- In all data sets the Z^*-model either outperforms the other models or exhibits performance that is very close to the performance of the best model. It must be noted that there were no case with the performance of the Z^*-model being inferior to both other models. This illustrates the capability of the proposed model and training algorithm to adapt to the geometry of each problem and efficiently utilize the available components to provide efficient classification systems.

It must also be noted that in all cases the solutions corresponding to the Z^*-model were approximately type 1 solutions of high likelihood. This fact constitutes an experimental validation of arguments presented in sections 2 and 3 concerning the quality of a solution from the classification perspective.

6 Conclusions and Future Research

We have generalized the idea of modeling class conditional densities by mixtures with common components [8, 9] relaxing the assumption that each component is used by all class density models. We focus on the objective of constructing effective classifiers through class density modeling An analysis was presented in Section 2 illustrating the comparative advantages and drawbacks of the common component and separate mixtures models and general criteria were specified concerning the preferable type of solutions from the classification point of view. In this spirit, the Z-model model formalism was introduced to express constraints on component sharing and an EM algorithm was presented to train a Z-model.

We also described a training method to create an efficient Z-model called Z^*-model. This is achieved through the introduction and adjustment of the constraint variables r_{jk} expressing the degree to which a component j is allowed to contribute to the model of class C_k. An EM algorithm was developed to simultaneously adjust both the constraint values and the component parameters. Experimental results using several well-known data sets indicate that the

proposed method for Z^*-model construction provides efficient solutions and is able to adjust component utilization among classes according to the geometry of the data set.

In what concerns future research, one direction is the development of alternative methods for Z-model specification using global optimization techniques tailored to the specific problem. Another research direction is the use of the Bayesian approach for model selection and training. This can be achieved through the introduction of a model prior $P(\Theta)$ expressing the preference towards a model solution based on the specifications discussed in Section 2. Finally it is also possible to examine techniques based on the combination (for example averaging) of multiple Z-models obtained for appropriately selected values of Z.

References

[1] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[2] A. P. Dempster, N. M. Laird and D. B. Rubin, 'Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm', *Journal of the Royal Statistical Society B*, vol. 39, pp. 1-38, 1977.

[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[4] Datasets and technical reports available via anonymous ftp from: ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases.

[5] G. J. McLachlan and K. Basford, *Finite Mixture Models*, Wiley, 2000.

[6] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Marcel Dekker, 1997.

[7] R. Redner and H. Walker, 'Mixture densities, Maximum Likelihood and the EM Algorithm', *SIAM Review*, vol. 26, no. 2, pp. 195-239, 1984.

[8] M. Titsias and A. Likas, 'A Probabilistic RBF network for Classification', *Proc. of International Joint Conference on Neural Networks*, Komo, Italy, July 2000.

[9] M. Titsias and A. Likas, 'Shared Kernel Models for Class Conditional Density Estimation', *IEEE Trans. on Neural Networks*, accepted with revision.

[10] D. M. Titterington, A. F. Smith and U.E. Makov, *Statistical Analysis of Finite Mixture Distributions*, Wiley, 1985.

[11] C. L. Blake and C. J. Merz, 'UCI repository of machine learning databases', University of California, Irvine, Dept. of Computer and Information Sciences, 1998.

[12] T. J. Hastie and R. J. Tibshirani, 'Discriminant Analysis by Gaussian Mixtures', *Journal of the Royal Statistical Society B*, vol. 58, pp. 155-176, 1996.

APPENDICES

Appendix A: EM algorithm for a Z-model. We first derive the update equations of the EM algorithm for a Z-model where the Z matrix is considered fixed.

To define the set of latent variables, we observe that for each class C_k the set of components used by the corresponding class density model is $S_k = \{j : z_{jk} = 1\}$. Hence, for each data point x of class C_k we define the $|S_k|$-dimensional latent vector $w(x)$ which indicates the component that generated x ($w_j(x) = 1$ if component j generated x and $w_i(x) = 0$, $i \neq j$)). The complete data log likelihood is

$$L_C(\Theta) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j:z_{jk}=1} w_j(x) \log\{\pi_{jk} p(x|j;\theta_j)\}. \tag{32}$$

The EM algorithm at the $t+1$ iteration uses the expectation of $L_C(\Theta)$ which is given by

$$Q(\Theta;\Theta^{(t)}) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j:z_{jk}=1} P(j|x, C_k; z_k, \pi_k^{(t)}, \theta^{(t)}) \log\{\pi_{jk} p(j|x;\theta_j)\}, \tag{33}$$

where we have replace $w_j(x)$ by its expected value $P(j|x, C_k; z_k, \pi_k^{(t)}, \theta^{(t)})$ (from equation (15)). The Q function can be written as

$$Q(\Theta;\Theta^{(t)}) = Q_1(\pi;\Theta^{(t)}) + Q_2(\theta;\Theta^{(t)}), \tag{34}$$

where π denotes all priors parameters and

$$Q_1(\pi;\Theta^{(t)}) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j:z_{jk}=1} P(j|x, C_k; z_k, \pi_k^{(t)}, \theta^{(t)}) \log \pi_{jk}, \tag{35}$$

$$Q_2(\theta;\Theta^{(t)}) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j:z_{jk}=1} P(j|x, C_k; z_k, \pi_k^{(t)}, \theta^{(t)}) \log p(x|j;\theta_j). \tag{36}$$

The two terms above can be maximized separately since they do not contain common parameters. If we assume that the mixture components are Gaussians of the general form

$$p(x|j;\mu_j,\Sigma_j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right\}, \tag{37}$$

then the maximization of $Q_1(\theta;\Theta^{(t)})$ (taking the derivatives and setting them equal to zero) leads to the following update equations

$$\mu_j^{(t+1)} = \frac{\sum_{k:z_{jk}=1}\sum_{x\in X_k} P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})x}{\sum_{k:z_{jk}=1}\sum_{x\in X_k} P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})}, \tag{38}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{k:z_{jk}=1}\sum_{x\in X_k} P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})(x-\mu_j^{(t+1)})(x-\mu_j^{(t+1)})^T}{\sum_{k:z_{jk}=1}\sum_{x\in X_k} P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)})}, \tag{39}$$

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

In order to maximize $Q_2(\pi;\Theta^{(t)})$ we must take into account the constraints (10), thus we introduce K Lagrange multipliers λ_k and the quantity to be maximized takes the form

$$\bar{Q}_2(\pi;\Theta^{(t)}) = Q_2(\pi;\Theta^{(t)}) - \sum_{k=1}^{K}\lambda_k\left(\sum_{j:z_{jk}=1}\pi_{jk} - 1\right). \tag{40}$$

Now, taking derivatives and setting them to zero we finally obtain

$$\pi_{jk}^{(t+1)} = \frac{1}{|X_k|}\sum_{x\in X_k} P(j|x,C_k;z_k,\pi_k^{(t)},\theta^{(t)}), \tag{41}$$

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

Appendix B: EM algorithm for Z^*-model selection. First the Q function computed at the E-step of the EM algorithm is derived. The objective function to be maximized is

$$L(\Theta,r) = \log P(X;\Theta,r) = \log\prod_{k=1}^{K}\prod_{x\in X_k}\sum_{j=1}^{M} r_{jk}\pi_{jk}p(x|j;\theta_j) \tag{42}$$

As noted in Section 4, since $P(X;\Theta,r)$ does not correspond to probability density (with respect to X), the objective function can be considered as the 'incomplete data log likelihood' in a broad sense. For each data point $x \in X$ we consider the latent variable $y(x) \in \{1,\ldots,M\}$ which indicates the mixture component which generated x. The whole vector of latent variables is denoted by $Y = (y(x^1),\ldots,y(x^N))$. Using the latent variable information, the 'complete data log likelihood' function is defined as follows

$$L_C(\Theta,r) = \log P(X,Y;\Theta,r) = \log\prod_{k=1}^{K}\prod_{x\in X_k}\{r_{y(x)k}\pi_{y(x)k}p(x|y(x);\theta_{y(x)})\}. \tag{43}$$

It can be shown that for the functions $P(X;\Theta,r)$ and $P(X,Y;\Theta,r)$ the following relation holds

$$P(X;\Theta,r)=\sum_{Y}P(X,Y;\Theta,r), \tag{44}$$

where the sum is over all possible values of the vector Y. In addition, we define the function

$$P(Y;X,\Theta,r)=\frac{P(X,Y;\Theta,r)}{P(X;\Theta,r)}, \tag{45}$$

which is the *probability density* of the latent variables Y conditioned on observed data X and the parameters Θ (since $\sum_Y P(Y;X,\Theta,r)=1$ due to (44)). Now, using (42), (43) and (24) $P(Y;X,\Theta,r)$ can also be written as

$$P(Y;X,\Theta,r)=\prod_{k=1}^{K}\prod_{x\in X_k}\frac{r_{y(x)k}\pi_{y(x)k}p(x|y(x);\theta_{y(x)})}{\sum_{j=1}^{M}r_{jk}\pi_{jk}p(x|j;\theta_j)}=\prod_{k=1}^{K}\prod_{x\in X_k}\Phi_{y(x)}(x;C_k,r_k,\pi_k,\theta). \tag{46}$$

Assume now that the algorithm is at iteration $t+1$ and the current parameter vector is $(\Theta^{(t)},r^{(t)})$. Then we define the function Q that is the expectation of $\log P(X,Y;\Theta,r)$ with respect to the probability density $P(Y;X,\Theta^{(t)},r^{(t)})$

$$Q(\Theta,r;\Theta^{(t)},r^{(t)})=\sum_{Y}\log\{P(X,Y;\Theta,r)\}P(Y;X,\Theta^{(t)},r^{(t)}) \tag{47}$$

and using (46)

$$Q(\Theta,r;\Theta^{(t)},r^{(t)})=\sum_{Y}\log\{P(X,Y;\Theta,r)\}\prod_{k=1}^{K}\prod_{x\in X_k}\Phi_{y(x)}(x;C_k,r_k^{(t)},\pi_k^{(t)},\theta^{(t)}). \tag{48}$$

Substituting for $\log P(X,Y;\Theta,r)$ according to (43) we find

$$Q(\Theta,r;\Theta^{(t)},r^{(t)})= \tag{49}$$
$$\sum_{Y}\left\{\sum_{k=1}^{K}\sum_{x\in X_k}\log\{r_{y(x)k}\pi_{y(x)k}p(x|y(x),\theta_{y(x)})\}\prod_{i=1}^{K}\prod_{x\in X_i}\Phi_{y(x)}(x;C_i,r_i^{(t)},\pi_i^{(t)},\theta^{(t)})\right\}$$

and using the Kronecker delta symbol

$$Q(\Theta,r;\Theta^{(t)},r^{(t)})= \tag{50}$$
$$\sum_{Y}\left\{\sum_{k=1}^{K}\sum_{x\in X_k}\sum_{j=1}^{M}\delta_{jy(x)}\log\{r_{jk}\pi_{jk}p(x|j,\theta_j)\}\prod_{i=1}^{K}\prod_{x\in X_i}\Phi_{y(x)}(x;C_i,r_i^{(t)},\pi_i^{(t)},\theta^{(t)})\right\}.$$

Using the fact that $\sum_{j=1}^{M}\Phi_j(x;C_k,r_k,\pi_k,\theta)=1$ equation (50) takes the final

form[6]

$$Q(\Theta, r; \Theta^{(t)}, r^{(t)}) = \sum_{k=1}^{K} \sum_{x \in X_k} \sum_{j=1}^{M} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)}) \log\{r_{jk} \pi_{jk} p(x|j; \theta_j)\}. \tag{51}$$

In Appendix C it is shown that an EM procedure which at the E-step of each iteration computes function Q and at the M-step maximizes Q with respect to Θ and r, guarantees the monotone increase of the objective function $L(\Theta, r)$. Next, we derive analytical solutions for the maximization step of the algorithm.

The function Q (51) can be written as the sum of three terms

$$Q(\Theta, r; \Theta^{(t)}, r^{(t)}) = Q_1(r; \Theta^{(t)}) + Q_2(\pi; \Theta^{(t)}) + Q_3(\theta; \Theta^{(t)}), \tag{52}$$

where, in analogy with Appendix A, the only adjustable parameters in $Q_1(r; \Theta^{(t)})$ are the constraints r, in $Q_2(\pi; \Theta^{(t)})$ the priors parameters and in $Q_3(\theta; \Theta^{(t)})$ the mixture component parameters. Obviously each term can be maximized independently. Assuming Gaussian components the terms $Q_1(\theta; \Theta^{(t)})$ and $Q_2(\pi; \Theta^{(t)})$ can be maximized in a way similar to that described in Appendix A and we finally obtain for each component j

$$\mu_j^{(t+1)} = \frac{\sum_{k=1}^{K} \sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)}) x}{\sum_{k=1}^{K} \sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)})}, \tag{53}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{k=1}^{K} \sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)})(x - \mu_j^{(t+1)})(x - \mu_j^{(t+1)})^T}{\sum_{k=1}^{K} \sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)})}, \tag{54}$$

$$\pi_{jk}^{(t+1)} = \frac{1}{|X_k|} \sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)}) \quad k = 1, \ldots, K. \tag{55}$$

To maximize the term $Q_3(r; \Theta^{(t)})$ we introduce M Lagrange multipliers λ_j (in order to ensure that the condition (19) is satisfied) and the quantity to be maximized is

$$\bar{Q}_3(r; \Theta^{(t)}) = Q_3(r; \Theta^{(t)}) - \sum_{j=1}^{M} \lambda_j \left(\sum_{k=1}^{K} r_{jk} - 1 \right). \tag{56}$$

Taking the derivatives equal to zero and after performing some algebra we finally find the following update equation

$$r_{jk}^{(t+1)} = \frac{\sum_{x \in X_k} \Phi_j(x; C_k, r_k^{(t)}, \pi_k^{(t)}, \theta^{(t)})}{\sum_{i=1}^{K} \sum_{x \in X_i} \Phi_j(x; C_i, r_i^{(t)}, \pi_i^{(t)}, \theta^{(t)})}, \tag{57}$$

[6] An analogous derivation for mixtures models is described in [1], pp. 69-72.

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

Appendix C: Proof that the EM algorithm described in Section 4 increases monotically the objective function. We give a simple proof that the EM algorithm described in Section 4 guarantees the increase of the objective function (22) at each iteration until a local maximum is reached. If we multiply and divide the argument of the logarithm in equation (47) with $P(X;\Theta,r)$, the Q computed at the E-step is written as

$$Q(\Theta, r; \Theta^{(t)}, r^{(t)}) = \sum_{Y} \log \left\{ \frac{P(X,Y;\Theta,r)P(X;\Theta,r)}{P(X;\Theta,r)} \right\} P(Y; X, \Theta^{(t)}, r^{(t)}). \tag{58}$$

By splitting the logarithm and using (45) we have

$$Q(\Theta, r; \Theta^{(t)}, r^{(t)}) = \sum_{Y} \log\{P(X;\Theta,r)\}P(Y;X,\Theta,r) + \sum_{Y} \log\{P(Y;X,\Theta,r)\}P(Y;X,\Theta^{(t)},r^{(t)}). \tag{59}$$

Subsequently, using (42) and the fact that $\sum_Y P(Y;X,\Theta,r) = 1$ we obtain

$$Q(\Theta, r; \Theta^{(t)}, r^{(t)}) = L(\Theta, r) + \sum_{Y} \log\{P(Y;X,\Theta,r)\}P(Y;X,\Theta^{(t)},r^{(t)}). \tag{60}$$

Suppose now that at the M-step of the algorithm we find a parameter vector $(\Theta^{(t+1)}, r^{(t+1)})$ such that $Q(\Theta^{(t+1)}, r^{(t+1)}; \Theta^{(t)}, r^{(t)}) \geq Q(\Theta^{(t)}, r^{(t)}; \Theta^{(t)}, r^{(t)})$, (this assumption concerns the most general case of GEM [6]). Then we can write that

$$L(\Theta^{(t+1)}, r^{(t+1)}) - L(\Theta^{(t)}, r^{(t)}) + \sum_{Y} \log \left\{ \frac{P(Y;X,\Theta^{(t+1)},r^{(t+1)})}{P(Y;X,\Theta^{(t)},r^{(t)})} \right\} P(Y;X,\Theta^{(t)},r^{(t)}) \geq 0 \tag{61}$$

The sum in the above equation cannot be positive according to the well-known Jensen's inequality (see [1], page 66). Thus, we find that $L(\Theta^{(t+1)}, r^{(t+1)}) \geq L(\Theta^{(t)}, r^{(t)})$.