

UC San Diego

Technical Reports

Title

Supervised dimensionality reduction using mixture models

Permalink

<https://escholarship.org/uc/item/2tz161zs>

Authors

Sajama, Sajama
Orlitsky, Alon

Publication Date

2004-12-27

Peer reviewed

Supervised dimensionality reduction using mixture models

Sajama, Alon Orlitsky

{sajama, alon}@ucsd.edu

Abstract

Given a classification problem, our goal is to find a low-dimensional linear transformation of the feature vectors which retains information needed to predict the class labels. We present a method based on maximum conditional likelihood estimation of mixture models. Use of mixture models allows us to approximate the distributions to any desired accuracy while use of conditional likelihood as the contrast function ensures that the selected subspace retains maximum possible mutual information between feature vectors and class labels. Classification experiments using Gaussian mixture components show that this method compares favorably to related dimension reduction techniques. Other distributions belonging to the exponential family can be used to reduce dimensions when data is of a special type, for example binary or integer valued data. We provide an EM-like algorithm for model estimation and present visualization experiments using both the Gaussian and the Bernoulli mixture models.

1 Introduction

Dimensionality reduction is a frequently used pre-processing step for supervised learning tasks. Reducing dimensions may improve classifier performance since it can suppress noise in the data and act as a form of regularization. Also, meaningful low dimensional representation can help in visualizing data sets and is an important tool in exploratory data analysis.

In this paper, we consider the problem of finding discriminative linear feature transformations. Given a collection of d -dimensional training samples and corresponding class labels, the goal is to find an L -dimensional hyperplane in \mathbb{R}^d such that the projected samples belonging to various classes are well separated. Our approach to this problem, termed supervised dimensionality reduction using mixture models (SDR-MM), is to model each class using a mixture model. The parameters of the model include affine parameters for a subspace to which the mixture means are constrained. Gaussian mixtures can approximate arbitrarily complex densities by lowering the minimum allowed variance and increasing the number of mixture components. Hence, this approach is *semi-parametric* - the subspace is determined by a set of affine parameters, while the distributions on the projected space are approximated non-parametrically. We use maximum conditional likelihood (MCL) estimation to determine the parameters of the lower dimensional subspace which ensures that the predictive information in the feature vectors is retained in the projected space. MCL has been widely used as a discriminative objective function for estimating hidden markov models in speech recognition and for Gaussian mixture models in the context of classification in [JP98].

Some dimension reduction methods make restrictive parametric assumptions about the distributions. For example, Fisher's linear discriminant analysis (LDA) can be obtained by maximum likelihood estimation assuming that the classes are Normally distributed with a common covariance matrix and different means, with the means constrained to lie in an L dimensional subspace. Other parametric methods include projection pursuit regression [FS81] and Generalized additive models [HT86].

More recently, several semi-parametric methods have been proposed for supervised dimensionality reduction including sliced inverse regression [Li91] and principal Hessian directions (pHd) [Li92]. In terms of the density model used, the method most closely related to SDR-MM is Mixture discriminant analysis (MDA) [HT96] which generalizes LDA by approximating each of the classes by a mixture of Gaussians all of which have a common covariance matrix. SDR-MM differs from MDA in two important ways. Firstly, in SDR-MM, we use spherical Gaussian distributions while in MDA each Gaussian has the same full-covariance matrix. While this may mean that SDR-MM needs to use more mixture components for each class, the total number of parameters to be estimated is often reduced from not having to estimate the d^2 parameters of the covariance matrix. The second difference is that in MDA, parameters are estimated using maximum likelihood, while in SDR-MM, the parameters are estimated discriminatively by maximizing the conditional likelihood which also eliminates the need for subclass shrinkage used in MDA.

The other dimensionality reduction method closely related to SDR-MM is kernel dimensionality reduction (KDR) [FBJ04] which also chooses the lower dimensional subspace based on maximum mutual information principle. SDR-MM differs from KDR in the way in which it measures the mutual information. While SDR-MM uses conditional likelihood, the KDR objective function is based on cross-covariance operators on reproducing kernel Hilbert spaces. Another related method was proposed in [TC00] in which instead of using the Shannon mutual information, a method is given to estimate a Renyi-entropy based expression for mutual information.

Our approach using Gaussian components can also be viewed as a prototype method with Euclidean distance as measure of similarity. Recently, there has been a lot of work on a probabilistic formulation of principal component analysis and its extension using the exponential family of distributions [Tip99, CDS01, KG01, SO04b]. The idea common to these extensions is that an appropriate Bregman distance (corresponding to various exponential family distributions) should be used in place of Euclidean distance if data is of a special type, for example binary or integer valued. In SDR-MM also, we allow the mixture components to be drawn from the exponential family in order to allow the method to be suitable for the various data types. SDR-MM is an adaptation of the unsupervised method - semi-parametric principal component analysis (SP-PCA) [SO04a, SO04b] to the supervised scenario.

We describe an EM-like algorithm for model estimation which uses iteratively re-weighted least squares in the maximization step. We present classification experiments which show that SDR-MM compares favorably to three related methods - pHd, MDA and KDR. We also show visualization examples for real-valued and binary data.

2 Density model with Gaussian mixture components

We are concerned with multi-class supervised problems where the feature vectors \mathbf{x} lie in \mathbb{R}^d and the class labels y are drawn from the set $\{1, \dots, M\}$. We are given training data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, which are independently and identically distributed samples from a probability distribution $P(y)P(\mathbf{x}|y)$. Each class m is modelled by a mixture of c_m number of Gaussians $\mathcal{N}(\mathbf{x}|\boldsymbol{\theta}, \sigma\mathbf{I})$ (σ common to all classes). Let $c = \sum_{m=1}^M c_m$ be total number of mixture components over all classes, $\Pi = \{\pi_1, \dots, \pi_c\}$ be the prior over these components and for each $k \in \{1, \dots, c\}$, let $\psi_k(m)$ be given by

$$\psi_k(m) = \begin{cases} 1 & \text{if mixture component } k \in \text{class } m \\ 0 & \text{otherwise} \end{cases}$$

Let $D(\mathbf{x}, \mathbf{w})$ denote the squared Euclidean distance between \mathbf{x} and \mathbf{w} . The distribution specified by this model is

$$\begin{aligned}
 P(Y = m) &= \sum_{k=1}^c \psi_k(m) \pi_k \\
 P(\mathbf{x}, Y = m) &= \sum_{k=1}^c \pi_k \psi_k(m) (2\pi)^{-d/2} e^{-D(\mathbf{x}, \boldsymbol{\theta}_k)/2\sigma^2}
 \end{aligned}$$

In order to obtain low dimensional representation and measure discriminative capability of feature transformations, we consider the *constrained* Gaussian mixture model. The means of Gaussians from all classes are restricted to lie in a lower (L) dimensional hyperplane in \mathbb{R}^d . We represent this constraint on mixture parameters using $L \times d$ rotation matrix V and d -dimensional displacement vector b . Each mean $\boldsymbol{\theta}_k$ belonging to this hyperplane can be represented by the L dimensional vector \mathbf{a}_k

$$\boldsymbol{\theta}_k = \mathbf{a}_k V + \mathbf{b}$$

We use the matrix A , whose k 'th row is \mathbf{a}_k , to represent the mixture component parameters. Hence our model is parameterized by $\Theta = \{\Pi, \psi, A, V, b\}$. The assumption that the mixture components are spherical Gaussians with common variance ensures that we measure the discriminative capabilities of *linear* projection, since the direction perpendicular to the plane (V, b) is irrelevant in any metric involving relative values of likelihoods $P(\mathbf{x}|\boldsymbol{\theta}_k)$. To see why this is the case, consider \mathbf{x}_p , the point on the hyperplane (V, b) closest to \mathbf{x} . Now, $P(\mathbf{x}|\boldsymbol{\theta}_k) \propto \exp(-\{D(\mathbf{x}, \mathbf{x}_p) + D(\mathbf{x}_p, \boldsymbol{\theta}_k)\}/2\sigma^2)$ and for a fixed \mathbf{x} , the factor involving $D(\mathbf{x}, \mathbf{x}_p)$ is common to all $\boldsymbol{\theta}_k$'s on the hyperplane (V, b) and hence cancels out.

Like LDA and MDA, there is an inherent classifier associated with the model we train for reducing dimensions. Since each class is modelled by a mixture, the distribution $P(y = m|\mathbf{x})$ can be obtained using Bayes rule and used to label any given test vector \mathbf{x} .

2.1 Relationship to radial basis function networks

The conditional distribution $P(y = m|\mathbf{x})$ is obtained from the model using Bayes formula

$$\begin{aligned}
 P(y = m|\mathbf{x}) &= \sum_{k=1}^c \psi_k(m) \frac{\pi_k e^{-D(\mathbf{x}, \boldsymbol{\theta}_k)/2}}{\sum_{k'=1}^c \pi_{k'} e^{-D(\mathbf{x}, \boldsymbol{\theta}'_{k'})/2}} \\
 &= \sum_{k=1}^c \psi_k(m) P(\boldsymbol{\theta}_k|\mathbf{x})
 \end{aligned}$$

This expression corresponds to a radial basis function (RBF) network in which the normalized basis functions are given by $\phi_k(\mathbf{x}) = P(\boldsymbol{\theta}_k|\mathbf{x})$ and the second layer weights are given by $\psi_k(m) = P(y = m|\boldsymbol{\theta}_k)$. RBF networks have been a topic of extensive research since their use can be motivated from the point of view of function approximation, regularization, noisy interpolation, density estimation, optimal classification theory and potential functions [Bis95]. The important distinction between our approach and typical RBF network applications lies in the training procedure used. We perform simultaneous supervised training of the first and second layer weights whereas in RBF networks, typically the first layer weights are picked using unsupervised techniques.

2.2 Use of spherical Gaussians

We have already noted that use of fixed-variance spherical Gaussians corresponds to measuring discriminative capability of a *linear* subspace when training samples are projected onto it. That sphericity is not a restrictive assumption follows from the universal approximation property of RBF networks with spherical gaussian kernels [PS91]. The idea is that spread of a given class along the subspace (V, b) can be approximated by spread of Gaussian means belonging to that class, assuming that a small enough variance is chosen. Use of full covariance matrices makes it necessary to regularize model estimation by penalizing the objective function. The assumption that all Gaussians have common spherical covariance reduces the number of parameters to be estimated by $\mathcal{O}(d^2)$ and thereby improves model generalization.

Our method is a soft equivalent of prototype methods like LVQ and its probabilistic nature allows data to simultaneously influence multiple prototypes - attracting prototypes of the same class and repelling prototypes belonging to a different class during MCL estimation - thereby generating a large-margin like effect. This provides a simple alternative to *subclass shrinkage* used in MDA [HT96]. There is a tradeoff between regularization and approximation capability - smaller variance is better for approximation and larger variance for the regularization effect described above.

3 The objective function

We propose using conditional likelihood of the training data as the objective function for choosing appropriate feature transformations, i.e., we pick the lower dimensional space specified by (V, b) using MCL estimation.

$$(V_{opt}, b_{opt}) = \arg \max_{(V, b)} \max_{A, \Pi} \prod_{i=1}^n P(y_i | \mathbf{x}_i, \Theta) \quad (1)$$

Use of this objective function can be motivated in several ways. In a classification problem, we are interested in finding a model which approximates the observed empirical conditional distribution $P_{emp}(y|\mathbf{x})$. Maximizing conditional likelihood is equivalent to minimizing the KL divergence between $P_{emp}(y|\mathbf{x})$ and the model $P_{(V, b)}(y|\mathbf{x})$. Also, on a related note, MCL estimation is equivalent to maximum mutual information estimation [JP98, KJO03]. Hence, using this objective function is equivalent to picking those transformations that preserve maximum amount of the relevant information (under our model) between the distributions of \mathbf{x} and y .

We present simple examples of projecting two-dimensional samples onto a line to illustrate how MCL estimation extends the applicability of past techniques based on constrained mixture of Gaussians. Figure 1 shows a two class example where each class is a mixture of four spherical Gaussians. Projection using low-rank ML estimation fully merges samples from the two classes while MCL estimated mixture model is able to find the best discriminant (see also [JP98]).

Figure 2 shows an interesting example where each of the two classes are generated by a single Gaussian with almost the same mean, but they have very different variance in one direction. If we used ML estimation with *no* constraints on the covariance matrices to find a one-dimensional subspace, we would get the ML solution subspace shown in Fig. 2 *even if each class is allowed to be modelled by a mixture of several Gaussians*. This is because no model can be better than the ‘true distribution’ in terms of likelihood of observed data (when data sample is large enough). However, since MDA imposes common covariance constraints on all mixture components of all classes, the MDA solution with three gaussian components for each class, coincides with the MCL solution in this case.

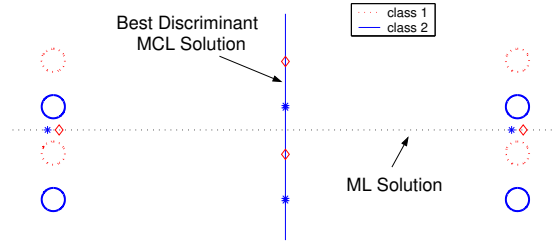


Figure 1: Illustrating advantage of MCL : Each class is a mixture of spherical Gaussians. \diamond and $*$ denote means of gaussian components of classes 1 and 2 respectively.

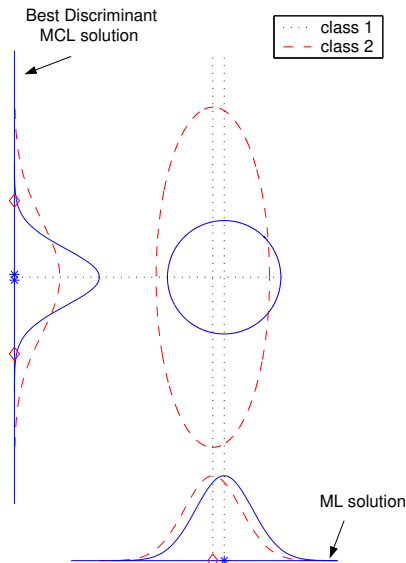


Figure 2: Illustrating advantage of MCL : Two classes with different covariance matrices. \diamond and $*$ denote means of gaussian components of classes 1 and 2 respectively.

Simulation studies [KJO03] have found that MCL classifiers can compete with and sometimes outperform other discriminative and generative classifiers. For fixed (V, b) , picking the Gaussian means which maximize conditional likelihood is equivalent to estimating a discriminative mixture classifier based on data projected onto the subspace given by (V, b) (see also Section 2). Hence optimizing the function (1) is equivalent to picking the best subspace for a discriminative Gaussian mixture classifier.

4 Exponential family distributions as mixture components

As we noted before, using Gaussian means and constraining them to a lower dimensional subspace of data space is equivalent to using a ‘soft’ prototype method where the prototypes are real valued and $D(\mathbf{x}, \boldsymbol{\theta})$, the distance between a point \mathbf{x} and prototype $\boldsymbol{\theta}$, is Euclidean. This distance measures the loss of information due to approximating \mathbf{x} using the prototypes. This Gaussian model may not be appropriate for other data types, for instance binary or integer data. The Bernoulli distribution may be better for binary data and Poisson for integer data. These three distributions, along with several others, belong to a family of distributions known as the *exponential family*

[MN83] and can be written in the form

$$\log P(x|\theta) = \log P_0(x) + x\theta - G(\theta)$$

Here, θ is called the *natural parameter* and $G(\theta)$ is a function that ensures that the probabilities sum to one. An important property of this family is that the mean μ of a distribution and its natural parameter θ are related through a monotone invertible, nonlinear function $\mu = g(\theta) = G'(\theta)$. It can be shown that the negative log-likelihoods of exponential family distributions can be written as Bregman distances (ignoring constants) which are a family of generalized metrics associated with convex functions [CDS01, AW01, FW00]. Several studies, in the area of unsupervised dimensionality reduction of special data types, have found that use of exponential family models yields better low dimensional representations [Tip99, CDS01, KG01, SO04b].

Hence we extend the model described in Section 2 by using the appropriate multivariate exponential family distributions for mixture components in the place of fixed variance Gaussians,

$$\log P(\mathbf{x}|\boldsymbol{\theta}) = \sum_{j=1}^d \{\log P_{0j}(x_j) + x_j\theta_j - G_j(\theta_j)\} \quad (2)$$

where x_j and θ_j are the j 'th components of \mathbf{x} and $\boldsymbol{\theta}$. Note that by using different distributions for different components of the feature vector \mathbf{x} , we can model mixed data types. The prototypes are now the natural parameters of these distributions and the measure of similarity is the negative log likelihood $-\log(P(\mathbf{x}|\boldsymbol{\theta}))$ which is equivalent to using sum of Bregman distances between components x_j and θ_j . We note that from the RBF network point of view, this extension corresponds to choosing different kernel functions (which determine region of influence of feature vectors) depending on the type of data.

5 Low dimensional representation

We discuss two of the several ways in which low dimensional representations can be obtained using the model Θ . The first method is to represent \mathbf{x} by that point $\boldsymbol{\theta}$ on (V, b) that is closest according to the appropriate Bregman distance (it can be shown that there is a unique such $\boldsymbol{\theta}_{opt}$ on the plane). This representation is a generalization of the standard Euclidean projection.

The second method of low dimensional representation is based on Bayes rule. Each feature vector \mathbf{x} induces a posterior distribution over the latent domain $P(\boldsymbol{\theta}_i|\mathbf{x}) = \pi_i P(\mathbf{x}|\boldsymbol{\theta}_i)/P(\mathbf{x})$. Under our model, all the information in \mathbf{x} about y is contained in this posterior distribution since y and \mathbf{x} are independent when conditioned upon the latent variable $\boldsymbol{\theta}$. Hence \mathbf{x} can be represented by a suitable function of this posterior and we choose to use the mean. This representation has been used in the unsupervised case successfully to get meaningful low dimensional views [TB97, Tip99, KG01, SO04b].

6 Algorithm

Several iterative algorithms have been proposed for MCL estimation of mixture models, see for example [JP98, KJO03]. The common thread in these algorithms is that each iteration involves evaluating a tight lower bound which touches the objective function at the current parameter value. Model parameters are then updated by maximizing this lower bound. This technique was called bound maximization in [JP98] and is the basis of many iterative algorithms including the expectation maximization (EM) algorithm.

We use the idea of bound maximization and derive an algorithm for MCL estimation under low rank constraint on mixture component parameters Θ . Let Θ^t and Θ^{t+1} denote the current

and updated parameter values at iteration t . The change in conditional log-likelihood at iteration t can be written as a difference of two terms

$$\begin{aligned}
\Delta l &= \sum_{i=1}^n \{\log P(y_i|\mathbf{x}_i, \Theta^{t+1}) - \log P(y_i|\mathbf{x}_i, \Theta^t)\} \\
&= \sum_{i=1}^n \log \frac{P(\mathbf{x}_i, y_i|\Theta^{t+1})}{P(\mathbf{x}_i, y_i|\Theta^t)} - \sum_{i=1}^n \log \frac{P(\mathbf{x}_i|\Theta^{t+1})}{P(\mathbf{x}_i|\Theta^t)} \\
&\geq \sum_{i=1}^n \sum_{k=1}^c \hat{z}_{ik} \log P(\boldsymbol{\theta}_k, \mathbf{x}_i, y_i|\Theta^{t+1}) \\
&\quad - \sum_{i=1}^n \rho_i P(\mathbf{x}_i|\Theta^{t+1}) + \text{constant}
\end{aligned}$$

where, $\rho_i = 1/P(\mathbf{x}_i|\Theta^t)$ and

$$\hat{z}_{ik} = \frac{P(\boldsymbol{\theta}_k, \mathbf{x}_i, y_i|\Theta^t)}{\sum_{k'=1}^c P(\boldsymbol{\theta}_{k'}, \mathbf{x}_i, y_i|\Theta^t)}$$

Here the first term was lower bounded using Jensen's inequality (similar to the EM algorithm) and the second term using $\log w \leq w - 1$. At each iteration, we compute the lower bound by computing \hat{z}_{ik} and ρ_i for $i = 1, \dots, n$ and $k = 1, \dots, c$. The lower bound is then optimized by alternately maximizing over each of Π , A , V and b while holding the rest of the parameters constant.

The lower bound can be written as (ignoring constants since they do not affect the optimization steps)

$$\begin{aligned}
\Delta l &= \sum_i \sum_k \hat{z}_{ik} \log \pi_k + \sum_i \sum_k \hat{z}_{ik} \log \psi_k(y_i) \\
&\quad + \sum_i \sum_k \hat{z}_{ik} \log P(\mathbf{x}_i|\boldsymbol{\theta}_k) - \sum_i \sum_k \rho_i \pi_k P(\mathbf{x}_i|\boldsymbol{\theta}_k)
\end{aligned} \tag{3}$$

UPDATING Π : Π^{t+1} is obtained by maximizing the Lagrangian (formed using terms in Δl involving π_k)

$$L = \sum_{k=1}^c \{c_{1k} \log \pi_k - c_{2k} \pi_k\} + \lambda \left(\sum_{k=1}^c \pi_k - 1 \right)$$

where, $c_{1k} = \sum_{i=1}^n \hat{z}_{ik}$, $c_{2k} = \sum_{i=1}^n \rho_i P(\mathbf{x}_i|\boldsymbol{\theta}_k)$ and λ is a lagrange multiplier used to impose the constraint that the latent distribution sums to one. This optimization is a little more complicated than its counterpart in the EM algorithm for ML estimation since we have both linear and logarithmic terms. Differentiating L and setting the derivative to zero, we get $\pi_k = c_{1k}/(c_{2k} - \lambda)$. We need to find λ that satisfies $f(\lambda) = \sum_{k=1}^c c_{1k}/(c_{2k} - \lambda) = 1$. There is no explicit solution for this equation, but it is easy to verify that at $\lambda_0 = \min_k(c_{2k} - c_{1k})$, $f(\lambda_0) > 1$ and that as $\lambda \rightarrow -\infty$, $f(\lambda) \rightarrow 0$. Moreover, $f(\lambda)$ is continuous and monotone in the region $[-\infty, \lambda_0]$ implying that there is a unique λ_{opt} such that $f(\lambda_{opt}) = 1$, which can be found using bisection line search.

OPTIMIZING A , V AND b : For optimizing A and V , we use an iterative weighted least squares method similar to that used in fitting generalized linear models [MN83], i.e., we apply the Newton-Raphson (NR) procedure to the equations obtained by setting the derivative of Δl to zero. Upon taking the first and second derivatives with respect to the components of the matrix A , it turns out that each row can be updated independently of the others in a given

iteration. This decoupling is convenient since it means that updating the parameters involves smaller matrix operations. Similarly, we find that each column of V and each component of b can be updated independently.

Δl depends on A , V and b only through the last two terms in Equation 3. So, ignoring constants, we want to maximize

$$\sum_{k=1}^c \sum_{j=1}^d (\boldsymbol{\theta}_{kj} \tilde{x}_{kj} - G(\boldsymbol{\theta}_{kj}) \tilde{z}_k) - \sum_{i=1}^n \sum_{k=1}^c \rho_i \pi_k P(\mathbf{x}_i | \boldsymbol{\theta}_k) \quad (4)$$

where, $\tilde{x}_{kj} = \sum_{i=1}^n \hat{z}_{ik} x_{ij}$ and $\tilde{z}_k = \sum_{i=1}^n \hat{z}_{ik}$ and $P(\mathbf{x}_i | \boldsymbol{\theta}_k)$ is as defined before in Equation 2.

Each row of A , \mathbf{a}_r is updated by adding $\delta \mathbf{a}_r$ which is calculated using $(V \Omega_r V^t) \delta \mathbf{a}_r = GR_r$, where the $d \times d$ matrix Ω_r and the $L \times 1$ matrix GR_r are given by

$$\begin{aligned} [\Omega_r]_{jj'} &= \left\{ \tilde{z}_r - \sum_{i=1}^n \rho_i \pi_r P(\mathbf{x}_i | \boldsymbol{\theta}_r) \right\} \frac{\partial g(\theta_{rj})}{\partial \theta_{rj}} \delta(j = j') \\ &\quad + \sum_{i=1}^n \rho_i \pi_r P(\mathbf{x}_i | \boldsymbol{\theta}_r) (x_{ij'} - g(\theta_{rj'})) (x_{ij} - g(\theta_{rj})) \\ [GR_r]_s &= \sum_{j=1}^d v_{sj} \tilde{x}_{rj} - \tilde{z}_r g(\theta_{rj}) \\ &\quad - \sum_{i=1}^n \rho_i \pi_r P(\mathbf{x}_i | \boldsymbol{\theta}_r) (x_{ij} - g(\theta_{rj})) \end{aligned}$$

Each column of the matrix V , \mathbf{v}_s is updated by adding $\delta \mathbf{v}_s$ obtained by solving $(A^t \Omega_s A) \delta \mathbf{v}_s = GR_s$, where the $c \times c$ diagonal matrix Ω_s , and the $L \times 1$ matrix GR_s are given by,

$$\begin{aligned} [\Omega_s]_{kk} &= \left\{ \tilde{z}_k - \sum_{i=1}^n \rho_i \pi_k P(\mathbf{x}_i | \boldsymbol{\theta}_k) \right\} \frac{\partial g(\theta_{ks})}{\partial \theta_{ks}} \\ &\quad + \sum_{i=1}^n \rho_i \pi_k P(\mathbf{x}_i | \boldsymbol{\theta}_k) (x_{is} - g(\theta_{ks}))^2 \\ [GR_s]_r &= \sum_{k'=1}^c a_{k'r} \{ \tilde{x}_{k's} - \tilde{z}_{k'} g(\theta_{k's}) \} \\ &\quad + \sum_{i=1}^n \rho_i \pi_{k'} P(\mathbf{x}_i | \boldsymbol{\theta}_{k'}) (x_{is} - g(\theta_{k's})) \} \end{aligned}$$

Each column of the row matrix b , b_s , is updated by adding δb_s obtained by solving $H_s \delta b_s = GR_s$, where the 1×1 matrices H_s and GR_s are given by

$$\begin{aligned} H_s &= \sum_{k=1}^c \left\{ \left\{ \tilde{z}_k - \sum_{i=1}^n \rho_i \pi_k P(\mathbf{x}_i | \boldsymbol{\theta}_k) \right\} \frac{\partial g(\theta_{ks})}{\partial \theta_{ks}} \right. \\ &\quad \left. + \sum_{i=1}^n \rho_i \pi_k P(\mathbf{x}_i | \boldsymbol{\theta}_k) (x_{is} - g(\theta_{ks}))^2 \right\} \end{aligned}$$

$$GR_s = \sum_{k'=1}^c \{ \tilde{x}_{k's} - \tilde{z}_{k'} g(\theta_{k's}) \\ \sum_{i=1}^n \rho_i \pi_{k'} P(\mathbf{x}_i | \theta_{k'}) (x_{is} - g(\theta_{k's})) \}$$

Note that using the NR method does not guarantee monotone increase in the value of \tilde{L} . Monotonicity can be enforced using standard optimization procedures like line search or the trust regions method [BGLS03].

COMPUTATIONAL COMPLEXITY : Time taken for each iteration of this algorithm is $\mathcal{O}(cdnL^2)$. Computing \hat{z}_{ik} and ρ_i involve computing $P(\mathbf{x}_i | \theta_k)$ which is expensive and is a common problem faced in maximum likelihood estimation and in training of RBF networks. [Omo87] gives a procedure for speeding up this procedure using the k-d tree data structure by identifying relevant prototypes (for each \mathbf{x}) thereby avoiding unnecessary computation.

7 Experiments

We experimented with the Gaussian mixture model on four real-valued datasets and with the Bernoulli mixture model on a binary set. As noted in Section 2.2, for the Gaussian mixture model, an appropriate variance should be chosen to achieve the right tradeoff between regularization and approximation capability. Also, the value of $P(\mathbf{x}_i | \theta_k)$ can become very small and lead to computational difficulties if the variance is chosen to be too small. In the experiments reported here, we used fixed variance Gaussians and the data was sphered. The variance was selected by trying a few values ranging between 0.5 and 2 and choosing the variance that maximized conditional log-likelihood (a part of the training set was used for validation). As with most iterative optimization methods, the model estimated by the SDR-MM algorithm depends on parameter initialization. We tried a few different random starts and chose the model which gives highest conditional log-likelihood on training data (validation was not used).

7.1 Classification results

We give classification results comparing SDR-MM with KDR, MDA and pHd. We modified the matlab package of Kernel ICA [Bac02] to obtain the KDR results. The variance parameter for KDR was gradually decreased (between iterations) to two as suggested in [FBJ04]. For the experiments with MDA and pHd, we used the *mda* and *dr* packages in the R language. We used four data sets from the UCI machine learning repository, viz. Heart disease, Ionosphere, Breast cancer and waveform data sets (summarized in Table 1).

Table 2 shows classification results obtained by first projecting data using the various methods and then using SVM to classify the projected data. For MDA and SDR-MM, we also give results using the inherent classifier that is obtained using the probability densities estimated in the process of finding the lower dimensional space. The classification rates shown in the table are averaged 10-fold cross validation results. The t-values of the paired significance tests comparing SDR-MM, MDA and KDR are given in Table 3. We found that SDR-MM performs significantly better than KDR on all of the data sets except one - the Ionosphere data. SDR-MM also did better than MDA consistently, but the significance t-values were not (on an average) as high as the comparison with KDR.

Table 1: Description of data sets for the classification problem.

DATA SET	DIM. OF X	TRAINING SAMPLE	TEST SAMPLE
HEART DISEASE	13	149	148
IONOSPHERE	34	151	200
BREAST CANCER - WISCONSIN	30	200	369
WAVEFORM	21	300	500

Table 2: Accuracies for classifiers associated with projection onto various lower dimensions.

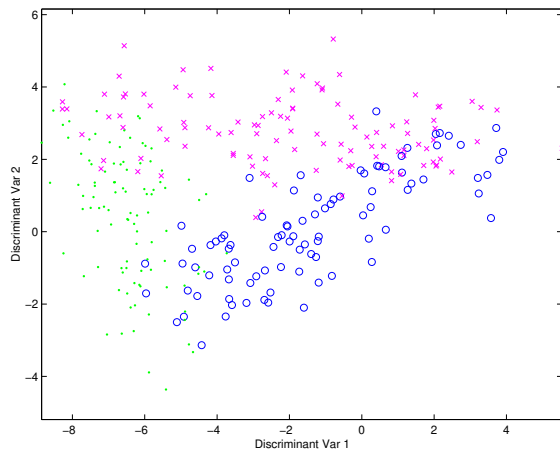
DATA SET	L	pHd	KDR	MDA		SDR-MM	
		SVM	SVM	SVM	INHERENT	SVM	INHERENT
HEART	1	52.37	80.68	77.84	78.31	80.81	81.28
	3	68.92	77.43	77.97	79.12	80.95	82.50
	5	73.31	76.82	80.74	77.97	81.49	82.03
IONOSPHERE	1	68.80	90.28	75.75	76.60	87.14	86.86
	3	82.75	95.28	86.9	85.55	89.71	87.71
	5	87.65	94.88	88.85	85.90	91.14	89.43
BREAST CANCER	1	73.88	93.82	92.55	93.31	95.50	95.80
	3	84.23	90.92	93.36	94.2	95.83	95.88
	5	90.41	88.59	93.88	94.39	95.85	96.02
WAVEFORM	1	-	59.32	60.58	61.00	60.98	61.58
	2	-	82.80	84.40	84.86	85.16	85.56
	4	61.6	79.08	83.78	83.10	84.36	84.88

Table 3: Calculated t-values for comparison between various dimension reduction methods followed by SVM classifier. Paired samples test of significance for 10-fold cross validation is significant with probability 0.05/0.01/0.001 if t-value is higher than 2.23/3.17/4.59, respectively. Positive/negative t-value means that the first/second classifier, respectively, is better than the other.

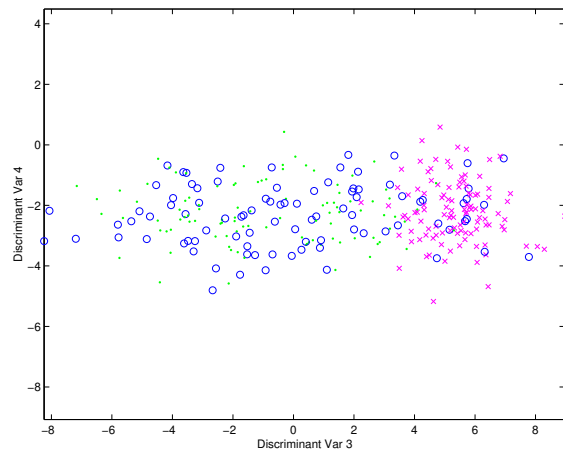
DATA SET	L	SDR-MM vs KDR	SDR-MM vs MDA	KDR vs MDA
HEART	1	0.13	0.90	0.70
	3	2.16	0.94	-0.17
	5	4.60	0.91	-2.82
IONOSPHERE	1	-1.62	3.44	6.06
	3	-3.34	1.94	7.37
	5	-2.78	1.18	7.06
BREAST CANCER	1	2.50	4.52	1.69
	3	4.12	4.00	-1.68
	5	5.23	2.44	-3.67
WAVEFORM	1	2.11	0.47	-1.40
	2	3.58	1.69	-4.18
	4	6.53	1.08	-6.06

7.2 Visualization - Gaussian case

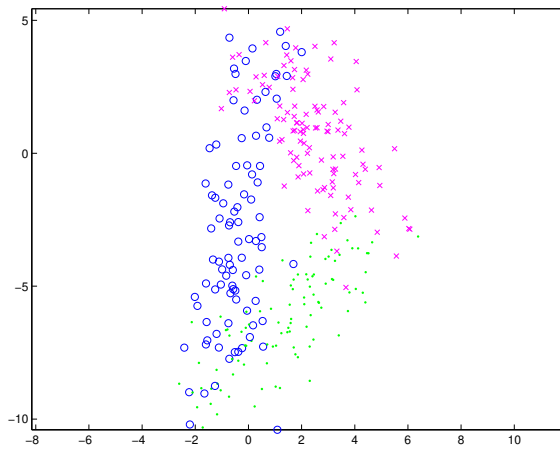
For the visualization experiment we used the Waveform data set. We trained a model with 30 Gaussian components (10 for each class) and with mean parameters constrained to a four-



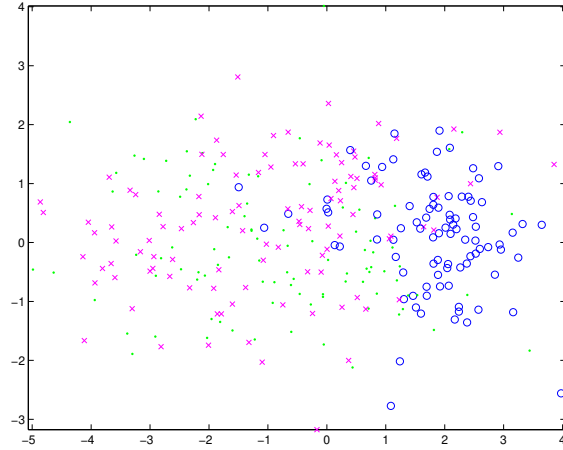
(a) Projection onto directions 1 and 2 - SDR-MM



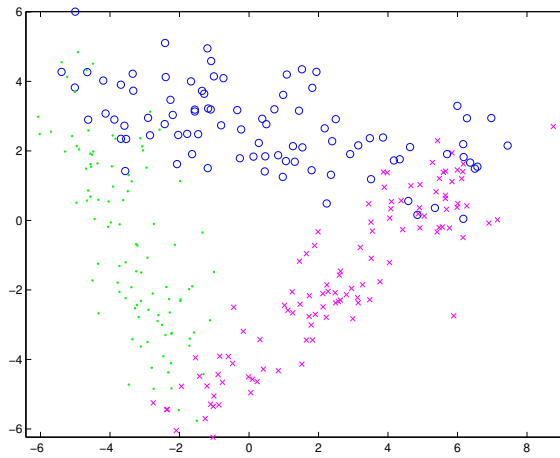
(b) Projection onto directions 3 and 4 - SDR-MM



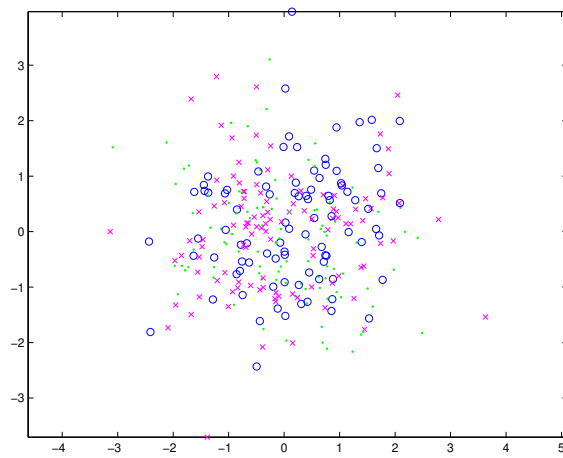
(c) Projection onto directions 1 and 2 - KDR



(d) Projection onto directions 3 and 4 - KDR



(e) Projection onto directions 1 and 2 - MDA



(f) Projection onto directions 3 and 4 - MDA

Figure 3: Some two dimensional views of waveform dataset projected onto the four basis vectors obtained using various methods

dimensional subspace. The estimated matrix V was processed using the Gram-Schmidt procedure to obtain orthogonal basis for the lower dimensional subspace and the training data was projected onto this subspace. Figures 3(a) and 3(b) show two views of this four-dimensional projected

set. The first two coordinates were sufficient to discriminate between the three classes since the two-dimensional model achieves an error rate close to the minimum possible (Bayes) error (see Table 2). However, we see that the third coordinate distinguishes one class from the other two, indicating that maximum mutual information based methods may be able to discover more discriminating information than what is needed for classification. KDR projection gave similar lower dimensional views, but with greater overlap among the three classes (Figures 3(c) and 3(d)). In the corresponding projections obtained using MDA, shown in Figures 3(e) and 3(f), the third and fourth discriminants do not significantly discriminate between the classes.

7.3 Visualization - Binary case

We demonstrate the binary data visualization capability of SDR-MM with Bernoulli conditional distribution. While performing the experiments we found that the algorithm was much more likely to get stuck in local minima when the Bernoulli mixture components are used than in the Gaussian case. The visualization shown in this section was obtained by running the SDR-MM algorithm several times and picking the best view. For this purpose, we use the ICU data set [LTAP88] which consists of a sample of 200 subjects who were part of a study on survival of patients following admission to an adult intensive care unit (ICU). We picked 190 patients (we only considered Caucasian and African-American patients) and 16 binary features from this data-set.

The goal is to extract features that predict whether a patient will leave the ICU alive. The features considered include presence or absence of coma, cancer, fracture and infection, the patient’s gender and race and whether the admission to ICU was elective or due to an emergency. The two dimensional projection obtained using MCL estimation of constrained Bernoulli mixture model is shown in Fig. 4(a). We examined the basis vectors of the lower-dimensional parameter space obtained using SDR-MM, and found that the features that change most significantly along the horizontal direction are the type of admission (elective versus emergency) and whether a fracture was involved. Along the vertical direction, the feature with maximum change is presence of cancer.

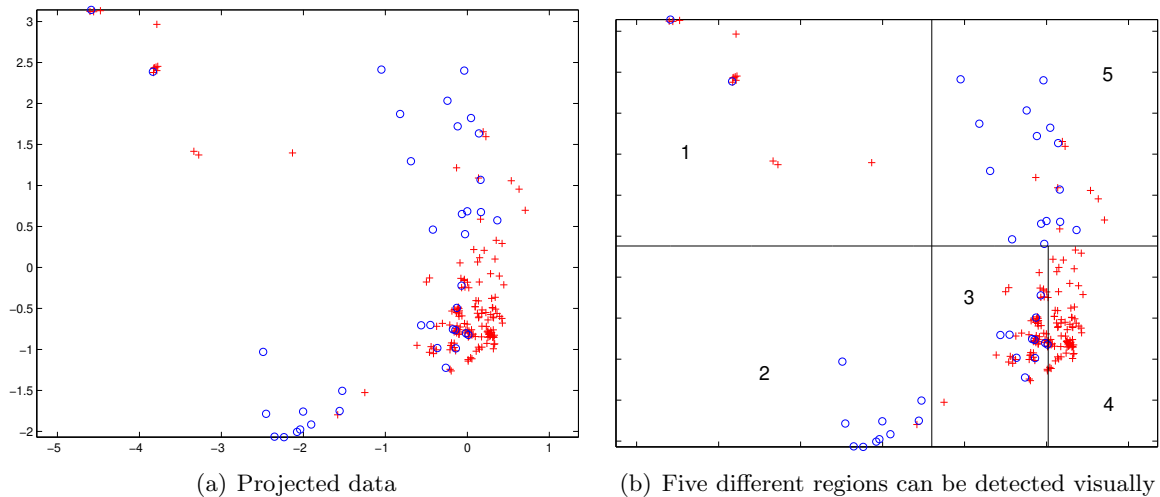


Figure 4: Two dimensional representation of binary data from the ICU data set : patients who left the ICU alive are shown by ‘+’ while the patients who died are shown by ‘o’.

The projected data can be visually divided into five clusters 4(b). Four of the clusters, numbered 1, 2, 4 and 5, were relatively ‘pure’, i.e., consist of either people who left the ICU alive

or those who did not, while cluster 3 consists of both types of people. To further understand the factors distinguishing the two groups, we compared the mean value of feature vectors corresponding to various pairs of these clusters. We found that the two '+' clusters (1 and 4) were distinguished mainly by presence of fracture, the type of admission and whether the patient was in coma. The two 'o' clusters (2 and 5) were distinguished by infection, coma, whether CPR was administered prior to admission and cancer. The most interesting comparison was between the clusters 3 and 4 which were distinguished mainly by whether the admission was elective or emergency and whether the service provided at admission was medical or surgical. Some conclusions that can be readily drawn from this are that people who elected to join ICU to receive medical attention survived with high probability. Among those who joined the ICU because of an emergency, those who joined because of a fracture survived with high probability (cluster 1), though some of these (presumably with severe damage) did not survive. The type of service at admission and type of admission are highly correlated for this cluster.

8 Conclusion

Semi-parametric PCA is a recently proposed probabilistic alternative to principal component analysis based on maximum likelihood estimation of latent variable models. In this paper, we argued that use of maximum conditional likelihood estimation is a natural way to extend this method to supervised problems. Experiments demonstrate the potential of this method to learn discriminating transformations and for supervised visualization of high dimensional data.

There are many promising directions for future work. Typically, supervised multi-class dimension reduction experiments involve learning directions which discriminate among all classes simultaneously. Finding projections suitable for separating pairs (or more generally subsets) of classes can give better discriminative directions. Outputs from these low-complexity classifiers can then be combined to obtain full classifiers with good performance. Another interesting extension would be to use mixture modelling approach with a suitable objective function for semi-supervised dimensionality reduction.

References

- [AW01] Katy S. Azoury and Manfred K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.
- [Bac02] Francis Bach. The kernel-ica package, <http://www.cs.berkeley.edu/~fbach/kernel-ica/index.htm>, 2002.
- [BGLS03] J. F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization*. Springer Verlag, 2003.
- [Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.
- [CDS01] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems*, volume 13, 2001.
- [FBJ04] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

- [FS81] Jerome H. Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the american statistical association*, 76:817–823, 1981.
- [FW00] Jurgen Forster and Manfred K. Warmuth. Relative expected instantaneous loss bounds. In *Computational Learning Theory*, pages 90–99, 2000.
- [HT86] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical Science*, 1:297–318, 1986.
- [HT96] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:158–176, 1996.
- [JP98] T. Jebara and A. Pentland. Maximum conditional likelihood via bound maximization and the CEM algorithm. In *Neural Information Processing Systems 11*, 1998.
- [KG01] A. Kaban and M. Girolami. A combined latent class and trait model for the analysis and visualization of discrete data. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(8):859–872, August 2001.
- [KJO03] Aldebaro Klautau, Nikola Jevtic, and Alon Orlitsky. Discriminative gaussian mixture models: A comparison with kernel classifiers. In *Proc. 20th International Conf. on Machine Learning*, pages 353–360, 2003.
- [Li91] Ker-Chau Li. Sliced inverse regression for dimension reduction (with discussion). *Journal of american statistical association*, 86:316–342, 1991.
- [Li92] Ker-Chau Li. On principal hessian directions for data visualization and dimension reduction: Another application of stein’s lemma. *Journal of american statistical association*, 87:1026–1039, 1992.
- [LTAP88] S. Lemeshow, D. Teres, J. S. Avrunin, and H. Pastides. Predicting the outcome of intensive care unit patients. *Journal of the American Statistical Association*, 83:348–356, 1988.
- [MN83] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1983.
- [Omo87] S. M. Omohundro. Efficient algorithms with neural networks behaviour. *Complex Systems*, 1:273–347, 1987.
- [PS91] J. Park and L. W. Sandberg. Universal approximation using radial basis function networks. *Neural Computation*, 3:246–257, 1991.
- [SO04a] Sajama and A. Orlitsky. Semi-parametric exponential family PCA : Reducing dimensions via non-parametric latent distribution estimation. Technical Report CS2004-0790, University of California at San Diego, <http://cwc.ucsd.edu/~sajama>, 2004.
- [SO04b] Sajama and Alon Orlitsky. Semi-parametric exponential family PCA. In *Advances in Neural Information Processing 17 (NIPS)*, December 2004.
- [TB97] M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, Birmingham, UK, September 1997.

- [TC00] Kari Torkkola and William M. Campbell. Mutual information in learning feature transformations. In *Proc. 17th International Conf. on Machine Learning*, pages 1015–1022, 2000.
- [Tip99] M. Tipping. Probabilistic visualisation of high-dimensional binary data, 1999.