Semi-parametric exponential family PCA : Reducing dimensions via non-parametric latent distribution estimation

Sajama, Alon Orlitsky

{sajama, alon}@ucsd.edu

June 2, 2004

Abstract

Principal component analysis is a widely used technique for dimensionality reduction, but is not based on a probability model. Many recently proposed dimension reduction methods are based on latent variable modelling with restrictive assumptions on the latent distribution. We present a semi-parametric latent variable model based technique for density modelling, dimensionality reduction and visualization. Unlike previous methods, we estimate the latent distribution non-parametrically. Using this estimated prior to reduce dimensions ensures that multi-modality is better preserved in the projected space. In addition, we allow the components of latent variable models to be drawn from the exponential family which makes the method suitable for special data types, for example binary or count data. We discuss connections to other probabilistic and non-probabilistic dimension reduction schemes based on gaussian and other exponential family distributions. Simulations on real valued, binary and count data show favorable comparison to other related schemes both in terms of separating different populations and generalization to unseen samples.

Keywords

Dimensionality reduction, Latent variables, Non-parametric prior estimation, Constrained mixture model, Exponential family, Principal components, Visualization, Clustering.

1 Introduction

Many machine learning applications involve high-dimensional data sets. Representing this high-dimensional data in lower-dimensional space is a pre-processing step performed before learning to classify the data or to discover structure in it. The reasons for reducing dimensions include difficulty of estimating densities and computational complexity of learning classifiers in high dimensions. Also, visualization requires representation in two or three dimensions.

Principal component analysis (PCA) is a widely used dimensionality reduction technique. Several of the proposed extensions/alternatives to PCA fall into one or more of the following three categories

- Alternatives based on approximating the data density by low dimensional probability models
- Extensions to deal with special data types for example binary or count data

- Non-linear alternatives which find a low dimensional non-linear manifold passing close to the data

In this paper, we concentrate on the first two categories and present a linear probabilistic dimensionality reduction technique called semi-parametric PCA (SP-PCA in short) which improves over some recently proposed alternatives to PCA. A common thread among many of these alternatives to PCA is that they are based on latent variable modelling which is commonly used in statistics to summarize observations [BK99]. A latent variable model assumes that the distribution of data is determined by a latent or mixing distribution $P(\boldsymbol{\theta})$ and a conditional or component distribution $P(\mathbf{x}|\boldsymbol{\theta})$, i.e., $P(\mathbf{x}) = \int P(\boldsymbol{\theta})P(\mathbf{x}|\boldsymbol{\theta})d\boldsymbol{\theta}$. In the discussion that follows, we present PCA and its extensions in the latent variable framework and motivate the need for SP-PCA.

PCA finds a lower dimensional space that minimizes $\sum_i ||\mathbf{x}_i - \boldsymbol{\theta}_i||^2$, the sum of squared distances from data $\mathbf{x}_i$ to their projections $\boldsymbol{\theta}_i$. In a *quasi*-probabilistic interpretation of PCA, each point $\mathbf{x}_i$ is thought of as a random draw from some unknown distribution $P(\mathbf{x}|\boldsymbol{\theta})$, where $P(\mathbf{x}|\boldsymbol{\theta})$ denotes a unit Gaussian with mean $\boldsymbol{\theta} \in \mathbb{R}^d$ [CDS02]. Then, PCA can be thought of as finding a set of parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ that maximize the likelihood of the data subject to the constraint that the parameters lie in a lower dimensional subspace.

Note that this interpretation does not mean that PCA is associated with a probability model, since the parameters $\boldsymbol{\theta}_i$ are assumed to be drawn arbitrarily from the subspace and not according to any distribution. A probabilistic formulation of PCA can offer several advantages like allowing statistical testing, application of Bayesian inference methods and naturally accommodating missing values [TB99].

Probabilistic PCA (PPCA) is an alternative to PCA based on the factor analysis model [TB99, Row98]. In this model, data is assumed to be generated by the following two step process. First a latent variable, $\boldsymbol{\theta}$, is drawn according to the prior or latent distribution $P(\boldsymbol{\theta})$, which is constrained to a lower dimensional Euclidean subspace of the data space. This models the belief that data is intrinsically low-dimensional. Then the observed data $\mathbf{x}$ is generated by adding a spherical gaussian noise to $\boldsymbol{\theta}$, i.e., the conditional distribution $P(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be Gaussian.

A key feature of PPCA is that the latent distribution $P(\boldsymbol{\theta})$ is also assumed to be a single Gaussian since it leads to simple and fast model estimation. This means that the density of $\mathbf{x}$ is approximated by a Gaussian distribution whose covariance matrix is aligned along a lower dimensional subspace. This may be a good approximation when data is drawn from a single population and the goal is to explain the data in terms of a few variables. However, in machine learning we often deal with data drawn from several populations and PCA is used to reduce dimensions to control computational complexity of learning. A mixture model with Gaussian latent distribution would not be able to capture this information. Also, dimension reduction is often used to visualize data, where the goal is to understand the data structure and detect the presence of sub-populations belonging to the same class. The projection obtained using a Gaussian latent distribution tends to be skewed toward the center [TB99] and hence the distinction between nearby sub-populations may be lost in the visualization space. For these reasons, it is important not to make restrictive assumptions about the latent distribution.

We present an alternative probabilistic formulation where no assumptions are made about the distribution of the latent random variable $\boldsymbol{\theta}$, i.e., we estimate the latent distribution non-parametrically. Non-parametric latent distribution estimation allows us to approximate data density better than other existing methods and hence gives better low dimensional representations. In particular, multi-modality of the high dimensional density is better preserved in the projected space. We argue that even when the 'true' density of $\mathbf{x}$ is far from the model space, this method captures as much of the structure of observed data density as possible in the low dimensional space. When the observed data is composed of several clusters, this technique can be viewed as performing simultaneous clustering and dimensionality reduction.

To make our method suitable for special data types, we allow the conditional distribution $P(\mathbf{x}|\boldsymbol{\theta})$ to be any member of the exponential family of distributions. Use of exponential family distributions for $P(\mathbf{x}|\boldsymbol{\theta})$ is common in statistics where it is known as latent trait analysis. It has also been used in several dimensionality reduction schemes [Tip99, BSW98, KG01, CDS02].

One possible way to estimate the $P(\boldsymbol{\theta})$ non-parametrically would be to assume that it belongs to a class of parametric distributions which is dense in the space of all distributions over the latent space. For example, we can approximate $P(\boldsymbol{\theta})$ by a mixture of Gaussians. However, using even a single Gaussian as prior requires use of monte-carlo techniques for model estimation when $P(\mathbf{x}|\boldsymbol{\theta})$ is not Gaussian, for example Bernoulli. Instead, we use Lindsay's non-parametric maximum likelihood estimation theorem to reduce the estimation problem to one with a large enough discrete prior. It turns out that this choice gives us a prior which is 'conjugate' to all exponential family distributions, allowing us to give a simple unified algorithm for all data types. This choice also makes it possible to efficiently estimate the model even in the case when different components of the data vector are of different types.

We present examples demonstrating the properties of semiparametric latent variable models with Gaussian conditional density and compare it to PCA and PPCA [TB99]. We also present simulation results on binary and count data which show that estimating the prior from data (instead of assuming a parametric form) can improve the quality of low dimensional projections both in terms of separating different populations and generalization to unseen samples. These properties, along with the fact that our algorithm remains computationally efficient for moderate values of projected space dimension, indicate that the method is suitable for general purpose projection in the pre-processing stage.

This report is organized in the following manner. Section 2.1 describes the probability model. In Sections 2.2 and 2.4 we describe how to use the model to get low dimensional representations and use it for data analysis and visualization. In Section 4 we discuss the consistency of maximum likelihood (ML) estimator for our model and derive an EM algorithm for ML estimation. Section 5 discusses how SP-PCA relates to past work and Section 6 presents simulation results comparing our work with some previously proposed methods. Section 7 contains conclusions and discussion of possible future work.

2 Semi-parametric PCA

2.1 The constrained mixture model

We assume that the d-dimensional observation vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are outcomes of iid draws of a random variable whose distribution $P(\mathbf{x}) = \int P(\boldsymbol{\theta})P(\mathbf{x}|\boldsymbol{\theta})d\boldsymbol{\theta}$ is determined by the latent distribution $P(\boldsymbol{\theta})$ and the conditional distribution $P(\mathbf{x}|\boldsymbol{\theta})$. This can also be viewed as a mixture density with $P(\boldsymbol{\theta})$ being the *mixing distribution*, the mixture components labelled by $\boldsymbol{\theta}$ and $P(\mathbf{x}|\boldsymbol{\theta})$ being the *component distribution* corresponding to $\boldsymbol{\theta}$.

The latent distribution is used to model the interdependencies among the components of $\mathbf{x}$ and the conditional distribution to model 'noise'. For example in the case of a collection of documents we can think of the 'content' of the document as a latent variable since it cannot be measured. For any given content, the words used in the document and their frequency may depend on random factors - for example what the author has been reading recently, and this can be modelled by $P(\mathbf{x}|\boldsymbol{\theta})$.

Conditional distribution $P(\mathbf{x}|\boldsymbol{\theta})$: We assume that $P(\boldsymbol{\theta})$ adequately models the dependencies among the components of $\mathbf{x}$ and hence that the mixture components are independent when conditioned upon $\boldsymbol{\theta}$, i.e., $P(\mathbf{x}|\boldsymbol{\theta}) = \Pi_j P(x_j|\theta_j)$, where x_j and θ_j are the j'th components of

$\mathbf{x}$ and $\boldsymbol{\theta}$. As noted in the introduction, using Gaussian means and constraining them to a lower dimensional subspace of the data space is equivalent to using Euclidean distance as a measure of similarity. This Gaussian model may not be appropriate for other data types, for instance 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]. Any member of this family can be written in the form

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

where 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 [CDS02]. Note that by using different distributions for the various components of $\mathbf{x}$, we can model mixed data types.

Latent distribution $P(\boldsymbol{\theta})$: Like previous latent variable methods, including PCA, we constrain the latent variable $\boldsymbol{\theta}$ to an ℓ-dimensional Euclidean subspace of $\mathbb{R}^d$ to model the belief that the intrinsic dimensionality of the data is smaller than d. One way to represent the (unknown) linear constraint on values that $\boldsymbol{\theta}$ can take is to write it as an invertible linear transformation of another random variable which takes values $\mathbf{a} \in \mathbb{R}^\ell$,

$$\boldsymbol{\theta} = \mathbf{a}V + \mathbf{b} \tag{1}$$

where V is an $\ell \times d$ rotation matrix and $\mathbf{b}$ is a d-dimensional displacement vector. Hence any distribution $P_\Theta(\boldsymbol{\theta})$ satisfying the low dimensional constraints can be represented using a triple $(P(\mathbf{a}), V, \mathbf{b})$, where $P(\mathbf{a})$ is a distribution over $\mathbb{R}^\ell$.

Lindsay's mixture non-parametric maximum likelihood estimate (NPMLE) theorem states that for fixed (V,$\mathbf{b}$), the maximum likelihood (ML) estimate of $P(\mathbf{a})$ exists and is a *discrete* distribution with no more than n distinct points of support [Lin83]. Hence if ML is the chosen parameter estimation technique, our model can be assumed (without loss of generality) to be a constrained finite mixture model with at most n mixture components. The number of mixture components in the model, n, grows with the amount of data and so does the computation required for model estimation (see Section 4.1.2). One way to address this problem is to start with $c = n$ and eliminating or merging components as the estimation procedure proceeds (Section 4.1.1). Also, it is often sufficient to start with $c \ll n$ since the distribution of the mixture component parameters along the hyperplane is used to capture the multi-modality and spread (or variance) of the density of X. The number of centers needed to get a good approximation will depend on how complex the distribution of X is or alternatively how much of the complexity we want to capture. For example, if all we want to do is separate out different populations, we do not need to approximate the variance of each population along the hyperplane very well. Use of small c also leads to better generalization. Finally, we note that instead of the natural parameter, any of its invertible transformations could have been constrained to a lower dimensional space. Choosing to linearly constrain the natural parameter affords us computational advantages similar to those available when we use the canonical link in generalized linear regression.

2.2 The low dimensional representation

There are several ways in which low-dimensional representations can be obtained using the constrained mixture model.

If the distribution of $\mathbf{x}$ is a constrained mixture density described above, we would ideally like to represent a given observation $\mathbf{x}$ by the unknown $\boldsymbol{\theta}$ (or the corresponding $\mathbf{a}$ related to $\boldsymbol{\theta}$ by Equation (1)) that generated it, since the conditional distribution $P(\mathbf{x}|\boldsymbol{\theta})$ is used to model random effects. However, the actual value of $\mathbf{a}$ is not known to us and all of our knowledge of $\mathbf{a}$ is contained in the posterior distribution

$$P(\mathbf{a}|\mathbf{x}) = \frac{P(\mathbf{a})P(\mathbf{x}|\mathbf{a})}{P(\mathbf{x})}$$

Since $P(\mathbf{x}|\mathbf{a}) = \prod_{j=1}^{d} P_0(x_j) \exp(x_j\theta_j - G(\theta_j))$, where $\theta_j = b_j + a_1V_{1j} + \ldots + a_LV_{\ell j}$, we can write the posterior as

$$P(\mathbf{a}|\mathbf{x}) = \frac{\exp(\sum_{j=1}^{d} x_j\theta_j - \sum_{j=1}^{d} G(\theta_j))}{\int_{\mathbf{a}} P(\theta) \exp(\sum_{j=1}^{d} x_j\theta_j - \sum_{j=1}^{d} G(\theta_j))}$$

Since $\mathbf{a}$ belongs to an ℓ-dimensional space, any of its estimators like the *posterior mean* or mode (MAP estimate) can be used to represent $\mathbf{x}$ in ℓ dimensions. For presenting the simulation results in this paper, we use the posterior mean as the representation point. This representation has been used in other latent variable methods to get meaningful low dimensional views [TB99, Tip99, KG01].

Note that the distribution $P(\mathbf{a}|\mathbf{x})$ depends on $\mathbf{x}$ only through

$$\sum_{j=1}^{d} x_j\theta_j = \sum_{l=1}^{\ell} a_l \sum_{j=1}^{d} x_jV_{lj}$$

Hence $\mathbf{x}$ can also be represented [BK99] by the ℓ-dimensional minimal sufficient statistic

$$\{\sum_{j=1}^{d} x_jV_{1j}, \ldots, \sum_{j=1}^{d} x_jV_{\ell j}\}$$

Yet another 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 [CDS02]). For the Gaussian case, this representation is the usual Euclidean projection.

2.3 Reference vectors view

Dimension reduction using this model can be viewed as a 'reference vectors' based method. In this view, each $\boldsymbol{\theta}_i$ acts as a reference vector and using ML estimation to find the distribution $P(\boldsymbol{\theta})$, is a natural way to find the appropriate locations and relative weights (importance) for $\boldsymbol{\theta}_i$'s. In the estimation process, the reference vectors are moved around so that they cluster toward the 'centers' of data clusters and the subspace on which they lie is moved as close as possible to the data. The posterior mean representation is the weighted average of these reference vectors where the weights are determined by how 'far' $\mathbf{x}$ is from each of them. Hence, we expect SP-PCA to generate meaningful projections even when data is not generated according to a constrained mixture model.

2.4 Visualization and data analysis

SP-PCA can be used in several ways to visualize high dimensional data. Firstly, the projections using posterior mean can reveal presence of clusters. Secondly, a topographic (contour) map of the posterior induced by data point $\mathbf{x}$ will reveal which sections of the projected space (sub-populations) it is close to in the appropriate Bregman divergence sense. If natural parameter vectors of two exponential family distributions are close to each other, then so are the corresponding mean parameters, since g, the one-one invertible function map between these two parameter spaces is typically continuous. This means that if representations of two data points are close to one another in the projected space, then so are the data points in some directions. Also, plotting the estimated prior $\hat{P}(\boldsymbol{\theta})$ will indicate clusters or reveal multi-modality in the pdf of $\mathbf{X}$ and examining the parameter values corresponding to these modes will reveal distinguishing characteristics of the clusters. However the actual values of the mixture parameters may not pass close to data points if $P(\boldsymbol{\theta})$ is not concentrated along some hyperplane of dimension ℓ.

3 The Gaussian case

When the exponential family distribution chosen is Gaussian, the model is a mixture of n spherical Gaussians, with a fixed variance σ, all of whose means lie on a hyperplane in the data space. This can be thought of as a 'soft' version of PCA, i.e., Gaussian case of SP-PCA is related to PCA in the same manner as Gaussian mixture model is related to K-means. The use of arbitrary mixing distribution over the plane allows us to approximate arbitrary spread of data along the hyperplane (see Fig. 1). Use of fixed variance spherical Gaussians ensures that like PCA, the direction perpendicular to the plane (V, b) is irrelevant in any metric involving relative values of likelihoods $P(\mathbf{x}|\boldsymbol{\theta}_k)$, including the posterior mean. To see why this is the case, consider $\mathbf{x}_p$, the point on the hyperplane (V, b) closest to $\mathbf{x}$. Now, $P(\mathbf{x}|\boldsymbol{\theta}_k) \propto \exp(-\{||\mathbf{x}, \mathbf{x}_p||^2 + ||\mathbf{x}_p, \boldsymbol{\theta}_k||^2\}/2\sigma^2)$ and for a fixed $\mathbf{x}$, the factor involving $||\mathbf{x}, \mathbf{x}_p||^2$ is common to all $\boldsymbol{\theta}_k$'s on the hyperplane (V, b) and hence cancels out.

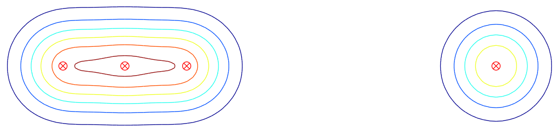


Figure 1: Subspace aligned variance approximated by clustered but slightly spread out mixture component mean parameters $\otimes$

When using SP-PCA as a low-dimensional density model, σ should be assumed to be unknown and estimated using ML along with other parameters of the model. When SP-PCA is being used only to project data into a lower dimensional space, we noticed that assuming a reasonable fixed variance (a few times the minimum distance between data points) worked well.

Consider the case when data density $P(\mathbf{x})$ belongs to our model space, i.e., it is specified by $\{A, V, b, \Pi, \sigma\}$ and let D be any direction parallel to the plane (V, b) along which the latent distribution $P(\boldsymbol{\theta})$ has non-zero variance. Since Gaussian noise with variance σ is added to this latent distribution to obtain $P(\mathbf{x})$, variance of $P(\mathbf{x})$ along D will be greater than σ. The variance of $P(\mathbf{x})$ along any direction perpendicular to (V, b) will be exactly σ. Hence, PCA of $P(\mathbf{x})$ yields the subspace (V, b) which is the same as that obtained using SP-PCA. This may not be true when $P(\mathbf{x})$ does not belong to our model space and we found that SP-PCA differs significantly from PCA and PPCA in the nature of low dimensional representations obtained using the estimated prior (see Section 6).

4 Model estimation

4.1 Algorithm for ML estimation

Given n iid samples of a d-dimensional random variable $\mathbf{X}$, we derive an EM algorithm for estimating parameters of a finite mixture model with the components constrained to an ℓ-dimensional Euclidean subspace. Let c be the number of components of the finite mixture and let the mixing density be $\Pi = (\pi_1, \ldots, \pi_c)$. Associated with each mixture component (indexed by k) is a parameter vector $\boldsymbol{\theta}_k$ and $\mathbf{a}_k$ which are related by $\boldsymbol{\theta}_k = \mathbf{a}_k V + b$. In this section we will work with the assumption that all features $\mathbf{X}_j$ correspond to the same exponential family for ease of notation.

Let A be an $c \times \ell$ matrix whose k'th row is $\mathbf{a}_k$, B be an $c \times d$ matrix all of whose rows equal $\mathbf{b}$ and Θ be an $c \times d$ matrix whose k'th row is $\boldsymbol{\theta}_k$. Hence we can rewrite Equation 1 as $\Theta = AV + B$. Our model is parametrized by $\{\Pi, A, V, B\}$. As in the case of usual (unconstrained) finite mixture model estimation, we introduce a 'missing' variable $\mathbf{Z}$ for use in EM derivation. For each observed $\mathbf{x}_i$ there is an unobserved $\mathbf{z}_i$, a c-dimensional binary vector whose k'th component is one if the k'th mixture component was the outcome in the i'th random draw and zero otherwise. Writing the complete data log likelihood function,

$$\begin{aligned}\log\ P(\mathbf{x}_1^n, \mathbf{z}_1^n / \Pi, A, V, B) &= \sum_{i=1}^{n} P(\mathbf{x}_i, \mathbf{z}_i / \Pi, A, V, B) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ik} \log \pi_k + \sum_{i=1}^{n} \sum_{k=1}^{c} \sum_{j=1}^{d} z_{ik} \log P(x_{ij} / \theta_{kj})\end{aligned}$$

The E-step is identical to unconstrained finite mixture case,

$$\hat{z}_{ik} = E(z_{ik}) = \frac{\pi_k P(\mathbf{x}_i / \boldsymbol{\theta}_k)}{\sum_{m=1}^{c} \pi_m P(\mathbf{x}_i / \boldsymbol{\theta}_m)}, \quad \text{for} \quad k = 1, \ldots, c \quad \text{and} \quad i = 1, \ldots, n$$

In the M-step we need to update all the parameters Π, A, V and B. The maximizing value of Π is independent of other parameters and is given by

$$\pi_k = \frac{\sum_{i=1}^{n} \hat{z}_{ik}}{\sum_{i=1}^{n} \sum_{m=1}^{c} x_{im}} = \frac{\sum_{i=1}^{n} \hat{z}_{ik}}{n}$$

A, V and $\mathbf{b}$ should be updated in such a way as to strictly increase the value of the function ℓ or equivalently of $\tilde{L}$ given by

$$\begin{aligned}L(A, V, \mathbf{b}) &= \sum_{i=1}^{n} \sum_{k=1}^{c} \sum_{j=1}^{d} \hat{z}_{ik} \{x_{ij} \theta_{kj} - G(\theta_{kj})\} \\ \tilde{L}(A, V, \mathbf{b}) &= \sum_{k=1}^{c} \sum_{j=1}^{d} \{\theta_{kj} \tilde{x}_{kj} - G(\theta_{kj})\}\end{aligned}$$

where,

$$\tilde{x}_{kj} = \frac{\sum_{i=1}^{n} \hat{z}_{ik} x_{ij}}{\sum_{i=1}^{n} \hat{z}_{ik}}$$

To optimize $\tilde{L}(A, V, \mathbf{b})$, we could use alternating minimization (similar to the algorithm in [CDS02]) since the function to be optimized is convex in each element of the matrices A, V and $\mathbf{b}$.

However, for the sake of speed, we propose an iterative weighted least squares method along the lines of Generalized linear models [MN83], i.e., we apply the Newton-Raphson (NR) procedure to find zeros of the derivative of $\tilde{L}(A, V, \mathbf{b})$. Use of NR does not guarantee monotone increase in the value of $\tilde{L}$. However, $\tilde{L}$ always increases locally in the direction in which NR moves the parameters and so we can move in small steps whenever NR stepping leads to a decrease in $\tilde{L}$. 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 smaller matrix operations. (Derivation of the NR update equations is shown in Appendix A). Similarly, we find that each column of V and each dimension of $\mathbf{b}$ can be updated independently.

Each row of A, $\mathbf{a}_i$, is updated by adding $\delta\mathbf{a}_i$ calculated using

$$(V\Omega_i V')\delta\mathbf{a}_i = GR_i$$

where the $d \times d$ diagonal matrix Ω_i, and the $\ell \times 1$ matrix GR_i are given by,

$$[\Omega_i]_{qq} = \frac{\partial g(\theta_{iq})}{\partial \theta_{iq}}$$

$$[GR_i]_{l1} = \sum_{j=1}^{d} (\tilde{x}_{ij} - g(\theta_{ij}))V_{lj}$$

Each column of the matrix V, $\mathbf{v}_s$, is updated by adding $\delta\mathbf{v}_s$ obtained by solving

$$(A'\Omega_s A)\delta\mathbf{v}_s = GR_s$$

where the $c \times c$ diagonal matrix Ω_s, and the $\ell \times 1$ matrix GR_s are given by,

$$[\Omega_s]_{kk} = \frac{\partial g(\theta_{ks})}{\partial \theta_{ks}}$$

$$[GR_s]_{l1} = \sum_{k'=1}^{c} (\tilde{x}_{k's} - g(\theta_{k's}))A_{k'l}$$

Each column of the row matrix $\mathbf{b}$, 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

$$H_s = \sum_{k'=1}^{c} \frac{\partial g(\theta_{k's})}{\partial \theta_{k's}}$$

$$GR_s = \sum_{k'=1}^{c} (\tilde{x}_{k's} - g(\theta_{k's}))$$

4.1.1 Pruning the mixture components

Redundant mixture components can be pruned between the EM iterations in order to improve speed of the algorithm and generalization properties while retaining the full capability to approximate $P(\mathbf{x})$. We propose the following criteria for pruning

- Starved components : If $\pi_k < C_1$, then drop the k'th component
- Nearby components : If $\max_i |P(\mathbf{x}_i|\boldsymbol{\theta}_{k1}) - P(\mathbf{x}_|\boldsymbol{\theta}_{k2})| < C_2$, then drop either $k1$'th or $k2$'th component

The value of C_1 should be $\Theta(1/n)$ since we want to measure how starved a component is based on what percentage of the data it is 'responsible' for. To measure the nearness of components we use the distance of between probabilities the components assign to observations. If we were working with mixture of Gaussians, we could have used the usual distance between mixture component parameters. However, for general exponential family distributions, the Euclidean distance between two components does not accurately reflect the difference in the distributions that they represent. For example, for Bernoulli distributions with natural parameter θ, $\theta = 1000$ is practically identical to $\theta = 10000$ whereas $\theta = 0$ is significantly different from $\theta = 1$. The ∞-norm of the difference between probability vectors is used instead of its two-norm since we do not want to lose mixture components that are distinguished with respect to a small number of observation vectors. In the case of clustering this means that we do not ignore under-represented clusters. C_2 should be chosen to be a small constant, depending on how much pruning is desired.

4.1.2 Convergence of the EM iterations and computational complexity

It is easy to verify that our model satisfies the continuity assumptions of Theorem 2, [Wu83], and hence we can conclude that any limit point of the EM iterations is a stationary point of the log likelihood function.

Time taken for the E-step is $\mathcal{O}(cdn)$ since for each data vector $\mathbf{x}$ and component $\boldsymbol{\theta}$ we need to compute $P(\mathbf{x}|\boldsymbol{\theta})$ which is a product of d univariate densities. In the M-step, each update of the parameter vector $(A, V, \mathbf{b})$ involves computing the hessian matrices and then inverting them. Using naive matrix multiplication and inversion, the time taken is $\mathcal{O}(cd\ell^2)$. Hence the computational complexity of each iteration of the EM algorithm is $\mathcal{O}(cd\ell^2 + cdn)$.

For the Gaussian case, the E-step only takes $\mathcal{O}(c\ell n)$ since we only need to take into account the variation of data along the subspace given by current value of V (as explained in Section 3). The most expensive step is computation of $P(\mathbf{x}_i|\boldsymbol{\theta}_j)$, and this is a common problem faced in neural network training. [Omo87] gives a procedure for speeding up this computation using the k-d tree data structure by identifying relevant prototypes (for each $\mathbf{x}$) thereby avoiding unnecessary computation.

4.1.3 Model selection : Choosing the dimension of projected space

In the derivation of the estimation algorithm, we assumed that ℓ, the dimension of representation space, is a known and fixed input. While any of the standard model selection methods based on penalizing complexity could be used to choose ℓ, an alternative reasonable method is to pick ℓ which minimizes a validation or bootstrap based estimate of the prediction error (negative log likelihood per sample). For the Gaussian case, a fast method to pick ℓ would be to plot the variance of data along the principal directions (found using PCA) and look for the dimension at which there is a 'knee' or a sudden drop in variance or where the total residual variance falls below a chosen threshold.

4.2 Consistency of the Maximum Likelihood estimator

We propose to use the ML estimator to find the latent space (V, b) and the latent distribution $P(\mathbf{a})$. Usually a parametric form is assumed for $P(\mathbf{a})$ and the consistency of the ML estimate is well known for this task where the parameter space is a subset of a finite dimensional Euclidean space. In our model, one of the parameters $P(\mathbf{a})$ ranges over the space of all distribution functions on $\mathbb{R}^\ell$ and hence we need to do more to verify the validity of our estimator.

Before defining consistency, one issue we need to address is the non-identifiability of some mixture distributions. Consider a parametric family of cumulative distribution functions, $\mathcal{F} = \{F(x/\gamma), \;\; \gamma \in \Gamma\}$ (parameter γ takes values in the parameter space Γ). The elements of Γ are said to be *identifiable* if $\forall\ \gamma \neq \gamma', \;\; \exists\ x \;\; \text{s/t} \;\; F(x/\gamma) \neq F(x/\gamma')$. Exponential family mixture distributions are not identifiable in general (for an example see [CPR00]). This, however, is not a problem for us since we are only interested in approximating $P(\mathbf{x})$ well and not in the actual parameters corresponding to the distribution. Hence we use the definition of consistency of an estimator given by Redner [Red81].

Let γ_0 be the 'true' parameter from which observed samples are drawn. Let C_0 be the set of all parameters γ corresponding to the 'true' distribution $F(x/\gamma_0)$ (i.e., $C_0 = \{\gamma : F(x/\gamma) = F(x/\gamma_0)\ \forall\ x\}$). Let $\hat{\gamma}_n$ be an estimator of γ based on n observed samples of X and let $\hat{\Gamma}$ be the quotient topological space obtained from Γ obtained by identifying the set C_0 to a point $\hat{\gamma}_0$.

Definition. *The sequence of estimators* $\{\hat{\gamma}_n, \; n = 1, \ldots, \infty\}$ *is said to be strongly consistent in the sense of Redner if* $\lim_{m\to\infty} \hat{\gamma}_n = \hat{\gamma}_0$ *almost surely.*

The following result follows by verifying that the assumptions of Kiefer and Wolfowitz [KW56] are satisfied by our model.

Theorem. *If* $P(\mathbf{a})$ *is assumed to be zero outside a bounded subset of* $\mathbb{R}^\ell$*, the ML estimator of parameter* $(V, b, P(\mathbf{a}))$ *is strongly consistent for Gaussian, Binary and Poisson conditional distributions.*

The proof details are given in Appendix B. The assumption that $P(\mathbf{a})$ is zero outside a bounded region is not restrictive in practice for Gaussian and Poisson distributions, since we expect the observations belong to a bounded region of $\mathbb{R}^d$. For the Bernoulli distribution, as we let $\theta \to +\infty$, the corresponding mean parameter $\mu \to 1$ slower and slower (similarly with $\theta \to -\infty$). Hence if we take the subset to be large enough, there is no restriction within computing precision.

5 Relation to past work

SP-PCA is a factor model that makes fewer assumptions about latent distribution than PPCA. Mixtures of probabilistic principal component analyzers (also known as mixtures of factor analyzers) is a generalization of PPCA which aims to overcome the limitation of global linearity of PCA via local dimensionality reduction. Mixtures of SP-PCA's can be similarly defined and used for local dimensionality reduction.

Tipping [Tip99] proposes a binary data visualization technique based on a latent trait model. This model is similar to PPCA in that it assumes that the latent distribution is Gaussian.

Collins et. al. [CDS02] proposed a generalization of PCA using exponential family distributions. Like PCA, this generalization is not associated with a probability density model for the data. SP-PCA with conditional distributions drawn from the exponential family can be thought of as a 'soft' version of this generalization of PCA.

Non-negative matrix factorization [LS00] is another non-probabilistic generalization of PCA for special data types in which the mean parameters of exponential family distributions are constrained to a lower dimensional subspace and no distribution is assumed over the latent space.

Probabilistic latent semantic indexing [Hof99] is a dimension reduction method based on a *latent class* model. In contrast with most methods we have discussed, in PLSI the latent distribution is not constrained to a lower dimensional subspace, but is instead constrained to be discrete over ℓ points, when the objective is to reduce data dimension to ℓ.

Generative topographic mapping (GTM) is a probabilistic alternative to Self organizing map (SOM) which aims at finding a nonlinear lower dimensional manifold passing close to data points. An extension of GTM using exponential family distributions to deal with binary and count data is described in [BSW98, KG01]. Apart from the fact that GTM is a non-linear dimensionality reduction technique while SP-PCA is globally linear like PCA, one main feature that distinguishes the two is the choice of latent distribution. GTM assumes that the latent distribution is uniform over a finite and discrete grid of points. Both the location of the grid and the nonlinear mapping are to be given as an input to the algorithm.

Tibshirani [Tib92] defined a semi-parametric latent variable model for estimation of principle curves. Unlike SP-PCA, the mixing density was not constrained to lie in a subspace, only Gaussian mixture components were considered and each Gaussian component was allowed to have arbitrary covariance matrix. This method makes no assumptions/restrictions on the the relative positions of the mean parameters of the Gaussian components and hence there is no topographic ordering on the mixture component mean parameters obtained at the end of model estimation. Hence, this method cannot be used to reduce dimensions when data is in more than three dimensions and a reasonable ordering of component means cannot be visually determined.

6 Experiments

In this section we present simulations on synthetic and real data to demonstrate the properties of SP-PCA. In factor analysis literature, it is commonly believed that choice of prior distribution is unimportant for the low dimensional data summarization (see [BK99], Sections 2.3, 2.10 and 2.16). Through the examples below we argue that estimating the prior instead of assuming it arbitrarily can make a difference both when we use the low dimensional representation for visualization/data analysis and also when we plan to use distances between points in the low dimensional space to measure similarity of data.

We present three sets of experiments. In the first set, we use synthetic two dimensional data to illustrate the properties of the Gaussian case of SP-PCA and compare it with PCA and PPCA. The second set of experiments over the Tobamovirus data set [Rip96] include a comparison of the predictive power of PPCA and SP-PCA. The third set consist of simulation results on binary and count data using the Bernoulli and Poisson cases of SP-PCA and comparison with exponential family PCA [CDS02] and with the exponential family version of GTM [KG01].

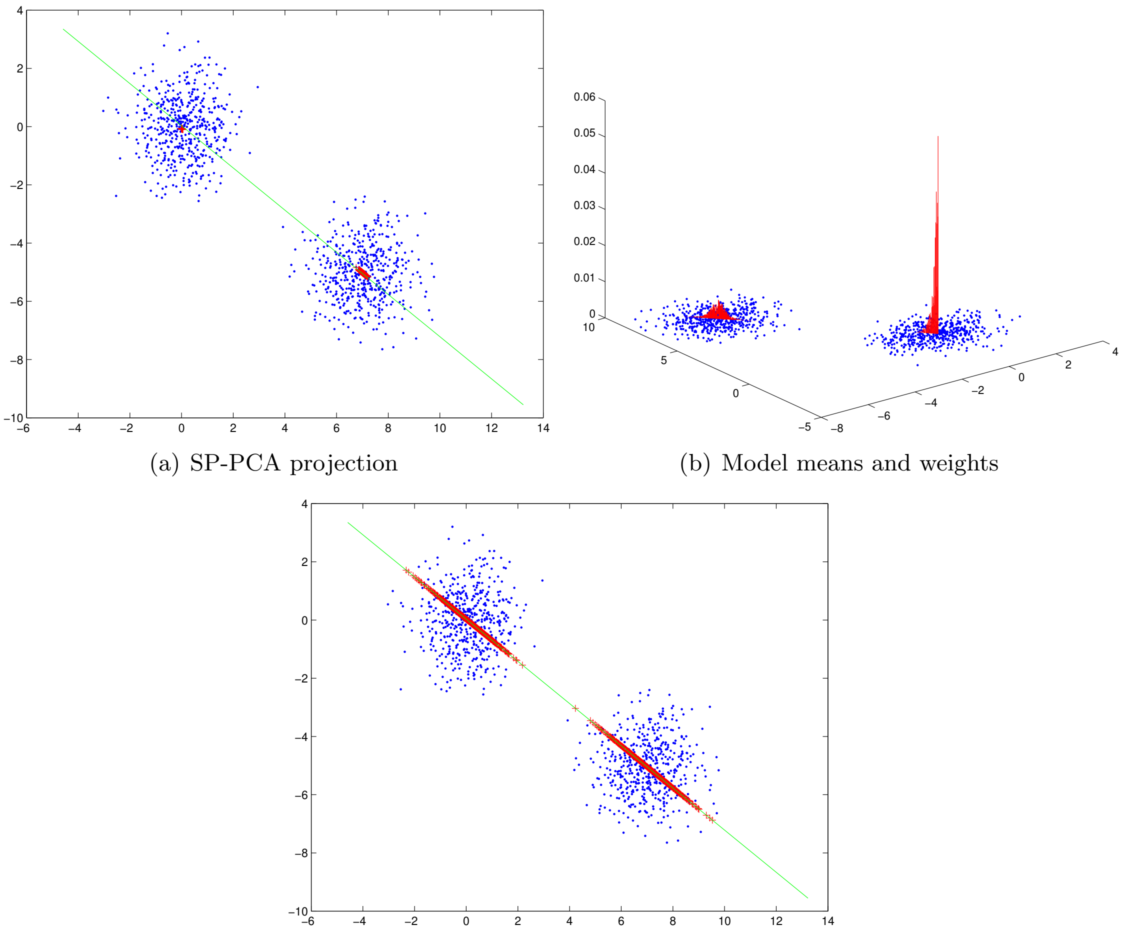
6.1 Illustrative examples of synthetic two dimensional data

In all the figures in this section, .'s represent data points and +'s represented projected points. To show the estimated constrained finite mixture model, we use straight lines located at the component parameters $\boldsymbol{\theta}_i$ or $\mathbf{a}_i$, and the height of the lines corresponds to the component weights π_i. For each of the examples in this subsection, the low dimensional projections obtained using PPCA [TB99] were indistinguishable from PCA and hence we present only PCA and SP-PCA results.

6.1.1 Simultaneous clustering and dimension reduction

Mixture modelling is commonly used for clustering, for example Gaussian mixture estimation is often used as a 'soft' alternative to K-means. Even when the mixture means are constrained to lie in a lower dimensional subspace, they tend to collect into groups corresponding to the clusters (when present). For this reason, our model performs simultaneous clustering and dimensionality reduction.

The first example is a mixture of two unit variance, symmetric Gaussians. Fig. 2 shows the projection of this two-dimensional data with two clusters onto a one-dimensional subspace. Note that the component means are clustered near the centers of the two clusters and that each cluster in the projection using PCA is more spread out than in the projection using SP-PCA.



(a) SP-PCA projection

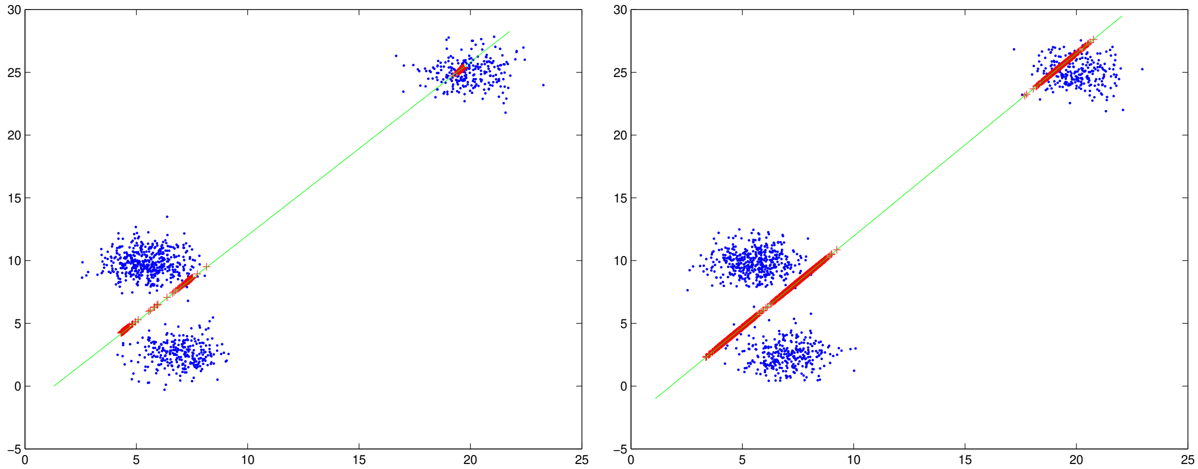
(b) Model means and weights

(c) PCA projection

Figure 2: Projection of samples drawn from a mixture of two spherical Gaussian densities.

In the second example, Fig. 3, we have a mixture of three Gaussians whose means do not lie on a one-dimensional plane. Also, two of the three clusters are close to each other. Even though projection using PCA preserves multi-modality in the lower dimensional space, the separation is better visualized in the projection using the SP-PCA.

It is possible that the SP-PCA model corresponding to parameters at a local maximum of the likelihood function clearly shows all clusters in the data while the global maximum does not. This is illustrated in Fig. 4 where a model with lower likelihood yields a much better view than the model with higher likelihood shown in Fig. 3(a). We observed this phenomenon in simulations on real data sets as well.



(a) Projection using SP-PCA reveals 3 clusters (b) Projection using PCA reveals only 2 clusters

Figure 3: Visual identification of clusters easier with SP-PCA projection than with PCA

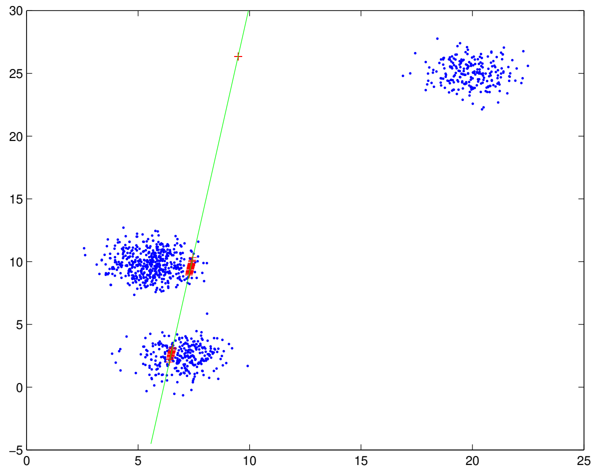
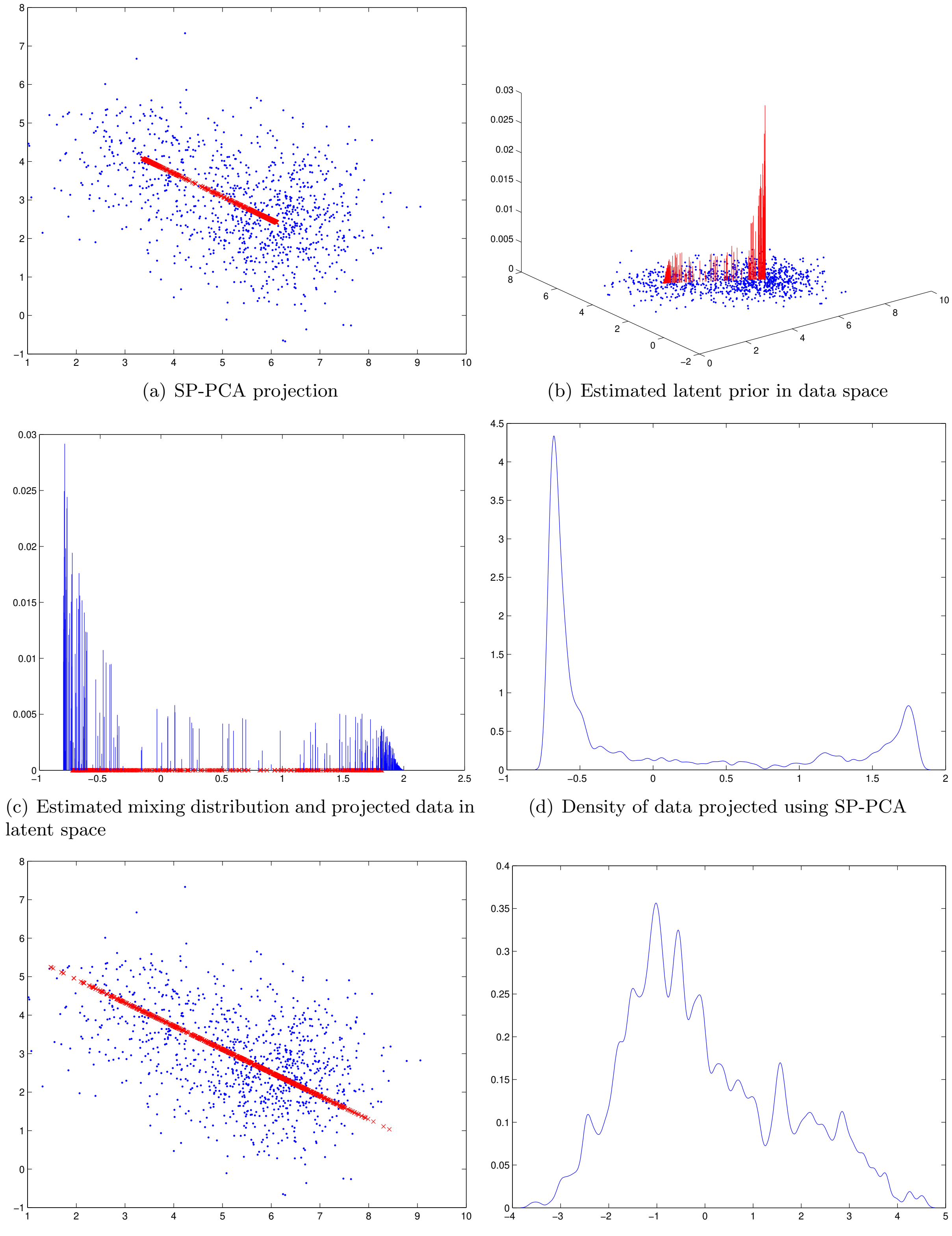


Figure 4: Projection using a local maximum of the likelihood surface may provide better view than the global maxima

6.1.2 Density aware nature of projection using SP-PCA

Now we consider more complex examples when the presence of multi-modality in the density of data is not obvious to human eye. In the first example (Fig. 5), we consider a mixture of two unit variance clusters which are close to one another. The data and its projection using SP-PCA is shown in Fig. 5(a). Fig. 5(b) shows the estimated parameters $\boldsymbol{\theta}_k$, where the height of the red lines is π_k, the weight of each component, and the location of the lines in the x-y plane is the mean parameter of the component gaussian. Fig. 5(b) shows the estimated mixture component parameters $\mathbf{a}_k$ and the projection of data in the latent space.

The density of projected data using SP-PCA (Fig. 5(d)) makes the presence of the two clusters clear and we see that the two clusters are separated by a low-density region. Hence SP-PCA projection summarizes the density of data and removing noise simultaneously with reducing dimensions.



(a) SP-PCA projection

(b) Estimated latent prior in data space

(c) Estimated mixing distribution and projected data in latent space

(d) Density of data projected using SP-PCA

(e) Projection using PCA

(f) Density of data projected using PCA

Figure 5: Two nearby unit-variance clusters

Table 1: Bootstrap estimates of prediction error for PPCA and SP-PCA.

Density	Isotropic gaussian	Diagonal gaussian	PPCA			SP-PCA			Full gaussian
			ℓ=1	ℓ=2	ℓ=3	ℓ=1	ℓ=2	ℓ=3	
Error	50.39	34.37	38.03	34.71	34.76	36.85	30.99	28.54	343.83

The second example (Fig. 6) uses data drawn from a mixture of two clusters which are close and have unequal variance (one cluster is unit Gaussian while the other has variance two along both axes). We observe that the latent distribution (Fig. 6(b)) is no longer clustered at the centers of the two clusters. The density of projected data using SP-PCA (Fig. 6(d)) indicates presence of two clusters, one much more spread out than the other, whereas the density of PCA projection shows just one widely spread mass of data.

6.2 Experiments with the Tobamovirus data

The Tobamovirus data [Rip96] consists of 38 examples and is 18-dimensional. It was used in [TB99] to illustrate properties of PPCA.

6.2.1 Missing data

A latent variable model estimated using the EM algorithm is naturally suited to deal with missing data. Missing data was simulated by randomly removing each value in the data set with probability 20%. Fig. 7(a) shows the direct projection of data on the subspace found using full data (this is similar to the PCA subspace projection which is not included for this reason). The projection obtained using missing data, Fig. 7(b), retains all the main features of PCA projection.

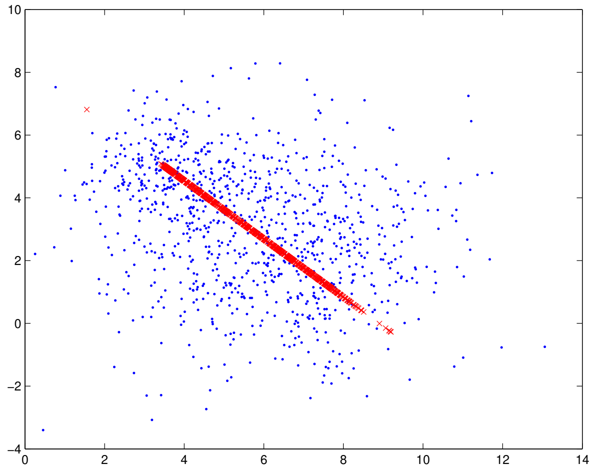
6.2.2 Predictive power

PPCA and SP-PCA can be thought of as density models of data. The densities defined by PPCA with lower dimension ℓ between 1 and the data dimension d can be thought of as models of intermediate complexity between an isotropic gaussian model and a Gaussian model with full covariance matrix. For data sets with small number of examples, the predictive error of a Gaussian model with full covariance matrix is likely to be high because of over fitting and that of an isotropic model may also be high since it is a very simple model. Hence Tipping and Bishop propose using PPCA as a means of controlling the degrees of freedom of the density model. SP-PCA also provides a range of density models with increasing complexity. For a fixed lower dimension ℓ, the density model of SP-PCA is more complex than that of PPCA.

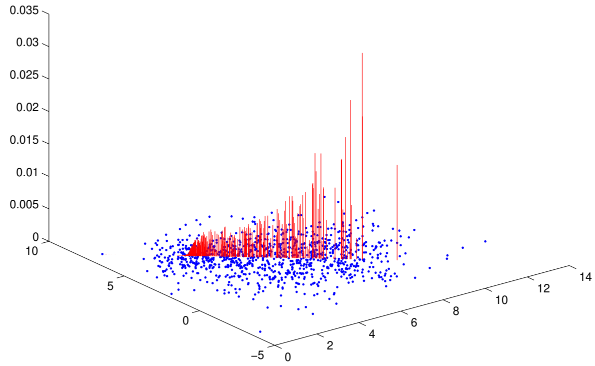
In Table 1, we present bootstrap estimates of the predictive power of PPCA and SP-PCA for various values of ℓ. SP-PCA has lower prediction error than PPCA for $\ell = 1,\ 2$ and 3. This indicates that SP-PCA has excellent generalization properties even when trained on a small amount of data.

6.3 Simulation results on discrete datasets

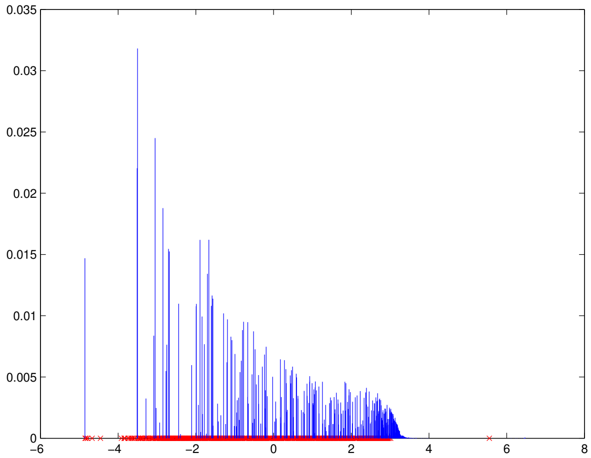
We performed simulations on a synthetic binary dataset and the 20 Newsgroups dataset (both binary and count data). Initialization of the matrices V and b was done randomly. In initialization of A it is important that the points $\mathbf{a}_i$ were spread out well on the plane. This can mean different things for different members of the exponential family, but we found that a uniform grid works well for Binary and Poisson distributions. While picking A we ensured that each component of $\mathbf{a}$ was bounded so that e^{θ} is neither too large nor too small. This is because the canonical link $g(\theta)$



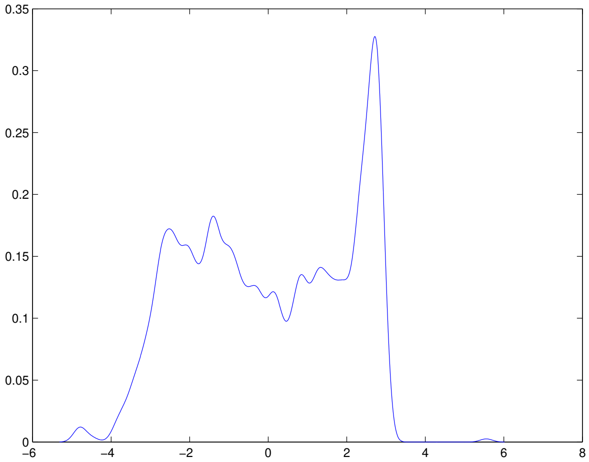
(a) SP-PCA projection



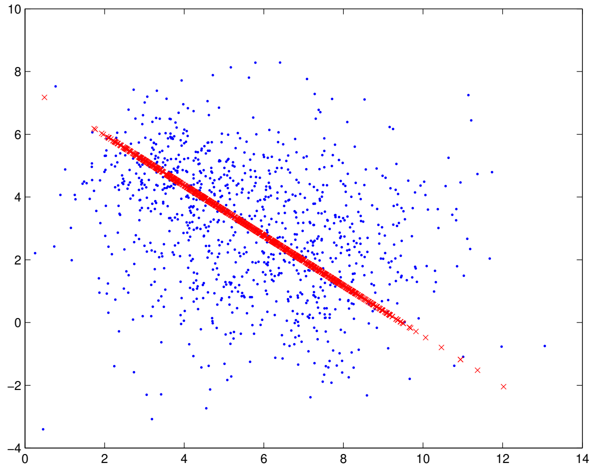
(b) Estimated latent prior in data space



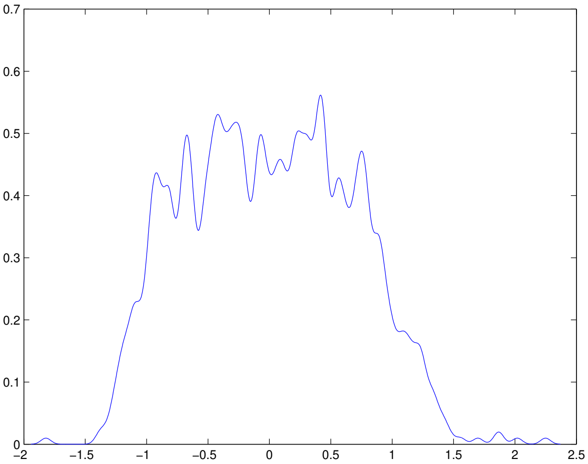
(c) Estimated mixing distribution and projected data in latent space



(d) Density of data projected using SP-PCA

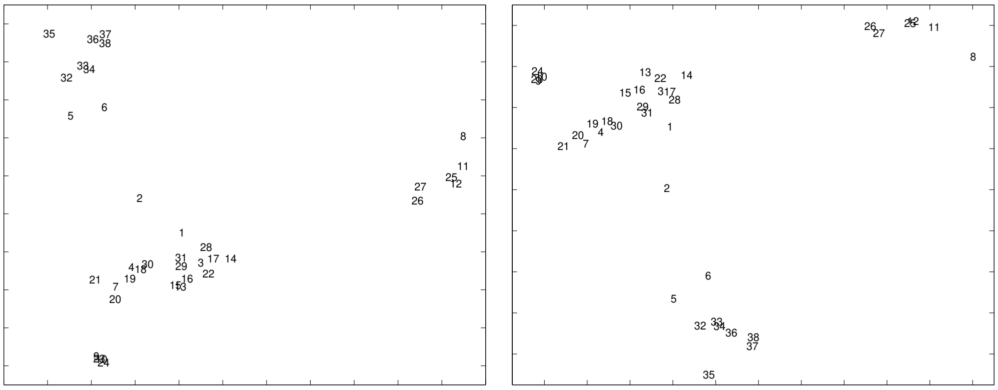


(e) Projection using PCA



(f) Density of data projected using PCA

Figure 6: Two nearby clusters - one cluster has twice the variance of the other along each axes



(a) Projection using full data (b) Projection with 130 missing values

Figure 7: Missing data

is a function of e^θ for the Binary and Poisson distributions. The iterations in M-step were not carried out until convergence since this is not necessary to ensure convergence to a local maxima (GEM algorithm). We did not use early stopping for improving model generalization for our simulations (though this will be useful when using the SP-PCA in applications).

We implemented Bernoulli and Poisson GTM and used simple gradient descent for optimization as suggested in [KG01]. We observed that random initialization of exponential GTM estimation often lead to poor projections of data while PCA-based initialization of the parameter matrix worked well.

We also implemented Exponential family PCA proposed by Collins et. al. [CDS02] and used iterative weighted least squares for optimizing the loss function defined in the paper. We observed that the problem of convergence of some representation points $\mathbf{y}_i$'s to a point at infinity often occurs and used the conjugate prior method proposed in the paper to prevent this from happening.

It is possible to encounter a similar problem in SP-PCA - convergence of $\mathbf{a}_i$'s to a point at infinity. However we cannot assume a conjugate prior on the mixture components in our scheme as it will destroy the non-parametric nature of prior assumption. We note that this is not a problem practically for the Poisson case since the counts we observed are finite and so are the means (irrespective of the responsibilities calculated in the E-step). A problem would arise if a component of V continually decreases and the corresponding component of A increases while keeping the mean bounded. However, we did not see this occur in our simulations. In the case of Bernoulli distribution, the mean and natural parameter are related by $\mu = \frac{e^\theta}{1+e^\theta}$, and if the optimal value of a parameter μ is close to zero or one, the corresponding θ and hence at least one of the corresponding entries in A or V would tend to infinity. In practice, this does not pose much of a problem if we use convergence of log likelihood as the stopping criterion because once θ becomes large/small enough, large changes in θ would lead to small changes in μ and correspondingly small changes in data likelihood and hence the algorithm would terminate automatically.

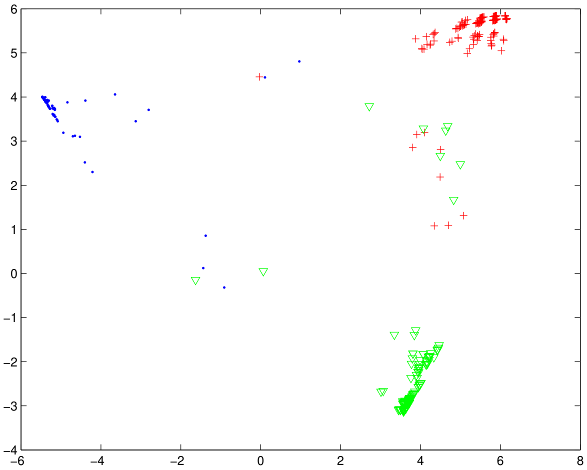
6.3.1 Projection of binary data

In order to compare various schemes for the binary case, we present projections (Fig. 8) of an artificial three cluster data used in [Tip99]. The 16-dimensional data vectors were generated by first randomly picking three 'prototype' vectors where each bit was drawn Bernoulli(1/2). Then

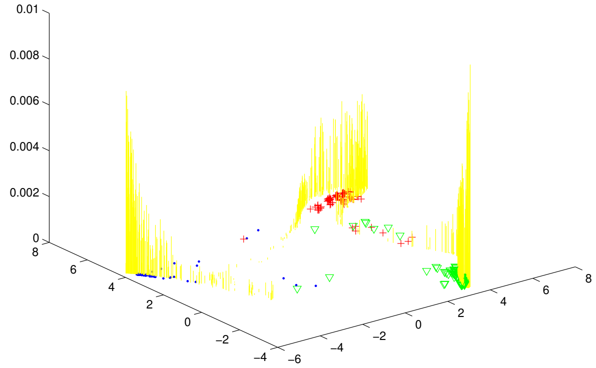
600 data points were generated by taking 200 copies of each prototype and inverting each bit with probability 0.15.

Note that both PCA (Fig. 8(d)) and Exponential family PCA [CDS02] (Fig. 8(c)) produced similar projections. These in turn are similar to the projections obtained using a latent trait analysis in which a gaussian prior is assumed (See [Tip99]). That the projection obtained using Exponential family PCA and latent trait analysis are similar is not surprising since we used a conjugate gaussian prior to project data in Exponential family PCA as suggested in [CDS02] to deal with the phenomenon of projections into latent space drifting to infinity.

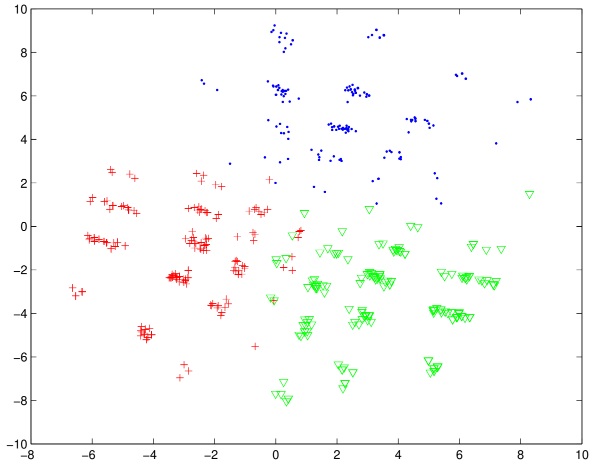
Projection using SP-PCA model shows three well separated clusters (Fig. 8(a)). In Fig. 8(b) we show the estimated prior distribution and note that latent parameters which were initialized uniformly on the latent space have migrated and formed three clusters.



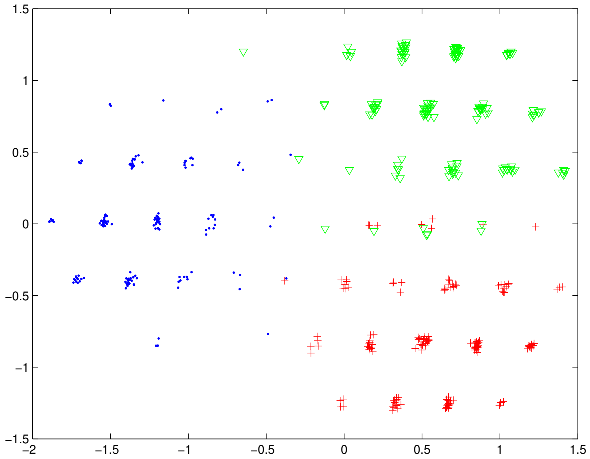
(a) Projection with SP-PCA



(b) Estimated latent prior and projected data in latent space



(c) Projection with Exponential family PCA



(d) Projection with PCA

Figure 8: Projection of binary data with three artificial clusters

6.3.2 Documents

We did simulations on two sets of samples drawn from the 20 Newsgroups data set. Data for the first set of simulations was drawn from comp.sys.ibm.pc.hardware, comp.sys.mac.hardware and sci.med newsgroups. A dictionary size of 150 words was chosen and the words in the dictionary were picked to be those which have maximum mutual information with class labels. 200 documents were drawn from each of the three newsgroups to form the training data. Two-dimensional representations obtained using various methods are shown in Fig. 9. In the projection obtained using Bernoulli GTM, the classes comp.sys.ibm.pc.hardware and comp.sys.mac.hardware were not well separated in the 2D space. This projection in Fig. 9(c) is similar to the results presented in [KG01] and the explanation given for the overlap between the two groups was that two newsgroups are very similar to one another and hence share many words in common. However we found that SP-PCA was able to separate the three sets reasonably well (Fig. 9(d)).

To quantify the separation of dissimilar groups in the two-dimensional projections we use the training set classification error of projected data using SVM. The accuracy of the best SVM classifier (we tried a range of SVM parameter values and picked the best for each projected data set) was 75% for bernoulli GTM projection and 82.3% for SP-PCA projection (the difference corresponds to 44 data points while the total number of data points is 600).

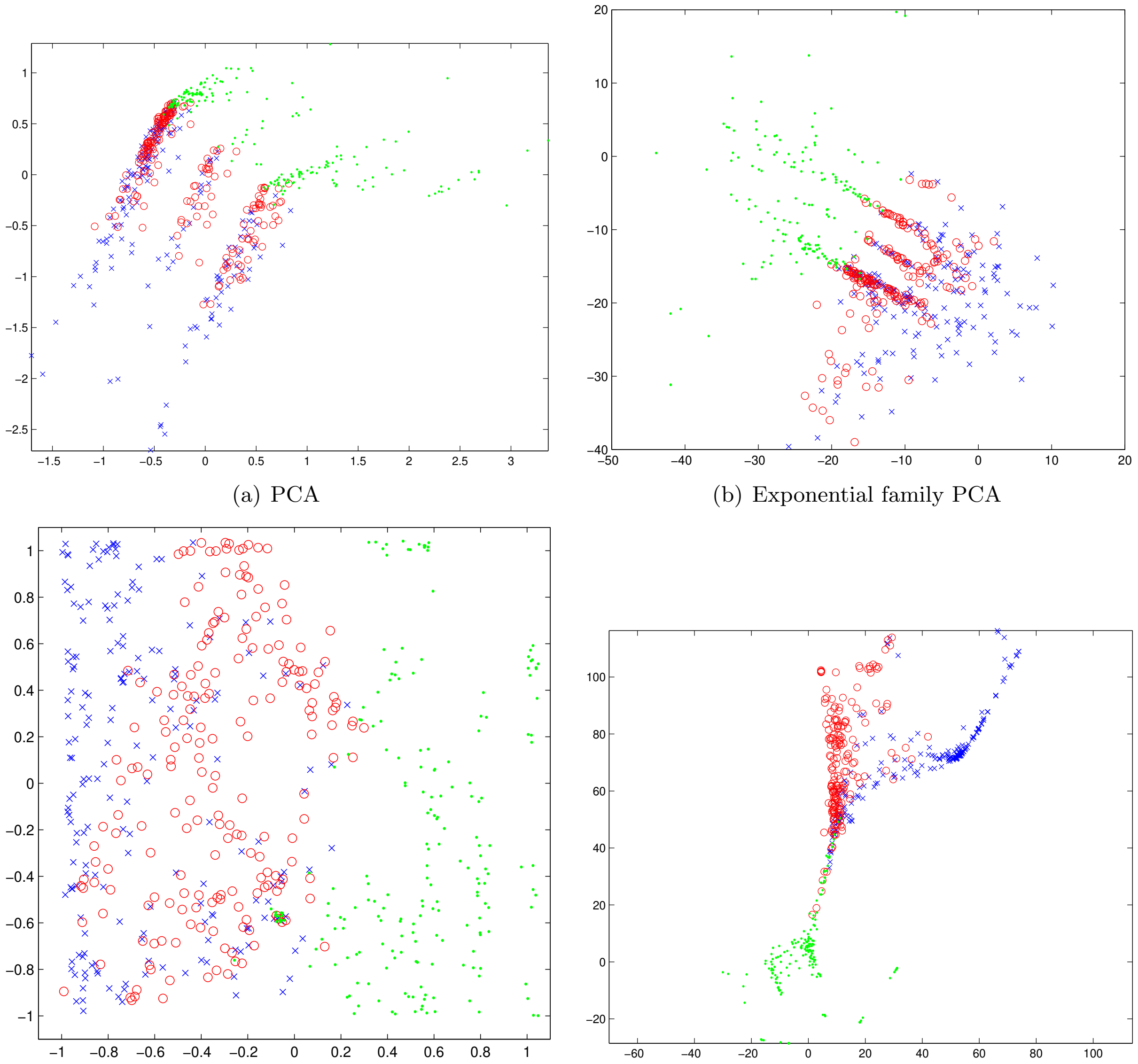
We conjecture that the reason Binary GTM did not succeed in separating comp.sys.ibm.pc.-hardware and comp.sys.mac.hardware is that the prior is over a pre-specified grid in latent space and the spacing between grid points happened to be large in the parameter space close to the two news groups. In contrast to this, in SP-PCA there is no grid and the latent 'reference vectors' $\boldsymbol{\theta}_i$ are allowed to move about freely to adapt to the data and hence are able to separate the two populations. Note that a standard clustering algorithm could be used on the data projected using SP-PCA to conclude that data consisted of three kinds of documents.

Data for the second set of simulations was drawn from sci.crypt, sci.med, sci.space and soc.-culture.religion.christianity newsgroups. A dictionary size of 100 words was chosen and again the words in the dictionary were picked to be those which have maximum mutual information with class labels. 100 documents were drawn from each of the newsgroups to form the training data.

Fig. 10 shows two-dimensional representations of binary data obtained using various methods. Newsgroups sci.space and sci.med are merged in projections by PCA. Note that while the four newsgroups are bunched together in the projection obtained using Exponential family PCA (Fig. 10(b)), we can still detect the presence four groups from this projection and in this sense this projection is better than the PCA projection. This result is pleasing since it confirms our intuition that using negative log-likelihood of Bernoulli distribution as a measure of similarity is more appropriate than squared Euclidean distance for binary data. We conjecture that the reason the four groups are not well separated in the Exponential family PCA projection is that a conjugate prior is used in its estimation for numerical reasons [CDS02] and the form and parameters of this prior are considered fixed and given inputs to the algorithm.

Both Binary GTM (Fig. 10(e)) and SP-PCA (Fig. 10(c)) were able to clearly separate the clusters in the training data. Figs. 10(f) and 10(d) show representation of test data using the models estimated for Binary GTM and SP-PCA respectively. To measure generalization of these methods, we use a K-nearest neighbors based non-parametric estimate of the projected training data density. The percentage difference between the log-likelihoods of training and test data with respect to this density was 9.1% for ExpPCA and 17.6% for GTM for K=40 (ExpPCA had smaller percentage change in log-likelihood for most values of K that we tried between 10 and 40). This indicates that SP-PCA generalizes better than GTM. This can be seen visually by comparing Figs. 10(e) and 10(f) where the projections of training and test data of sci.space (∇) differ significantly.

Fig. 11 shows two-dimensional representations of word count data (both training and test) obtained using Poisson GTM and SP-PCA. The percentage difference between the log-likelihoods of training and test data with respect to KNN estimate of projected training data density was 6.8% for SP-PCA and 12.1% for GTM for K=40 (Again, SP-PCA had smaller percentage change



(a) PCA (b) Exponential family PCA

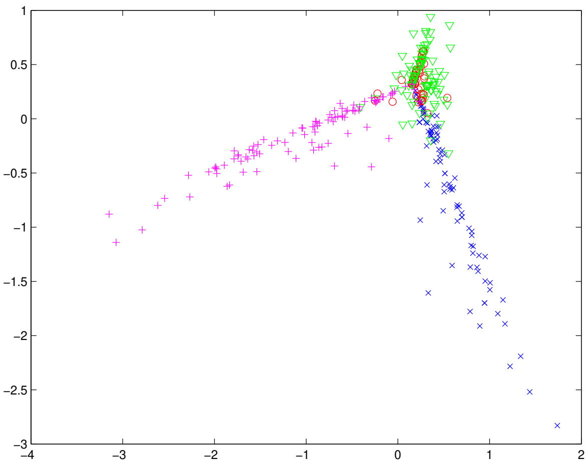
(c) GTM (d) SP-PCA

Figure 9: Projection by various methods of **binary** data from 200 documents each from comp.sys.ibm.pc.hardware ($\times$), comp.sys.mac.hardware ($\circ$) and sci.med (.)

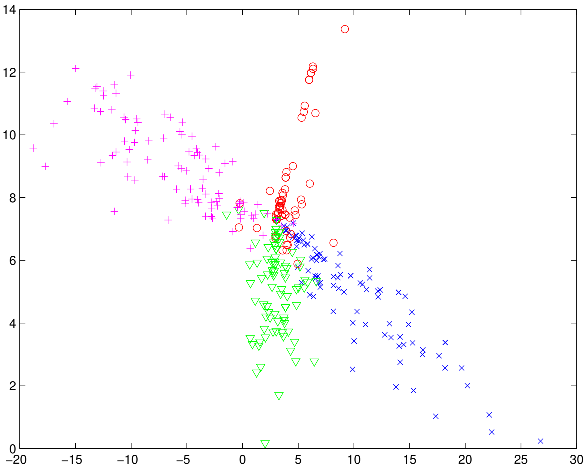
in log-likelihood for most values of K that we tried between 10 and 40). The better generalization property of SP-PCA is probably because it has greater flexibility in estimating the latent variable prior and hence it is better able to adapt to observed data.

7 Conclusions

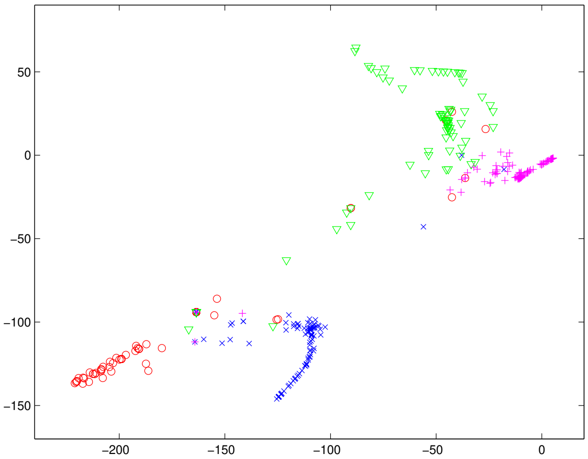
An intuitive and effective dimensionality reduction and visualization method for various data types is presented. A key feature of this model is that the mixing distribution is estimated non-parametrically and hence is better able to capture any multi-modality in data density. The method performs simultaneous clustering and dimensionality reduction since it is based on a finite mixture model. Simulations on standard datasets demonstrate that it is effective in separating



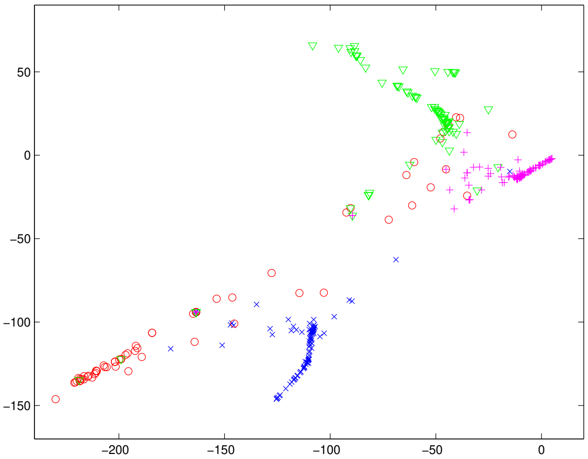
(a) PCA



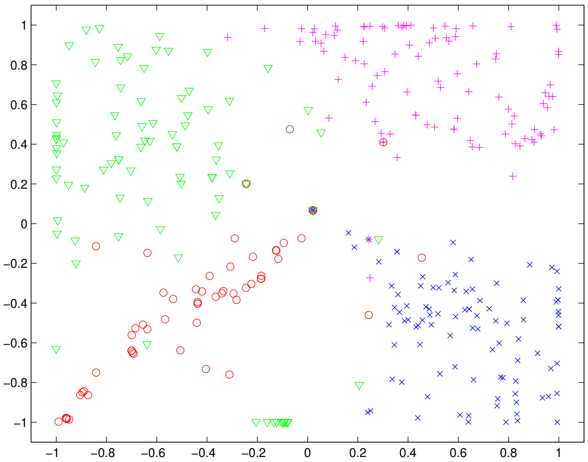
(b) Exponential family PCA



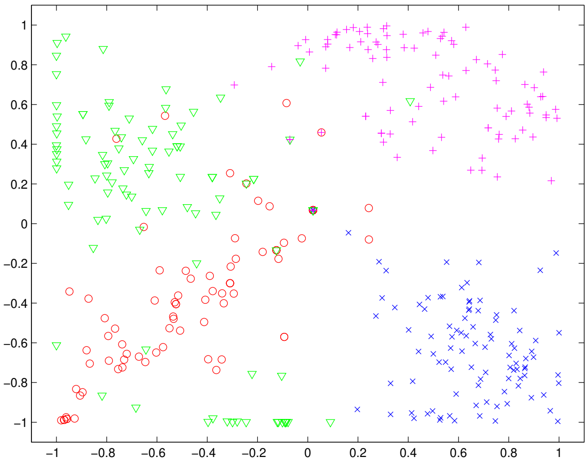
(c) SP-PCA



(d) Test data : SP-PCA

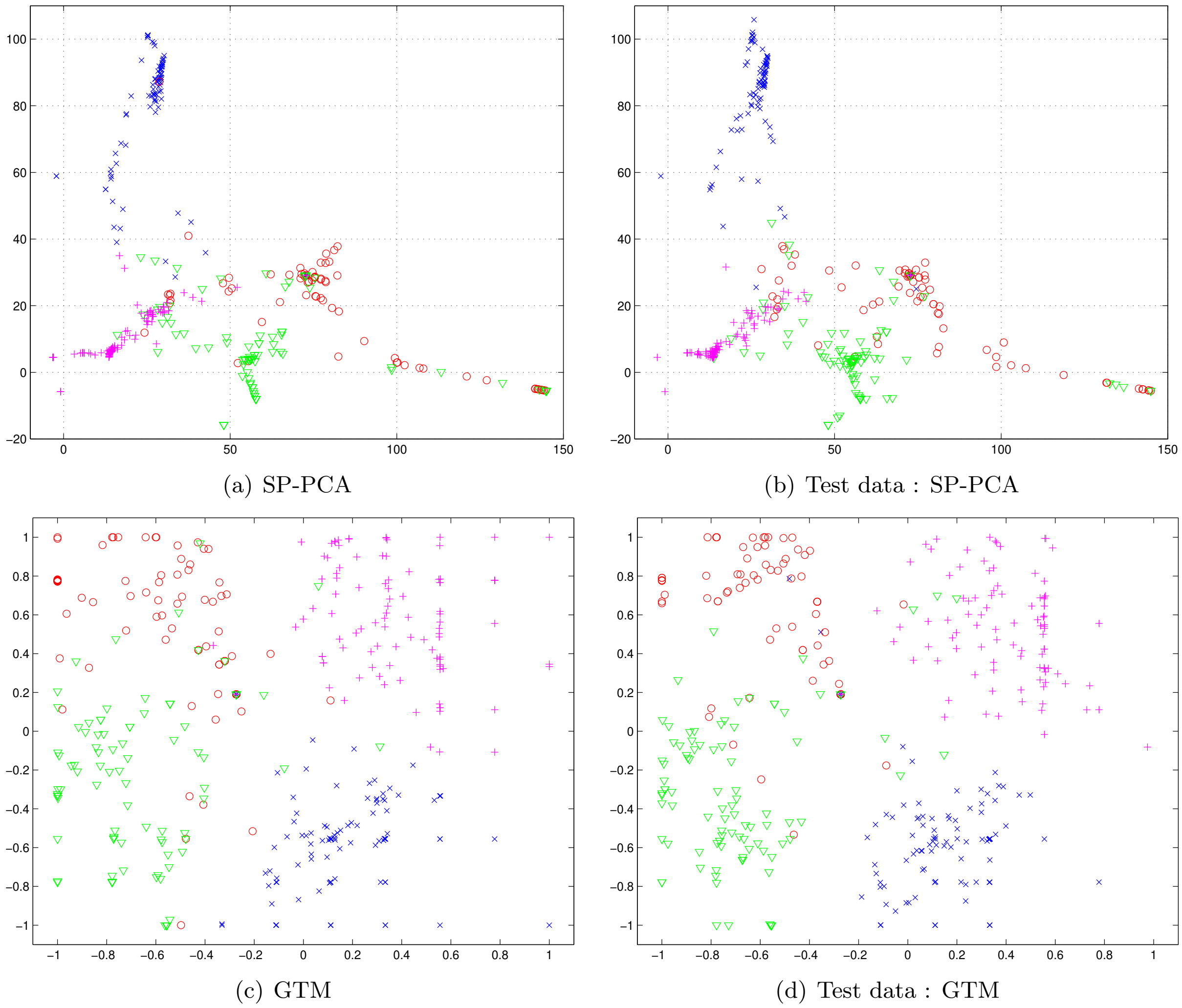


(e) GTM



(f) Test data : GTM

Figure 10: Projection by various methods of **binary** data from 100 documents each from sci.crypt ($\times$), sci.med ($\circ$), sci.space (∇) and soc.culture.religion.christianity ($+$)



(a) SP-PCA

(b) Test data : SP-PCA

(c) GTM

(d) Test data : GTM

Figure 11: Projection by various methods of **count** data from 100 documents each from sci.crypt ($\times$), sci.med ($\circ$), sci.space (∇) and soc.culture.religion.christianity ($+$)

different populations and projecting similar observed data points close to one another in the representation space. It was also observed to generalize well to unseen samples.

Future work includes extending the model to categorical data, finding a suitable objective function for semi-supervised training of the mixture model and extending the method to find non-linear lower dimensional latent spaces.

References

[BK99] David J. Bartholomew and Martin Knott. *Latent variable models and Factor analysis*, volume 7 of *Kendall's Library of Statistics*. Oxford University Press, New York, 2nd edition, 1999.

[BSW98] C. M. Bishop, M. Svensén, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21:203–224, 1998.

[CDS02] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems 14*, 2002.

[CPR00] Miguel A. Carreira-Perpinan and Steve Renals. Practical identifiability of finite mixtures of multivariate bernoulli distributions. *Neural Computation*, 12(1):141–152, 2000.

[Hof99] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999.

[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.

[KW56] J. Kiefer and J. Wolfowitz. Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics*, 27:887–906, 1956.

[Lin83] B. G. Lindsay. The geometry of mixture likelihoods : A general theory. *The Annals of Statistics*, 11(1):86–04, 1983.

[LS00] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 11*, pages 556–562, 2000.

[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.

[Red81] R. Redner. Note on the consistency of the maximum likelihood estimate for nonidentifiable distribution. *The Annals of Statistics*, 9(1):225–228, 1981.

[Rip96] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[Row98] Sam Roweis. EM algorithms for PCA and SPCA. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[TB99] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

[Tib92] R. Tibshirani. Principal curves revisited. *Statistics and Computation*, 2:183–190, 1992.

[Tip99] M. Tipping. Probabilistic visualisation of high-dimensional binary data. In *Advances in Neural Information Processing Systems 11*, 1999.

[Wu83] C. F. J. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95–103, 1983.

A The Newton-Raphson updates for use in EM algorithm

In this section, we describe how to get the NR update equations for the matrix A. The updates for V and $\mathbf{b}$ can be obtained in a similar manner.

Let $\tilde{l}_{kj} = \theta_{kj}\tilde{x}_{kj} - G(\theta_{kj})$. From Section 4.1, $\tilde{L}$ is given by

$$\tilde{L}(A, V, \mathbf{b}) = \sum_{k=1}^{c}\sum_{j=1}^{d}\{\theta_{kj}\tilde{x}_{kj} - G(\theta_{kj})\} = \sum_{k=1}^{c}\sum_{j=1}^{d}\tilde{l}_{kj}$$

The optimal value for matrix A will solve the following equations

$$\frac{\partial \tilde{L}}{\partial a_{kl}} = 0 \qquad \forall\, k = 1, \ldots, c \quad \text{and} \quad \forall\, l = 1, \ldots, \ell$$

Let A_t be the current value of A and A_{t+1} be the value of A at the next step. Then the NR equation is $A_{t+1} = A_t - H_t^{-1}GR_t$, where H_t is the Hessian matrix of $\tilde{L}$ and GR_t is its gradient at $A = A_t$.

The first and second derivatives of $\tilde{L}$ with respect to entries in A are

$$\frac{\partial \tilde{L}}{\partial a_{rs}} = \sum_{j=1}^{d}\frac{\partial \tilde{l}_{rj}}{\partial \theta_{rj}}V_{sj} = \sum_{j=1}^{d}(\tilde{x}_{rj} - g(\theta_{rj}))V_{sj}$$

$$\begin{aligned}\frac{\partial^2 \tilde{L}}{\partial a_{tu}\partial a_{rs}} &= -\sum_{j=1}^{d}V_{sj}\frac{\partial g(\theta_{rj})}{\partial a_{tu}} = -\sum_{j}V_{sj}\frac{\partial g(\theta_{rj})}{\partial \theta_{rj}}\frac{\partial \theta_{rj}}{\partial a_{tu}} \\ &= \begin{cases} 0 & \text{if} \quad r \neq t \\ -\sum_j V_{ju}V_{js}\frac{\partial g(\theta_{rj})}{\partial \theta_{rj}} & \text{if} \quad r = t \end{cases}\end{aligned}$$

Since $\frac{\partial^2 \tilde{L}}{\partial a_{tu}\partial a_{rs}}$ is zero when $r \neq t$, the rows of A can be updated independently of each other and the updates in Section 4.1 easily follow from the above equations.

B Consistency of the ML estimate

Let $\mathbf{X_i} = (X_{i1}, \ldots, X_{id})$ be iid draws of a d-dimensional random variable. Assume that the frequency function is $f(\mathbf{x}|\mathbf{s}, \mathbf{a})$, where $\mathbf{s} \in \Omega \subseteq \mathbb{R}^{k1}$ is a structural parameter and $G(\mathbf{a}) \in \Gamma$ is a distribution over incidental parameters $\mathbf{a} \in \mathbb{R}^{k2}$. $\gamma = (\mathbf{s}, G)$ is a generic point in the parameter space $\Omega \times \Gamma$. In the space $\Omega \times \Gamma$, we define the metric

$$\delta(\gamma_1, \gamma_2) = \delta((\mathbf{s}_1, G_1), (\mathbf{s}_2, G_2)) = \sum_{j=1}^{k1}|\tan^{-1}\mathbf{s}_{1j} - \tan^{-1}\mathbf{s}_{2j}| + \int_{\mathbb{R}^{k2}}|G_1(z) - G_2(z)|e^{-|z|}d\tau(z)$$

Let γ_0 be the 'true' parameter from which observed samples are drawn. The following are some of the assumptions made by Kiefer and Wolfowitz

Assumption 1 $f(\mathbf{x}|\mathbf{s}, \mathbf{a})$ is a density with respect to a σ-finite measure μ on a Euclidean space of which $\mathbf{x}$ is a generic point.

Assumption 2 It is possible to extend the definition of $f(\mathbf{x}|\gamma)$ so that the range of γ will be in $\bar{\Omega} \times \bar{\Gamma}$ and so that, for any $\{\gamma_i\}$ and γ^* in $\bar{\Omega} \times \bar{\Gamma}$, $\gamma_i \longrightarrow \gamma^*$ implies $f(\mathbf{x}|\gamma_i) \longrightarrow f(\mathbf{x}|\gamma^*)$ except perhaps on a set of $\mathbf{x}$ that has zero probability according to the true distribution.

Assumption 3 For any γ in $\bar{\Omega} \times \bar{\Gamma}$ and any $\rho > 0$, $w(\mathbf{x}|\gamma, \rho)$ is a measurable function of $\mathbf{x}$, where $w(\mathbf{x}|\gamma, \rho) = \sup f(\mathbf{x}|\gamma')$, the supremum being taken over all γ' in $\bar{\Omega} \times \bar{\Gamma}$ for which $\delta(\gamma, \gamma') < \rho$.

Assumption 5 For any $\gamma \in \bar{\Omega} \times \bar{\Gamma}$ we have, as $\rho \downarrow 0$,

$$\lim \; E\left[\log \frac{w(\mathbf{x}|\gamma, \rho)}{f(\mathbf{x}|\gamma_0)}\right]^+ < \infty$$

As defined in Section 4.2, let C_0 be the set of all parameters γ corresponding to the 'true' distribution $F(x/\gamma_0)$ (i.e., $C_0 = \{\gamma : F(x/\gamma) = F(x/\gamma_0) \; \forall \; x\}$). Let $\hat{\gamma}_n$ be an estimator of γ based on n observed samples of X and let $\hat{\Gamma}$ be the quotient topological space obtained from Γ obtained by identifying the set C_0 to a point $\hat{\gamma}_0$.

Definition. *The sequence of estimators* $\{\hat{\gamma}_n, \; n = 1, \ldots, \infty\}$ *is said to be strongly consistent in the sense of Redner if* $\lim_{m\to\infty} \hat{\gamma}_n = \hat{\gamma}_0$ *almost surely.*

Theorem. *If assumptions 1,2,3 and 5 are satisfied, the ML estimator of the parameter* $\gamma = (\mathbf{s}, G)$ *is strongly consistent in the sense of Redner.*

Verifying that the assumptions are satisfied for the model considered in this paper Our model consists of a system $X_{i1}, \ldots, X_{id}$, $i = 1, 2, \ldots$, independent draws of a d-dimensional random variable $\mathbf{X}$. The distribution $f(\mathbf{x}|\gamma)$ is determined by parameter $\gamma = (\mathbf{s}, G)$. Here $\mathbf{s} = (V, b) \in \Omega = \mathbb{R}^{(\ell+1)*d}$ is the structural part of the parameter which determines the subspace to which natural parameters of the exponential family distributions are constrained and $G \in \Gamma$ is the distribution of according to which the natural parameters are picked on the subspace. Γ consists of all the distributions G on $\mathbb{R}^\ell$ such that the corresponding density function $g(\mathbf{a}) = 0$ for $||\mathbf{a}|| > B$ (B is some constant fixed apriori).

Hence the model $f(\mathbf{x}|\gamma)$, with parameter $\gamma = (V, b, G)$ belonging to the space $\Omega \times \Gamma$ is specified by

$$\begin{aligned}
\mathbf{a} &\sim G && (2)\\
\boldsymbol{\theta} &= \mathbf{a}V + b && (3)\\
\log f(x_j|\theta_j) &= \log f_0(x_j) + x_j\theta_j - G(\theta_j) \quad j = 1, \ldots, d && (4)\\
f(\mathbf{x}|\boldsymbol{\theta}) &= \prod_{j=1}^{d} f(x_j|\theta_j) && (5)\\
& && (6)
\end{aligned}$$

From the definition of $f(\mathbf{x}|V, b, G)$, it follows immedietly that Assumptions 1 and 2 are satisfied. Assumption 3 is satisfied since both Ω and Γ are separable spaces.

To verify Assumption 5, note that $f(\mathbf{x}|\mathbf{s}, G)$ is uniformly bounded in $\mathbf{x}$, $\mathbf{s}$ and G (since the mean of the poisson is assumed to be bounded above). Hence $E[\log \omega] < \infty$.

Also, to show that $E[\log f(X_j|\gamma_0)] > -\infty$, it is sufficient to show that $E[\log |X_j|]^+ < \infty$ (by Lemma in Section 2 of [KW56]).

$$E[\log |X_j|]^+ \leq E[\log(|X_j - g(\theta_j)| + |g(\theta_j)|)]^+ \leq E[\log(|X_j - g(\theta_j)| + 1)]^+ + E[\log |g(\theta_j)|]^+$$

$E[\log |g(\theta_j)|]^+ \leq \infty$ since we have assumed that $P(\mathbf{a})$ is zero outside a bounded region and since $g(\theta_j)$ is a continuous function of $\mathbf{a}$ for all the distributions we are considering. That

$E[\log(|X_j - g(\theta_j)| + 1)]^+$ is $\leq \infty$ follows from the fact that variance of Poisson, Gaussian, Bernoulli and Exponential distributions is bounded if $\mathbf{a}$ and hence θ_j are bounded.

Gaussian case when the common variance parameter σ is considered unknown and estimated using ML: For this case, the ML estimator is consistent if we make an additional assumption that σ is bounded below by a small constant. This assumption ensures that $f(\mathbf{x}|\mathbf{s}, G)$ is uniformly bounded in $\mathbf{x}$, $\mathbf{s}$ and G and hence $E[\log \omega] < \infty$ which is needed to satisfy Assumption 5.