UNIVERSITY OF CALIFORNIA, SAN DIEGO

Nonparametric methods for learning from data

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Sajama

Committee in charge:

Professor Alon Orlitsky, Chair
Professor Sanjoy Dasgupta
Professor Bhaskar Rao
Professor Nuno Vasconcelos
Professor Ruth Williams

2006

The dissertation of Sajama is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2006

To Thomas John

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Alon Orlitsky, for introducing me to the subject of statistical learning and for the constant guidance, support and help he has given me throughout my Ph.D. program. He always encouraged me to think independently and methodically when looking at research problems and I am certain that the things I have learnt from him will help me well into the future.

I would like to thank Prof. Sanjoy Dasgupta who was generous with his time and helped me greatly by discussing my research problems. I also gained a lot from the several reading groups that he organized on topics related to machine learning. I would like to thank Thomas John who helped the research presented here in many ways including helping me with several of the proofs and algorithms and for helping me understand differential geometry. I would like to thank Prof. Bhaskar Rao, Prof. Nuno Vasconcelos and Prof. Ruth Williams for the many things I learnt in their classes and for serving on my doctoral committee. Prof. Toby Berger has advised and helped me several times over the last few years and I am thankful for having had the opportunity to with him. I want to thank members of my research group - Aldebaro, Anand, Junan, Krishna, Nikola and Prasad - and other friends at UCSD and elsewhere who have enriched the years I have spent here. Thojo, I will thank you personally.

The material presented in Chapter III has been published in Advances in Neural Information Processing Systems 2005. The material presented in Chapter IV has been published in the Proceedings of the International Conference on Machine Learning 2005. The material presented in Chapter VI has been published in the Proceedings of the International Conference on Machine Learning 2005 and as a chapter in the book 'Semi-supervised Learning', MIT press 2006. The dissertation author was the primary investigator and the first author of these publications.

VITA

1977	Born, Krishna, A. P., India
1998	Bachelor of Technology Indian Institute of Technology, Mumbai, India
2001	Master of Science Cornell University, Ithaca, New York
2006	Doctor of Philosophy University of California, San Diego, California

PUBLICATIONS

Sajama and A. Orlitsky, "Estimating and computing density based distance metrics", In Proceedings of the 22nd International conference on Machine learning, Morgan Kauffmann Publishers 2005

Sajama and A. Orlitsky, "Modifying Distances", In Semi-Supervised Learning, O. Chapelle, A. Zien, and B. Scholkopf, Editors, MIT Press, Boston 2005

Sajama and A. Orlitsky, "Supervised dimensionality reduction using mixture models", In Proceedings of the 22nd International conference on Machine learning, Morgan Kauffmann Publishers 2005

Sajama and A. Orlitsky, "Semi-parametric Exponential family PCA", In Advances in Neural information processing systems, 17, MIT press 2005

A. Dhulipala, A. Orlitsky and Sajama, "Recent results on compression of large alphabets", In 41st Annual Allerton Conference on Communication, Control, and Computing, 2003

A. Orlitsky, Sajama, N. P. Santhanam, K. Viswanathan and J. Zhang, "Practical algorithms for modeling sparse data", In IEEE International Symposium on Information Theory, 2004

FIELDS OF STUDY

Major Field: Engineering

Studies in Machine learning.

Professors Alon Orlitsky, Sanjoy Dasgupta, Bhaskar Rao, Nuno Vasconcelos, Ruth Williams

ABSTRACT OF THE DISSERTATION

Nonparametric methods for learning from data

by

Sajama
Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, and Control)
University of California, San Diego, 2006
Professor Alon Orlitsky, Chair

Developing statistical machine learning algorithms involves making various degrees of assumptions about the nature of the data being modeled. Non-parametric methods are useful when prior information regarding the parametric form of the model is unavailable or invalid. This thesis presents non-parametric methods for tackling various modeling requirements.

The first part of this thesis presents a pair of unsupervised and supervised linear dimensionality reduction techniques that are suitable for various data types like binary and integer along with real-valued data. They are based on a semi-parametric mixture of exponential family distributions where no parametric assumptions are made about the latent distribution and the parametric form of the noise distribution is to be chosen based on the data type, for example Bernoulli for binary data, etc. A key feature of the unsupervised method is that it guarantees asymptotic consistency of the estimated lower dimensional signal subspace, which is not guaranteed for other recently proposed methods. The supervised method finds the lower dimensional space that retains maximum possible information regarding the labels. We present efficient algorithms and experimental results for these methods.

The second part of this thesis considers unsupervised learning of a density based distance. We decompose the errors that can arise in approximating

these density based distances into estimation and computation components. We prove upper and lower bounds on the rate of convergence of the estimation error in terms of data dimensionality and smoothness of the data density. We present a method for constructing a graph on the data and a performance guarantee on the computation error when using this method. We also show an upper bound on the approximation error that applies to approximating distances using nearest-neighborhood based graphs and is applicable to several other similarity measuring algorithms. Finally, we show that this graph construction enables consistent approximation of the minimal geodesics themselves for the non-linear interpolation application and present experimental results.

Chapter I

Introduction

Recent decades have seen an explosive growth in the amount of data collected in fields spanning from social studies to marketing, online data analysis, drug discovery and biological research. This growth in the amount of available data has been made possible because of rapid growth in information technology and is expected to continue to increase with further improvements in our ability to collect and store data. The field of statistical machine learning is concerned with modeling this data in order to extract useful information from it. In other words, it involves coming up with a statistically accurate probabilistic summary of the way the samples behave.

Traditionally, machine learning algorithms have been classified into two main categories, unsupervised and supervised [43, 4, 5, 6, 7, 8]. Unsupervised learning deals with the problem of selecting a model from model space Θ based on a training set $\{x_i\}_{i=1}^n \subset \mathcal{X}$. In contrast to this, supervised learning works with training data of the form $\{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ where x_i's are thought of as input and the y_i's as output, and the model is selected so that it is possible to predict, as accurately as possible, the output y corresponding any given input x.

I.A Adapting to the charecteristics of data-sets

Developing algorithms that are practically applicable involves taking into account the nature of the data and its collection process. This dissertation presents methods that are adapted to work with these requirements imposed by the new characteristics of data-sets like high dimensionality, high volume, heterogeneous data types and differences in the relative costs of acquiring different components of the data.

An important feature of data-sets that makes machine learning challenging is that they are increasingly high dimensional. It is well known that learning becomes difficult as the dimension of the data sets increases. This difficulty is often called the curse of dimensionality - a term that was first used in [9] to describe the fact that the number of function evaluations required to perform optimization (within some given error tolerance) by exhaustive enumeration grows exponentially with the dimension of the space over which the function is defined. In the context of function approximation this curse can be seen as follows : if we must approximate a function of d variables and we know only that it is Lipschitz, say, then we need order $(1/\epsilon)^d$ evaluations on a grid in order to obtain an approximation scheme with uniform approximation error ϵ.

To see how the number of samples needed grows rapidly with data dimension in the case of statistical estimation, consider a $d+1$-dimensional data-set with the property that the first component y is dependent on the other components $\mathbf{x}$, through a model of the form $y_i = f(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,d}) + \text{noise}_i$. Suppose that we are only able to assume that f is a Lipschitz function of these variables and that the noise_i variables are in fact i.i.d. Gaussian with mean 0 and variance 1. It can be shown [11] that for any estimator $\hat{f}$, we have $\sup_f E(\hat{f} - f(\mathbf{x}))^2 = Const \sum N^{-2/(2+D)}, n \longrightarrow \infty$. According to [10], the curse of dimensionality in the context of statistical estimation refers to this slowing of the rate of convergence of the minmax estimation error with increase in data di-

mension. Note that by making certain stronger assumptions than the Lipschitz assumption made for this previous result, it is possible to show rates of convergence that are independent of the data dimension (see for example the function approximation result in [12]).

A common way to deal with this difficulty in high dimensions is to reduce the dimensionality of data, removing the 'noise' while retaining the useful or 'signal' part of the data. Chapters III and IV present methods for unsupervised and supervised, linear dimensionality reduction. One way to understand the complexity of a learning problem is to study the lower and upper bounds on the appropriate approximation errors. In Chapter VI, we derive such bounds in the context of unsupervised learning of a distance metric and show how the data dimension affects the achievable and best possible error rates under various assumptions for the data density.

Another aspect of data that needs to be accounted for in learning algorithms is that it can often be heterogeneous, i.e., some components of the data may be of a different type than being real valued. For example, text documents are often represented as binary or integer data and black-and-white images are represented as binary data. The dimensionality reduction techniques presented in Chapters III and IV are specifically equipped to deal with such situations.

Another issue of practical importance is to be able to deal with training data that is not complete. Often, certain parts of the training data are not available because of limitations in data collection process, for example, survey data might have missing components because of some people not wishing to share certain information or dropping out of the program after some time. Another cause of missing data could be that it is more expensive to collect some parts of the data and so a choice might be made to collect more of the inexpensive data in relation to the amount of expensive data. The case when the output variable y_i is expensive to collect and hence not available for a lot of the input variables x_i in the training set is known as semi-supervised learning [117]. The combination of expensive labeled

data and inexpensive unlabeled data occurs in many important application areas including text classification, computer vision and biological research (genetic or proteomic). In Chapter VI, we present analysis of a measure of similarity between data points which is based on a common assumption made in semi-supervised learning methods.

I.B Non-parametric methods

As discussed before, learning methods are essentially concerned with selecting a model from model space Θ based on the available training set. One important feature of the learning method is how the model space itself is chosen. For many statistical problems there are several possible solutions, differing in their suitability for different types of underlying 'truth' that governs the data sample. Learning methods are called parametric when they make inferences based on the assumption that the underlying distribution has a particular parametric form. In this case the model space Θ is indexed by the parameters of this distribution.

Nonparametric models differ from parametric models in that the model structure is not specified a priori, but is instead determined from data. This does not imply that non-parametric methods completely lack parameters, it only means that the number and nature of the parameters is flexible and not fixed in advance. A key feature of non-parametric methods is that they have certain desirable properties that hold under relatively mild assumptions regarding the underlying data distribution. Non-parametric methods are widely applicable because they are often significantly better when the true form of the underlying distribution is not known apriori and because of their relative insensitivity to outliers.

While nonparametric methods require no assumptions about the population probability distribution functions, they are based on some of the same assumptions as parametric methods, such as randomness and independence of the samples. Also, the price for wider applicability is that when reasonable parametric

assumptions can be made, parametric methods outperform non-parametric methods for small sample sizes.

In this paper we take advantage of this flexibility of nonparametric methods to model a wide range of distributions by modeling parts of our probabilistic models non-parametrically. On the other hand, we retain the advantage of using a parametric form, or more generally of using prior information, in order to constrain parts of our model in order to achieve good numerical results even in the presence of limited data samples. For example in the case of the dimensionality reduction methods, we assume that the 'signal' or lower dimensional distribution is non-parametric while we assume that the noise added follows a parametric form whose parameters are then estimated from the data. In the case of the density based distance, we consider the case when no parametric assumptions are made on the data density, but the function that maps the density to the measure of similarity is assumed to be known. This approach helps us combine the advantageous features of the parametric and nonparametric methods by making the model flexible in the necessary parts and by lowering the sample complexity by making assumptions in those areas where flexibility may not be as necessary.

I.C Dissertation outline

The first part of this dissertation considers the problem of dimensionality reduction which helps extract the latent subspace or signal, reduces noise and acts as a form of regularization. The probabilistic approach considered is also useful in low-dimensional density modeling for sparse data and in visualization. In Chapter II, we review related work on dimensionality reduction and motivate the need for the new techniques that we subsequently propose.

In Chapter III, we propose an unsupervised dimensionality reduction method suitable for various data types like binary and integer along with real-valued data. It is based on a semi-parametric mixture of exponential family distri-

butions where no parametric assumptions are made about the latent distribution and the parametric form of the noise distribution is to be chosen based on the data type, for example Bernoulli for binary data, etc. A key feature of this method is that it guarantees asymptotic consistency of the estimated lower dimensional signal subspace, which is not guaranteed for other recently proposed methods. We used Lindsay's theorem to propose an efficient expectation-maximization algorithm for estimating the latent distribution non-parametrically.

To illustrate the properties of this method, we present experiments on artificial data where binary and integer-valued samples are generated using a low-dimensional mixture model with various latent distributions. We found that this method outperformed other recently proposed methods in terms of recovering the true lower-dimensional subspace. We also present an experiment demonstrating its use as an effective density estimation method when data is sparse and showed improvement over another successful low-dimensional model, namely the probabilistic principal component analysis. Finally, we present visualization experiments which demonstrate that this method compares favorably to other methods in terms of sending similar points to nearby locations in the lower dimensional representation.

In Chapter IV, we extend this semi-parametric approach to supervised dimensionality reduction and propose a latent variable based method in which the subspace is chosen to retain the maximum possible information regarding the labels. Again, we present an efficient optimization algorithm for estimating the model based on bound maximization. Using experiments on data from the UCI repository, we demonstrate that this method yields more informative lower dimensional subspaces than than the latest kernel based method, where the informativeness of the low dimensional projection is measured using classification results on the projected data. We also illustrate the use of this method for supervised data visualization using real-world datasets.

Semi-supervised learning is the name given to learning classifiers or regression functions wherein one uses the unlabeled data along with labeled data in

the learning process. This is an important problem since in many practical applications, labeling data is expensive while large amounts of unlabeled data can be acquired easily. In the second part of this dissertation, we consider the problem of learning a metric based on data density which has applications in semi-supervised learning, clustering and non-linear interpolation. Unlabeled data help by allowing us to learn which data points belong to the same high density region of the data. Learning these density based metrics can help in incorporating into learning algorithms the prior information that two points are likely to be similar to one another if they belong to the same high density region. In Chapter V, we review related work on semi-supervised learning, density based distances and other learning situations that use this prior information.

In Chapter VI, we consider a definition of density based distances that is based on a Riemannian manifold structure defined as a function of data density. We decompose the errors that can arise in approximating these density based distances into estimation and computation components. Using techniques from mathematical statistics, we prove upper and lower bounds on the rate of convergence of the estimation error in terms of data dimensionality and smoothness of the data density. We present a method for constructing a graph on the data and showed a performance guarantee on the computation error when using this method. We also show an upper bound on the approximation error that applies to approximating distances using nearest-neighborhood based graphs and is applicable to several similarity measuring algorithms, including the ISOMAP [79]. This bound shows the effect of data dimensionality on the approximation error when using a neighborhood-based graph to measure distances. Finally, we show that this graph construction enables consistent approximation of the minimal geodesics themselves for the non-linear interpolation application and presented experimental results comparing the use of these metrics to a recently proposed algorithm for semi-supervised classification.

Chapter II

Dimensionality Reduction

Dimensionality reduction is the mapping of a high dimensional space into a lower-dimensional space. It is an important pre-processing step in many learning tasks because of increase in dimensionality of available data-sets which in turn is caused by increase in the ease of acquisition and storage of data.

Ideally, if data is not noisy and inherently lies on a lower dimensional space, we can do dimension reduction without loss of information. However, data is often noisy and there must be a loss of information. Dimensionality reduction is effective if the loss of information due to mapping to a lower-dimensional space is less than the gain due to simplifying the problem. Advantages of dimension reduction include reduction in computation time and in the number of parameters of a learning task. It can lead to better classification or regression accuracy since it can supress noise and act as a form of regularization. Lower dimensional models also have the advantage of greater interpretability. Reducing dimensions plays an important role in exploratory data analysis where it can help in visualizing the data structure in terms of groups, outliers etc.

Two main approaches to dimensionality reduction are feature selection and feature extraction. In feature selection, those dimensions of the data-set that contain maximal information are retained and the others are discarded, i.e, the mapping which defines dimensionality reduction is a projection along the axis

of the feature space. This has the advantage that fewer measurements need to be made when more samples are collected in the future. In feature extraction, the mapping which defines dimensionality reduction is a more complicated function than projection along the axes. In this dissertation, we will be presenting linear feature extraction methods, where the lower dimensional space is a linear space (a subspace with some displacement added) contained in the original higher-dimensional space. Linear dimensionality reduction is used extensively in signal processing, data compression, statistics, machine learning, machine perception, and data mining. It is a core component of technologies as diverse as face recognition, web searching, visual target tracking, audio source separation, and image compression.

Optimal feature extraction involves picking the mapping which maximizes some objective. In the case of unsupervised dimension reduction the goal is to obtain good signal representation, i.e., the goal is to represent the samples accurately in the lower dimensional space. When class labels are available and we are doing supervised dimension reduction, the goal of feature extraction is to enhance the class discriminatory information in the lower-dimensional space.

II.A Latent variable models

A latent variable model specifies a joint distribution of a set of random variables, some of which are unobservable (and hence called latent variables). One of the most important uses of latent variables is in dimensionality reduction where it is used to capture in a small set, the interrelationships of many variables. This is the idea behind factor analysis and the more recent applications of linear structural models.

Another reason for the popularity of latent variables is that they occur in many fields where statistical methods are used including social sciences and text analysis. For instance, the occurrence or non-occurrence of words in a document is

often modeled using a latent variable model where the latent variables are thought of as representing the topic, authors writing style, etc. We will denote the collection of manifest or observed random variables by $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and the collection of latent or hidden random variables by $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_L)$. For the sake of notational simplicity, we will not distinguish between random variables and the values they take.

Since only $\mathbf{x}$ can be observed, any inference is based on its distribution

$$p(\mathbf{x}) = \int p(\boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{II.1}$$

where $p(\boldsymbol{\theta})$ is the prior distribution over the latent variables. Given an observation $\mathbf{x}$, the posterior distribution over $\boldsymbol{\theta}$ is given by the Bayes rule

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}.$$

From Equation II.1, it is clear that $p(\boldsymbol{\theta})$ and $p(\mathbf{x}|\boldsymbol{\theta})$ are not uniquely identified for a given $p(\mathbf{x})$. A commonly used simplification that improves this identifiability situation is to assume that $x_1, x_2, \ldots, x_d$ are independent when conditioned upon the latent variable $\boldsymbol{\theta}$. This is a reasonable assumption since we often wish to model the interrelationships between $x_1, x_2, \ldots, x_d$ using the latent variables $\boldsymbol{\theta}$. For the same reason the number of latent variables, L, is assumed to be much smaller than d.

Many of the models we will be concerned with are instances of a General Linear Latent Variable Model (GLLVM). In this model, the conditional distribution is assumed to belong to the one-parameter exponential family, whose parameter is determined by the latent variables.

$$p(x_i|\alpha_i) = F_i(\mathbf{x}_i) G_i(\alpha_i) \exp(\alpha_i u_i(x_i)) \qquad i = 1, 2, \ldots, d$$

where α_i is some function linear function of $\boldsymbol{\theta}$. For a discussion of the properties of this model please see [22].

II.B Unsupervised dimension reduction

Unsupervised dimension reduction deals with the problem of finding a suitable mapping from $\mathcal{R}^d$ to a lower dimensional space $\mathcal{R}^L$ based on a training sample $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ which are n samples of $\mathbf{x}$.

II.B.1 Principal component analysis

Principal component analysis (PCA) is widely used for dimensionality reduction with applications ranging from pattern recognition and time series prediction to visualization. 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 $\boldsymbol{\theta}_i$. This is equivalent to choosing a subspace that maximizes the empirical variance of the projections of the data points onto the subspace.

Two basic methods for performing the PCA computations are : the power method and the Jacobi method. The power method computes the eigenvalues one by one starting with the largest one (which is associated with the principal component that contains most of the information), then moving to the next largest and so on. This method is known to converge to the optimal solution [1]

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 $\boldsymbol{\theta} \in \mathcal{R}^d$ [13]. 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.

II.B.2 Factor and latent trait analysis

Factor analysis [22] was invented more than a 100 years ago by psychologist Charles Spearman, who used it to postulate that a general mental ability, or g, underlies and shapes human performance in a variety of tests. Factor analysis aims to explain (most of) the variability of the observable random variables in terms of a few latent variables called factors. It is based on a latent variable model, a key feature of which is that the latent distribution $P(\boldsymbol{\theta})$ is assumed to be Gaussian. Another feature of the factor model is that the conditional distribution $P(\mathbf{x}|\boldsymbol{\theta})$ models the additive noise and is also usually assumed Gaussian for modeling real-valued data.

Factor analysis can be used to formulate a probabilistic alternative to PCA called Probabilistic PCA (PPCA) [23, 24]. This probabilistic formulation of PCA offers several advantages like allowing statistical testing, application of Bayesian inference methods and naturally accommodating missing values [23].

Latent trait analysis (LTA), a form of latent structure analysis [2], is used for the analysis of categorical data. This model is similar to PPCA in that it assumes that the latent distribution is Gaussian. In order to model binary data, this model assumes that the conditional distribution $P(\mathbf{x}|\boldsymbol{\theta})$ is Bernoulli. Tipping [15] proposes a binary data visualization technique based on the latent trait model.

II.B.3 Other methods

Collins et. al. [13] proposed a generalization of PCA using exponential family distributions. Like PCA, this generalization is not associated with a probability density model for the data. Non-negative matrix factorization [19] 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 (PLSI) [20] 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 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 [18, 21]. 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 [28] used a semi-parametric latent variable model for estimation of principle curves. 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.

II.C Supervised dimension reduction

Supervised dimension reduction deals with the problem of finding a suitable mapping from $\mathcal{R}^d$ to a lower dimensional space $\mathcal{R}^L$ based on a training sample $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots, (\mathbf{x}_n, y_n)$ which are n samples of $(\mathbf{x}, y)$, where y is the class label for $\mathbf{x}$. The goal is to maximize class discriminatory information contained in the lower dimensional points to which the $\mathbf{x}_i$ are mapped.

II.C.1 Linear discriminant analysis

Linear discriminant analysis (LDA) is a standard tool used for classification where an observation is classified to the class with centroid closest to the observation, where the distance is measured in the Mahalanobis metric using the pooled within group covariance matrix. This procedure is equivalent to assuming that the feature vectors belonging to the two classes have a Gaussian distribution with a different mean, but a common covariance matrix among the classes. If the classes are further assumed to have equal prior probabilities, assigning an observation to the class that has maximum posterior probability is equivalent to the classification rule used in LDA [4, 3].

Let c denote the number of classes that the output variable y belongs to. The LDA classification rule means that all the relevant distance information is contained in the at most $c-1$ dimensional subspace of $\mathcal{R}^d$ spanned by the c group centroids. A reduced form of LDA due to Fisher and Rao adds a graphical component to the procedure. One finds the L $(< c)$ dimensional subspace of $\mathcal{R}^d$ in which the group centroids are maximally separated (once again using the Mahalanobis metric confined to this subspace) and then classifies new data to the closest centroid in the reduced space. This further reduction in dimensions, beyond c can lead to illustrative graphical representations of the data points and also more stable classifiers.

II.C.2 Sliced inverse regression and principal hessian directions

Sliced inverse regression (SIR) and Principal hessian directions (pHd) are dimensionality reduction methods [53, 54] based on the following model of data

$$y = g(\beta_1 \mathbf{x}, \beta_2 \mathbf{x}, \ldots, \beta_L \mathbf{x}, \epsilon)$$

The random error ϵ is assumed to be independent of $\mathbf{x}$, but its probability distribution is unknown. This model leads to dimensionality reduction since the relationship between $\mathbf{x}$ and y is determined only through $\beta_1 \mathbf{x}, \beta_2 \mathbf{x}, \ldots, \beta_L \mathbf{x}$. The

β_i's are termed effective dimension-reduction (e.d.r.) directions. SIR and pHd are two different methods of finding these directions.

SIR proceeds by partitioning the range of the response variable y into a set of slices, and the sample means of the observations $\mathbf{x}$ are computed within each slice. This can be viewed as a rough approximation to the inverse regression of $\mathbf{x}$ on y. Noting that the inverse regression must lie in the effective subspace if the forward regression lies in such a subspace, principal component analysis is then used on the sample means to find the effective subspace. It can be shown [53] that this approach can find effective subspaces, but only under strong assumptions on the marginal distribution $p(\mathbf{x})$ (the marginal distribution must be elliptically symmetric).

Let $f(\mathbf{x})$ be the regression function $E(y|\mathbf{x})$, which is a d dimensional function. pHd works with the Hessian matrix $H(\mathbf{x})$, the p by p matrix with the $i-j^{th}$ entry equal to $\frac{\partial^2}{\partial x_i\,\partial x_j} f(\mathbf{x})$. The Hessian matrix typically varies as $\mathbf{x}$ changes unless the surface is quadratic. Difficulties associated with the curse of dimensionality arise quickly if we were to estimate it for each location. Instead, we turn to the average Hessian, $\bar{H} = EH(\mathbf{x})$. The principal Hessian directions are defined to be the eigenvectors of the matrix $\bar{H}\Sigma_{\mathbf{x}}$, where $\Sigma_{\mathbf{x}}$ denotes the covariance matrix of $\mathbf{x}$. With right-multiplication of $\mathbf{x}$ the procedure becomes invariant under affine transformation of $\mathbf{x}$. This is an important property to have for our purpose of visualization and dimension reduction. It can be shown that the pHds with nonzero eigenvalues are in the e.d.r. space.

II.C.3 Mixture discriminant analysis

As described in Section II.C.1, 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. Mixture discriminant analysis (MDA) [35] generalizes LDA by approximating each of the classes by a mixture of Gaussians all of which have a

common covariance matrix. In MDA, like in LDA, all the means of the Gaussians are constrained to lie in an L dimensional subspace. The mixture model is estimated using the EM algorithm [31] and the lower dimensional representation is obtained by projecting the data points into the L dimensional space in which the means lie. The authors describe further modifications to this method to improve performance including shrinking the means of a single class toward a common center and using flexible discriminant analysis [36].

Heteroscedastic discriminant analysis (HDA) extends LDA by allowing each of the classes to have its own covariance with the expense of resorting to numerical multivariate optimization to find the low-dimensional transform [37].

II.C.4 Kernel Dimensionality Reduction

Kernel Dimensionality Reduction (KDR) [34] solves a problem of feature selection in which the features are linear combinations of the components of X. In particular, it assumes that there is an L-dimensional subspace S such that

$$p(y|\mathbf{x}) = p(y|\Pi_S\mathbf{x}) \tag{II.2}$$

for all $\mathbf{x}$ and y, where Π_S is the orthogonal projection of $\mathcal{R}^d$ onto S. A key feature of KDR is that the distributions $p(y|\Pi_S\mathbf{x})$ and $p(\mathbf{x})$ are treated nonparametrically. Using the fact that finding a subspace S with the property II.2 is equivalent to finding a projection Π_S which makes y and $(I - \Pi_S)\mathbf{x}$ conditionally independent given $\Pi_S\mathbf{x}$, KDR turns the dimensionality reduction problem into an optimization problem by expressing conditional independence in terms of covariance operators on reproducing kernel Hilbert spaces.

II.C.5 Other methods

Many regression methods inherently perform some form of linear or non-linear dimensionality reduction in the process of estimating the regressor. For example, the classical two-layer neural networks involve a linear transformation in

the first layer, which can be seen as attempting to estimate an effective subspace based on specific assumptions about the regressor. Similar comments apply to projection pursuit regression [51], ACE [58] and additive models [52], all of which provide a methodology for dimensionality reduction in which an additive model is assumed for the regressor.

Some methods use approximations of the error rate based on Bhattacharyya bound or on an interclass divergence criterion [59, 60, 61]. These approximations make use of class-conditional density functions, and they must be accompanied by a parametric estimation of the densities followed by numerical optimization of the approximation. Gaussian assumption usually needs to be made about the class-conditional densities to make optimization tractable.

A related dimension reduction problem is feature selection where the goal is to select a subset of the features that have the most information about the labels. For a recent review of feature selection methods, please see [57]. [56] presents a method to select features based on the mutual information criterion.

Chapter III

Semi-parametric exponential family principal component analysis (SP-PCA)

In this chapter, we present a linear, unsupervised dimensionality reduction method. We show that it has the advantage of being provably asymptotically consistent and demonstrate using experiments that is compares favorably to the state of the art.

III.A Motivation and overview

Many of the dimension reduction methods recently proposed in the machine learning literature can be thought of as special cases of latent variable modelling which is commonly used in statistics to summarize observations [22]. For this reason, we use the language of latent variable models to describe these methods in the process of motivating out approach.

As explained in Chapter II, PCA has a quasi-probabilistic interpretation - it 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 [23].

Probabilistic PCA (PPCA) [23, 24] borrows from one popular latent variable model called factor analysis to propose a probabilistic alternative PCA. A key feature of this probabilistic model is that the latent distribution $P(\boldsymbol{\theta})$ is also assumed to be Gaussian since it leads to simple and fast model estimation, i.e., 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. The projection obtained using a Gaussian latent distribution tends to be skewed toward the center [23] 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, called semi-parametric PCA (SP-PCA), where no assumptions are made about the distribution of the latent random variable $\boldsymbol{\theta}$. Non-parametric latent distribution estimation allows us to approximate data density better than previous schemes and hence gives better low dimensional representations. In particular, multi-modality of the high dimensional density is better preserved in the projected 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 and they have also been used in several recently proposed dimensionality reduction schemes [15, 18, 21, 13].

The dimension reduction methods which assume some parametric form for the latent distribution [23, 15, 18, 21] do not guarantee consistent estimation of the low-dimensional 'data-space' when the true latent distribution does not satisfy the assumptions made. On the other hand PCA and Exponential PCA [13] do not assume even the existence of a latent distribution. Consistent estimation of the lower dimensional space is not guaranteed by Exponential PCA and it is known that for some exponential family conditional distributions, this method has significant asymptotic bias [14]. We show that using maximum likelihood approach with the SP-PCA model guarantees consistent estimation of the low-dimensional space modulo identifiability.

We use Lindsay's non-parametric maximum likelihood estimation theorem to reduce the estimation problem to one with a discrete prior with large enough support set size. It turns out that this choice gives us a prior which is 'conjugate' to all exponential family distributions, allowing us to give a 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 experiments with SP-PCA (with Gaussian conditional density) and compare it to PCA and PPCA [23]. 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.

III.B 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 $P(\mathbf{x}|\boldsymbol{\theta})$.

III.B.1 Conditional distribution

We assume that $P(\boldsymbol{\theta})$ adequately models the dependencies among the components of $\mathbf{x}$ and hence that the components of $\mathbf{x}$ 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* [29]. 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 [13]. Note that by using different distributions for the various components of $\mathbf{x}$, we can model mixed data types.

III.B.2 Latent distribution

Like previous latent variable methods, including PCA, we constrain the latent variable $\boldsymbol{\theta}$ to an ℓ-dimensional Euclidean subspace of 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 $\mathbf{a} \in \mathcal{R}^\ell$,

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

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 R^ℓ. Lindsay's mixture non-parametric maximum likelihood estimation (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 [27]. Hence if ML is the chosen parameter estimation technique, the SP-PCA 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 we propose to use pruning to reduce the number of components during model estimation to help both in computational speed and model 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.

III.C 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 (III.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 chapter, 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 [23, 15, 21].

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 [22] by the ℓ-dimensional minimal sufficient

statistic

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

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 [13]). For the Gaussian case, this representation is the usual Euclidean projection.

III.D Discussion of the model

III.D.1 The Gaussian case

When the exponential family distribution chosen is Gaussian, the model is a mixture of n spherical Gaussians 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. III.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.

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

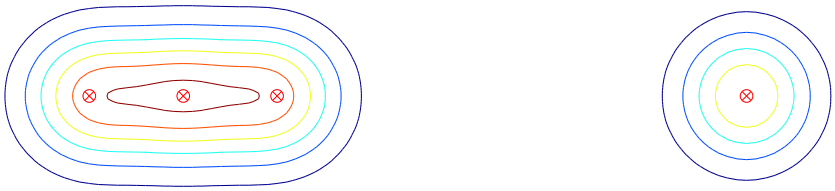


Figure III.1: Subspace aligned variance approximated by clustered but slightly spread out mixture component mean parameters $\otimes$

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). We found that SP-PCA differs significantly from PPCA in the predictive power of the low-dimensional density model (see Section III.G).

III.D.2 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.

III.D.3 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.

III.E 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 R^ℓ 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, $\mathcal{F} = \{F(x/\gamma), \quad \gamma \in \Gamma\}$ (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 [30]).

If a set of distributions parametrized by a (V, b) and $P(\mathbf{a})$ is not identifiable, it will not be possible distinguish some parameters from one another based on the density $p(\mathbf{x})$ and hence it will not be possible to recover the latent subspace (V, b). Consider for example the following two mixtures of bernoulli distributions, both of which represent the same distribution $p(\mathbf{x})$ in a 3-dimensional binary space. We use the convention of denoting mean parameters of exponential family distributions using μ and the mixing distribution of a mixture using $\pi_1, \ldots, \pi_k$ where k is the number of components in the mixture distribution. In terms of the mean parameters, the two mixture distributions are

1. Distribution 1 : $\pi_{11} = 0.25, \mu_{11} = (1, 0.5, 0.5)$ and $\pi_{12} = 0.75, \mu_{12} = (1/3, 0.5, 0.5)$
2. Distribution 2: $\pi_{21} = 0.25, \mu_{21} = (0.5, 0, 0.5)$ and $\pi_{22} = 0.75, \mu_{22} = (0.5, 2/3, 0.5)$

These mixture distributions, when translated into the natural parameter space correspond to

1. Distribution 1 : $\pi_{11} = 0.25, \theta_{11} = (\infty, 0, 0)$ and $\pi_{12} = 0.75, \theta_{12} = (-\log(2), 0, 0)$
2. Distribution 2 : $\pi_{21} = 0.25, \theta_{21} = (0, -\infty, 0)$ and $\pi_{22} = 0.75, \theta_{22} = (0, \log(2), 0)$

The natural parameters of the first distribution lie on a 1-dimensional subspace parallel to the first natural parameter axis and the parameters of the second distribution lie on a line parallel to the second natural parameter axis. It is easily verified that though the signal subspace of both these distribution is vastly different (the subspaces are perpendicular to one another), they induce the same distribution on the 3-dimensional binary space and hence cannot be distinguished from one another using samples from the binary space.

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 [25]. 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 1. *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.*

Consistency of estimating the subspace (V, b), under the assumption of a conditional distribution model $p(y|\mathbf{x})$ and some assumptions on p_U, follows by verifying that the assumptions of Kiefer and Wolfowitz [26] are satisfied. 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 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.

Theorem 2. *If* $P(\mathbf{a})$ *is assumed to be zero outside a bounded subset of* R^ℓ*, the ML estimator of parameter* $(V, b, P(\mathbf{a}))$ *is strongly consistent for Gaussian, Binary and Poisson conditional distributions.*

Proof. Assume that the frequency function of the conditional distribution $p_{y|\mathbf{x}}$ is $f(\mathbf{x}|\mathbf{s}, \mathbf{a})$, where $\mathbf{s} \in \Omega \subseteq \mathcal{R}^{k1}$ is a structural parameter and $G(\mathbf{a}) \in \Gamma$ is a distribution over incidental parameters $\mathbf{a} \in \mathcal{R}^{k2}$. $\gamma = (\mathbf{s}, G)$ is a generic point in the parameter space $\Omega \times \Gamma$. In the space $\Omega \times \Gamma$, we define the metric

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

Let γ_0 be the 'true' parameter from which observed samples are drawn. It follows from the proof of Kiefer and Wolfowitz's Theorem [26] that to prove our claim, it is sufficient to verify the following assumptions for the density models that we are considering.

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

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

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 $\mathbf{s}=(V,b)\in\Omega=\mathcal{R}^{(\ell+1)*d}$ 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 according to which the natural parameters are picked on the subspace. Γ consists of all the distributions G on $\mathcal{R}^\ell$ 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

$$\mathbf{a} \sim G \tag{III.2}$$

$$\boldsymbol{\theta} = \mathbf{a}V+b \tag{III.3}$$

$$\log f(x_j|\theta_j) = \log f_0(x_j)+x_j\theta_j-G(\theta_j)\quad j=1,\ldots,d \tag{III.4}$$

$$f(\mathbf{x}|\boldsymbol{\theta}) = \prod_{j=1}^{d} f(x_j|\theta_j) \tag{III.5}$$

(III.6)

From the definition of $f(\mathbf{x}|V, b, G)$, it follows immediately 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 [26]).

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

$E[\log |g(\theta_j)|]^+ \leq \infty$ 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. Note that this argument holds whether $\mathbf{x}$ correspond to the mean parameters or the natural parameters. In fact, it would hold for $\mathbf{x}$ corresponding to any 1-1, smooth, invertible transformation of the mean or natural parameters.

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. □

III.F Model estimation

III.F.1 Algorithm

We present an EM algorithm for estimating parameters of a finite mixture model with the components constrained to an ℓ-dimensional Euclidean subspace.

We propose an iterative re-weighted least squares (IRLS) method for the maximization step along the lines of generalized linear model estimation. Use of weighted least squares does not guarantee monotone increase in data likelihood. To guarantee convergence of the algorithm, we can check the likelihood of data at the IRLS update and decrease step size if necessary. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be iid samples drawn from a d-dimensional density $P(\mathbf{x})$, c be the number of mixture components and let the mixing density be $\Pi = (\pi_1, \ldots, \pi_c)$. Associated with each mixture component (indexed by k) are parameter vectors $\boldsymbol{\theta}_k$ 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 components of $\mathbf{x}$ correspond to the same exponential family for ease of notation. For each observed $\mathbf{x}_i$ there is an unobserved 'missing' variable $\mathbf{z}_i$ which is 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. If $\mathbf{y}_l$ is a vector, we use y_{lm} to denote its m'th component.

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 (III.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,

$$\hat{z}_{ik} = E(z_{ik}) = \frac{\pi_k P(\mathbf{x}_i/\boldsymbol{\theta}_k)}{\sum_{m=1}^{c} \pi_m P(\mathbf{x}_i/\boldsymbol{\theta}_m)} \quad ; \quad \tilde{x}_{kj} = \frac{\sum_{i=1}^{n} \hat{z}_{ik} x_{ij}}{\sum_{i=1}^{n} \hat{z}_{ik}}$$

In the M-step we update Π, V, b, and $\mathbf{a}_k$ in the following manner

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

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 [13]) 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 [29], 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. Similarly, we find that each column of V and each dimension of $\mathbf{b}$ can be updated independently.

$\mathbf{a}_i$ is updated by adding $\delta\mathbf{a}_i$ calculated using

$$(V\Omega_i V')\delta\mathbf{a}_i = GR_i \quad ; \quad [\Omega_i]_{qq} = \frac{\partial g(\theta_{iq})}{\partial \theta_{iq}} \quad ; \quad [GR_i]_{l1} = \sum_{j=1}^{d} (\tilde{x}_{ij} - g(\theta_{ij}))V_{lj}$$

Here the function $g(\theta)$ is as defined in Section III.B and depends on the member of the exponential family that is being used. Each column of the matrix V, $\mathbf{v}_s$, is updated by adding $\delta\mathbf{v}_s$ calculated using

$$(A'\Omega_s A)\delta\mathbf{v}_s = GR_s \quad ; \quad [\Omega_s]_{kk} = \frac{\partial g(\theta_{ks})}{\partial \theta_{ks}} \quad ; \quad [GR_s]_{l1} = \sum_{k'=1}^{c} (\tilde{x}_{k's} - g(\theta_{k's}))A_{k'l}$$

Each component of vector $\mathbf{b}$, b_s, is updated by adding δb_s calculated using

$$H_s \delta b_s = GR_s \quad ; \quad H_s = \sum_{k'=1}^{c} \frac{\partial g(\theta_{k's})}{\partial \theta_{k's}} \quad ; \quad GR_s = \sum_{k'=1}^{c} (\tilde{x}_{k's} - g(\theta_{k's}))$$

III.F.2 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 $\overset{\max}{i}|P(\mathbf{x}_i|\boldsymbol{\theta}_{k1}) - P(\mathbf{x}_|\boldsymbol{\theta}_{k2})| < C_2$, 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.

III.F.3 Convergence and computational complexity

It is easy to verify that our model satisfies the continuity assumptions of Theorem 2, [32], 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 III.D.1). 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. [42] 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.

III.F.4 Model selection

While any of the standard model selection methods based on penalizing complexity could be used to choose ℓ, an alternative 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.

III.G 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 [22], 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 when latent variable models are used for density approximation, data analysis and visualization.

III.G.1 Efficacy of SP-PCA in recovering the lower dimensional subspace

We present experiments demonstrating consistency properties of the Sum-squared estimator (PCA), Variance Ignoring estimator (Var-Ig), Maximum conditional likelihood estimator (MCL) and Semiparametric-PCA (SP-PCA). Figure III.2 shows the canonical angles between estimated subspace and V_0 when the conditional distribution is Bernoulli and the natural parameters are constrained to true lower dimensional subspace while Figures III.3 and III.4 show examples where the conditional distribution is Poisson. These experiments demonstrate that the subspace estimated by Exponential PCA can be very far from the true subspace.

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

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

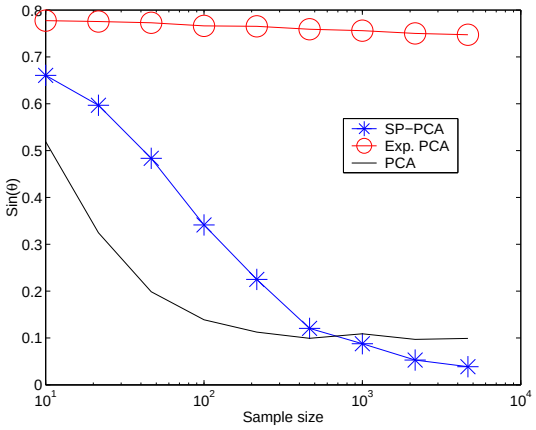
In all of these experiments, we found that the limit points of Exponential PCA subspace were either close to one of the axes or close to the true subspace. This bias toward the axes is explained [14]. Another interesting thing we noticed is that whether or not Exponential PCA converges to the true subspace can depend strongly on the latent distribution (this is demonstrated in the Poisson example in Figures III.3 and III.4).

III.G.2 Use of SP-PCA as a low dimensional density model

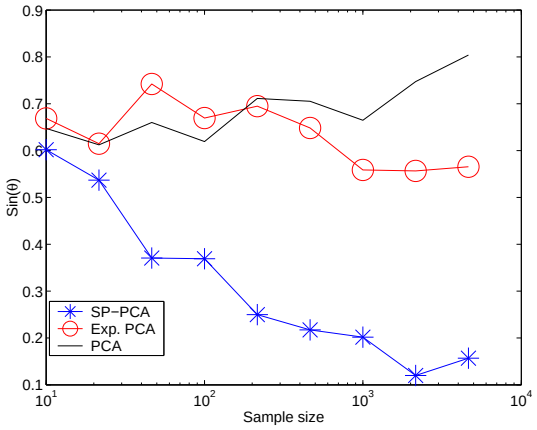
The Tobamovirus data which consists of 38 18-dimensional examples was used in [23] to illustrate properties of PPCA. PPCA and SP-PCA can be thought of as providing a range of low-dimensional density models for the data. The complexity of these densities increases with and is controlled by the value of ℓ (the projected space dimension) starting with the zero dimensional model of an isotropic Gaussian. For a fixed lower dimension ℓ, SP-PCA has greater approximation capability than PPCA. In Table III.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 combines flexible density estimation and excellent generalization even when trained on a small amount of data.

III.G.3 Visualization results on discrete datasets

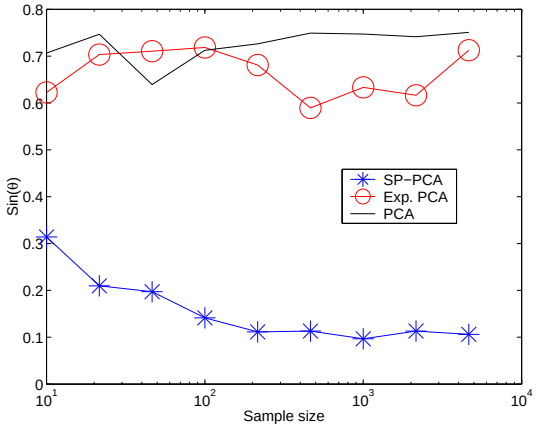
We present experiments on 20 Newsgroups dataset comparing SP-PCA to PCA, exponential family GTM [21] and Exponential family PCA [13]. Data for the first set of simulations was drawn from comp.sys.ibm.pc.hardware, comp.sys.-



(a) $V_0 = (0.62, 0.46, 0.64)$, $p_0 \sim$ uniform over $\{-3, -2, -1, 0, 1, 2, 3\}$



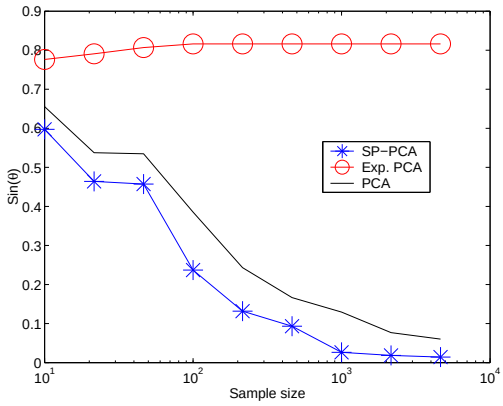
(b) $V_0 = (2, 1)$, $p_0 \sim \mathcal{N}(0, 1)$



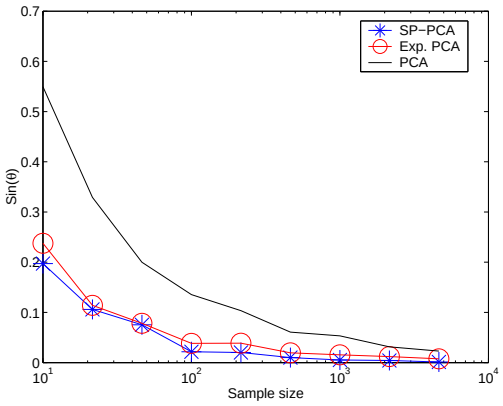
(c) $V_0 = (2, 1)$, $p_0 \sim \mathcal{N}(0, 10)$

Figure III.2: Norm of sines of canonical angles to the correct subspace to which the distribution over 'natural parameters' of the Bernoulli mixture are constrained.

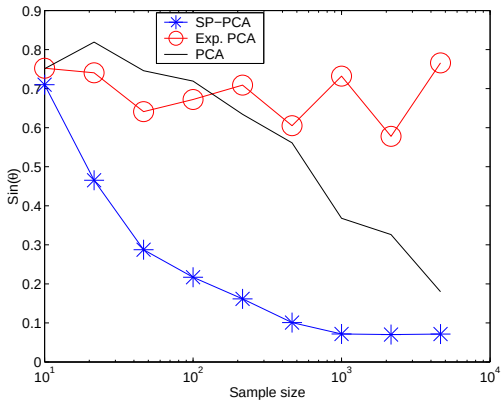
mac.hardware and sci.med newsgroups. A dictionary size of 150 words was chosen and the words in the dictionary were picked to be those which have maximum mutual information with class labels. 200 documents were drawn from each of the three newsgroups to form the training data. Two-dimensional representations obtained using various methods are shown in Fig. III.5. In the projection obtained



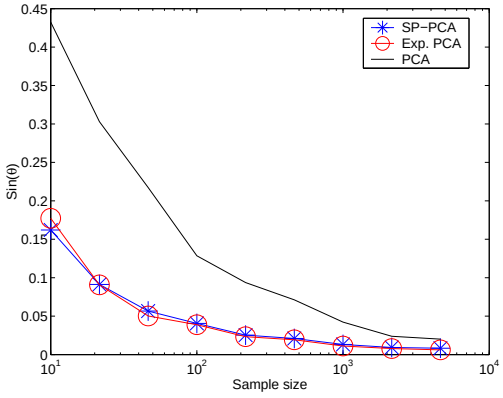
(a) $V_0 = (1, 1, 1)$, $p_0 \sim$ uniform over $\{-2, 0\}$



(b) $V_0 = (1, 1, 1)$, $p_0 \sim$ uniform over $\{0, 2\}$



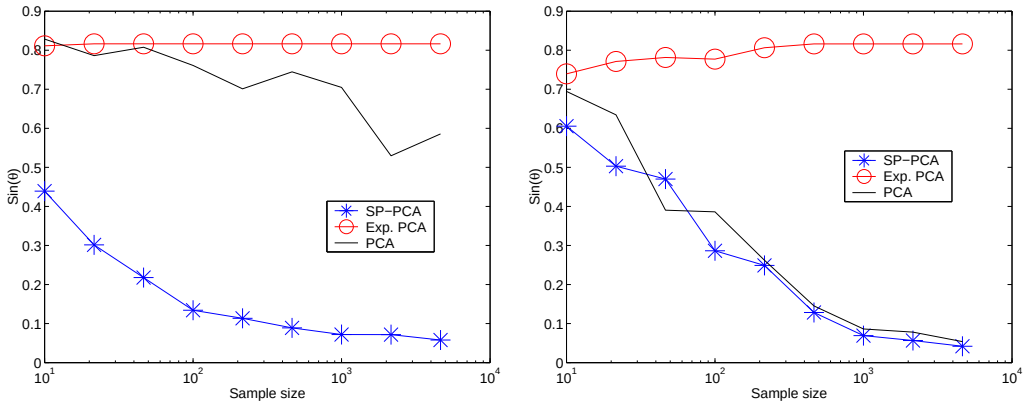
(c) $V_0 = (1, 1, 1)$, $p_0 \sim \mathcal{N}(0, 1)$



(d) $V_0 = (1, 1, 1)$, $p_0 \sim \mathcal{N}(0, 3)$

Figure III.3: Norm of sines of canonical angles to the correct subspace to which the distribution over 'natural parameters' of the Poisson mixture are constrained - Part 1.

using PCA, Exponential family PCA and Bernoulli GTM, the classes comp.sys.ibm.pc.hardware and comp.sys.mac.hardware were not well separated in the 2D space. This result (Fig. III.5(c)) was presented in [21] and the the overlap between the two groups was attributed to the fact that they are very similar and hence share many words in common. However, SP-PCA was able to separate the

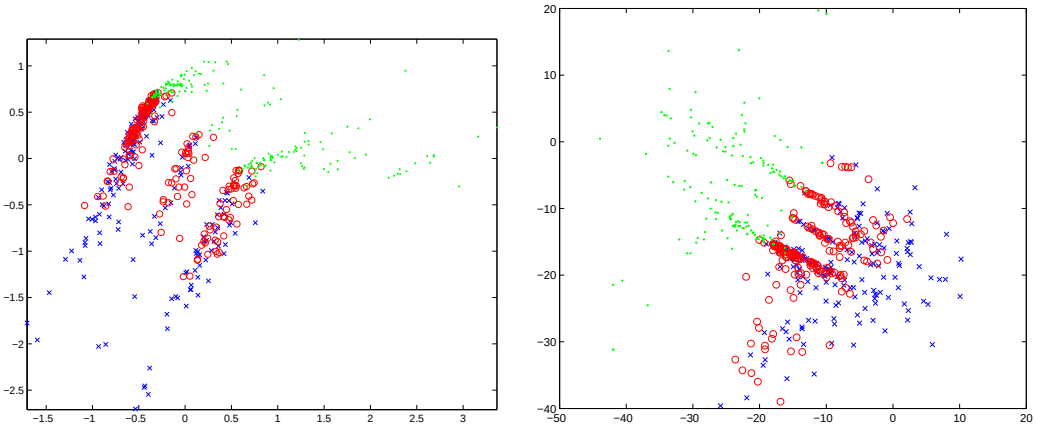


(a) $V_0 = (1, 1, 1)$, $p_0 \sim \mathcal{N}(-2, 1)$ (b) $V_0 = (1, 1, 1)$, $p_0 \sim \mathcal{N}(-2, 3)$

Figure III.4: Norm of sines of canonical angles to the correct subspace to which the distribution over 'natural parameters' of the Poisson mixture are constrained - Part 2.

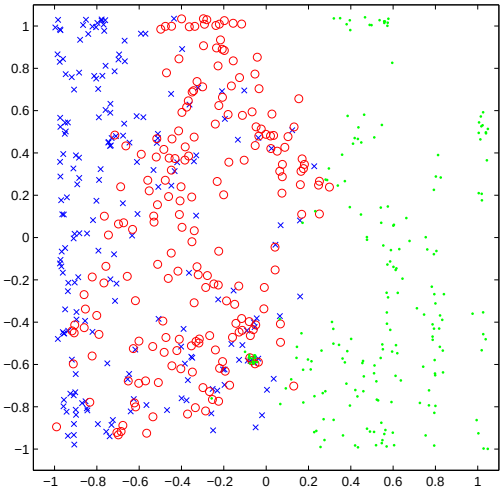
three sets reasonably well (Fig. III.5(d)). One way to quantify the separation of dissimilar groups in the two-dimensional projections is to use the training set classification error of projected data using SVM. The accuracy of the best SVM classifier (we tried a range of SVM parameter values and picked 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 comp.sys.ibm.pc.hardware and comp.sys.mac.hardware have overlap in projection using Bernoulli GTM is that the prior is assumed to be 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 distribution is allowed to adapt to the given data set. 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,

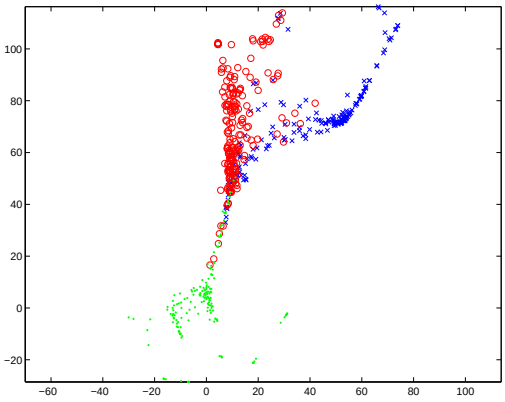


(a) PCA

(b) Exponential PCA



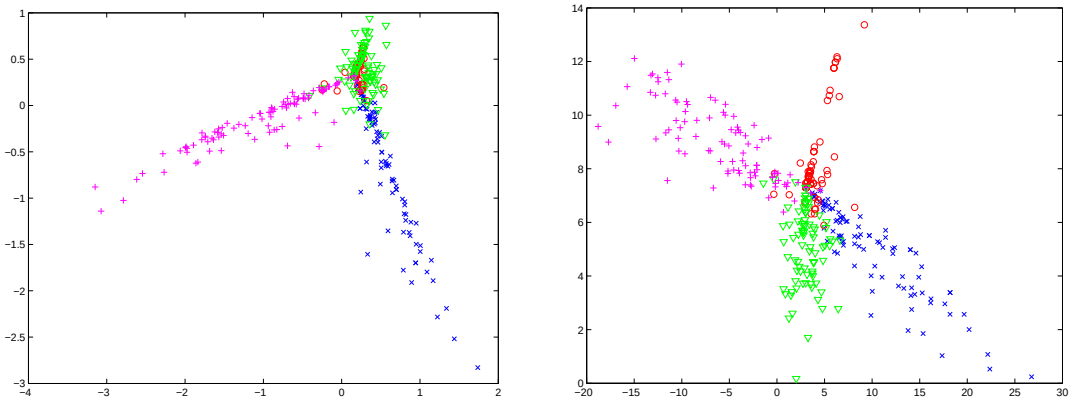
(c) GTM



(d) SP-PCA

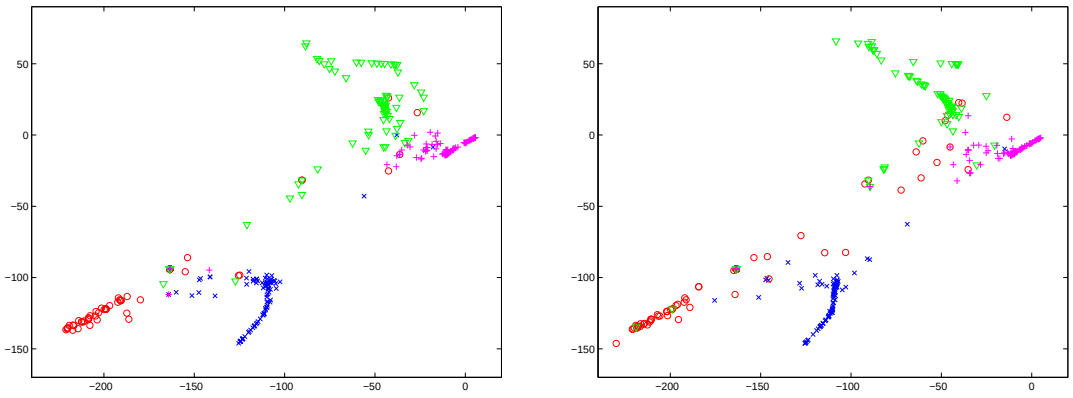
Figure III.5: Projection by various methods of binary data from 200 documents each from comp.sys.ibm.pc.hardware (×), comp.sys.mac.hardware (○) and sci.med (.)

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 and 100 more to



(a) PCA

(b) Exponential PCA

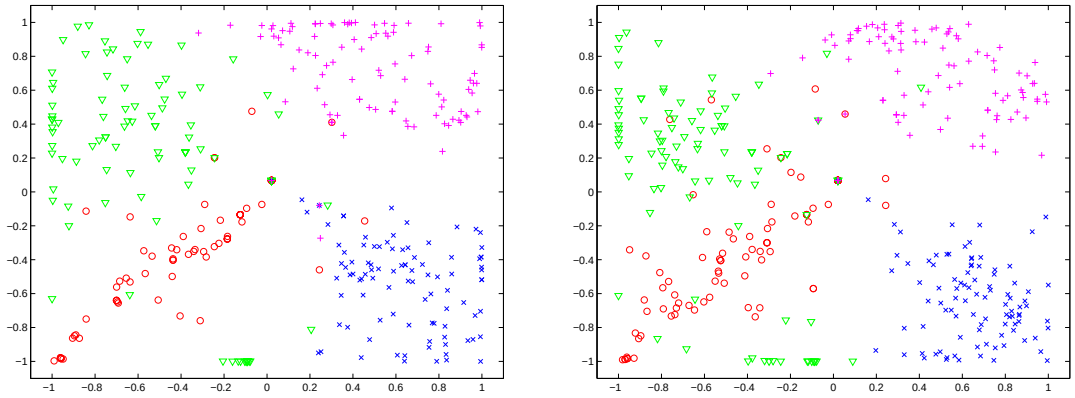


(c) SP-PCA

(d) Test data - SP-PCA

Figure III.6: 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 ($+$) - Part 1

form the test data. Figures III.6 and III.7 show two-dimensional representations of binary data obtained using various methods. Note that while the four newsgroups are bunched together in the projection obtained using Exponential family PCA [13] (Fig. III.6(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



(a) Bernoulli GTM

(b) Test data - GTM

Figure III.7: 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 ($+$) - Part 2

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 this projection is that a conjugate prior has to be used in its estimation for computational purposes [13] and the form and parameters of this prior are considered fixed and given inputs to the algorithm. Both SP-PCA (Fig. III.6(c)) and Bernoulli GTM (Fig. III.7(a)) were able to clearly separate the clusters in the training data. Figures III.6(d) and III.7(b) show representation of test data using the models estimated by SP-PCA and Bernoulli GTM respectively. To measure generalization of these methods, we can use a K-nearest neighbors based non-parametric estimate of the density of the projected training data and compare the percentage difference between the log-likelihoods of training and test data with respect to this density. SP-PCA 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 in the difference in the projections of training and test data of sci.space (∇) in Figures III.7(a) and

III.7(b).

III.H Acknowledgement

The material presented in this chapter has been published in Advances in Neural Information Processing Systems 2005. The dissertation author was the primary investigator and the first author of this paper.

Chapter IV

Supervised dimensionality reduction using mixture models (SDR-MM)

In this chapter, we present a linear, supervised dimensionality reduction method using exponential family mixture models and demonstrate using experiments that it compares favorably with other state of the art, non-parametric methods.

IV.A Motivation and Overview

We consider the problem of finding discriminative linear feature transformations. Given a collection of d-dimensional training samples and their 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 [40].

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 [51] and Generalized additive models [52]. More recently, several semi-parametric methods have been proposed for supervised dimensionality reduction including sliced inverse regression [53] and principal Hessian directions (pHd) [54]. Sufficient dimensionality reduction [55] is designed for the unsupervised case and uses maximum entropy principle for estimating the exponential models involved.

In terms of the density model used, the method most closely related to SDR-MM is Mixture discriminant analysis (MDA) [35] 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. Secondly, 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) [34] 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. A related method was proposed in [38] in which instead of using the Shannon mutual information, a Renyi-entropy based expression for mutual information is estimated.

Recently, several methods have been proposed for probabilistic formulation of principal component analysis and its extension using the exponential family of distributions (see for e.g., [50] and the references therein). 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) [50] to the supervised scenario. We describe an simple and efficient 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.

IV.B Model with Gaussian 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 $\{1, \ldots, M\}$. We are given training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, which are independent and identically distributed samples, drawn 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, $\Pi = \{\pi_1, \ldots, \pi_c\}$ be the prior over these components and for each $k \in \{1, \ldots, c\}$, 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 is given by

$$P(Y = m) = \sum_{k=1}^{c} \psi_k(m)\pi_k$$

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

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 the SDR-MM 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 SDR-MM model trained 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 $\mathbf{x}$.

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 sphericality is not a restrictive assumption follows from the universal approximation property of RBF networks with spherical gaussian kernels [48]. 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. Experimental results in section IV.G support these intuitive arguments.

The SDR-MM 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 [35]. There is a tradeoff between regularization and approximation capability - smaller variance is better for approximation and larger variance for the regularization effect described above.

IV.C 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.

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

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 [40, 39]. Hence, this objective function is equivalent to picking transformations that preserve maximum amount of the relevant information (under the SDR-MM model) between distributions of $\mathbf{x}$ and y.

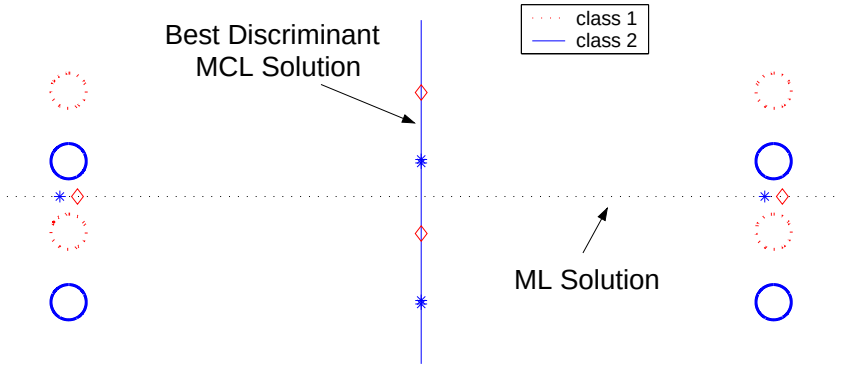


Figure IV.1: Advantage of maximum conditional likelihood : Each class is a mixture of spherical Gaussians. $\Diamond$ and $*$ denote means of gaussian components of classes 1 and 2 respectively. In this case the subspace mixture discriminant analysis finds is the same as the maximum likelihood solution.

We present simple examples of projecting two-dimensional samples onto a line to illustrate how MCL estimation extends the applicability of previously studies methods that are also, like SDR-MM, based on constrained mixture of Gaussians. Figure IV.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

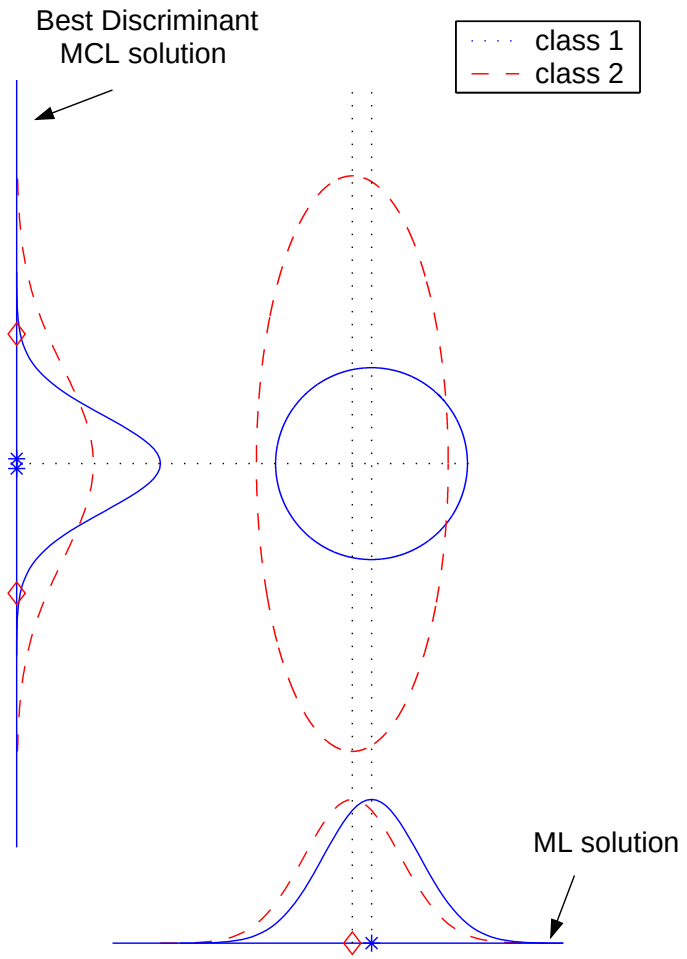


Figure IV.2: Advantage of maximum conditional likelihood : Two classes with different covariance matrices. $\Diamond$ and $*$ denote means of gaussian components of classes 1 and 2 respectively. In this case the subspace mixture discriminant analysis finds is the same as the maximum conditional likelihood solution.

the best discriminant (see also [40]). Figure IV.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 figure IV.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.

Simulation studies [39] 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 IV.B). Hence optimizing the function (IV.1) is equivalent to picking the best subspace for a discriminative Gaussian mixture classifier.

IV.D Exponential family components

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 Gaussian model may not 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* [29] 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. 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 (e.g., [50] and the references therein). Hence we extend the model described in section IV.B by using multivariate exponential family distributions for mixture components in the place of fixed variance

Gaussians,

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

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.

IV.E 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 (exponential family-based) 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 the SDR-MM 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 successfully by several probabilistic methods in the unsupervised case, to get meaningful low dimensional views.

IV.F The optimization algorithm

Several iterative algorithms have been proposed for MCL estimation of mixture models, see for example [40, 39]. 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 [40] 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

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

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

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 $i = 1,\ldots,n$ and $k = 1,\ldots,c$. 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)

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

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

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

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 $\lambda \to -\infty$, $f(\lambda) \to 0$. 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 [29], i.e., we apply the Newton-Raphson 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. Update equations for A and V are given here, and can be derived similarly for b (not included here because of space constraints). Δl depends on A, V and b only through the last two terms in equation IV.3. Hence, ignoring constants, we want to maximize

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

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 IV.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

$$[\Omega_r]_{jj'} = \{\tilde{z}_r - \sum_{i=1}^{n} \rho_i \pi_r P(\mathbf{x}_i|\boldsymbol{\theta}_r)\} \frac{\partial g(\theta_{rj})}{\partial \theta_r j} \delta(j = j') + \sum_{i=1}^{n} \rho_i \pi_r P(\mathbf{x}_i|\boldsymbol{\theta}_r)(x_{ij'} - g(\theta_{rj'}))(x_{ij} - g(\theta_{rj}))$$

and

$$[GR_r]_s = \sum_{j=1}^{d} v_{sj} \tilde{x}_{rj} - \tilde{z}_r g(\theta_{rj}) - \sum_{i=1}^{n} \rho_i \pi_r P(\mathbf{x}_i|\boldsymbol{\theta}_r)(x_{ij} - g(\theta_{rj})).$$

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,

$$[\Omega_s]_{kk} = \{\tilde{z}_k - \sum_{i=1}^{n} \rho_i \pi_k P(\mathbf{x}_i|\boldsymbol{\theta}_k)\} \frac{\partial g(\theta_{ks})}{\partial \theta_{ks}} + \sum_{i=1}^{n} \rho_i \pi_k P(\mathbf{x}_i|\boldsymbol{\theta}_k)(x_{is} - g(\theta_{ks}))^2$$

and

$$[GR_s]_r = \sum_{k'=1}^{c} a_{k'r} \{\tilde{x}_{k's} - \tilde{z}_{k'} g(\theta_{k's}) + \sum_{i=1}^{n} \rho_i \pi_{k'} P(\mathbf{x}_i|\boldsymbol{\theta}_{k'})(x_{is} - g(\theta_{k's}).\}$$

Note that using the Newton-Raphson 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.

COMPUTATIONAL COMPLEXITY : Time taken for each iteration of this algorithm is $\mathcal{O}(cdnL^2)$. Computing $\hat{z}_{ik}$ and ρ_i involve computing $P(\mathbf{x}_i|\boldsymbol{\theta}_k)$ which is

expensive and is a common problem faced in maximum likelihood estimation and in training of RBF networks. [42] 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.

IV.G 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 IV.B, 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 $P(\mathbf{x}_i|\boldsymbol{\theta}_k)$ 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 for this purpose).

IV.G.1 Classification results

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

Data set	data dimension	training set size	test set size
Heart disease	13	149	148
Ionosphere	34	151	200
Breast cancer	30	200	369
Waveform	21	300	500

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

Data set	L	pHd	KDR	MDA	SDR-MM
Heart	1	52.37	80.68	77.84	80.81
	3	68.92	77.43	77.97	80.95
	5	73.31	76.82	80.74	81.49
Ionosphere	1	68.80	90.28	75.75	87.14
	3	82.75	95.28	86.9	89.71
	5	87.65	94.88	88.85	91.14
Breast	1	73.88	93.82	92.55	95.50
	3	84.23	90.92	93.36	95.83
	5	90.41	88.59	93.88	95.85
Waveform	1	-	59.32	60.58	60.98
	2	-	82.80	84.40	85.16
	4	61.6	79.08	83.78	84.36

Table IV.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	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

We give classification results comparing SDR-MM with KDR, MDA and pHd. We modified the matlab package of Kernel ICA [33] to obtain the KDR results. The variance parameter for KDR was gradually decreased (between iterations) to two as suggested in [34]. 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 IV.1).

Table IV.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 obtained results similar to SVM using the inherent classifier, that uses the probability densities estimated in the process of finding the lower dimensional space (not shown here for lack of 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 IV.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.

IV.G.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-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. Figure IV.3 shows 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 IV.2). However, we see that the third coordinate distin-

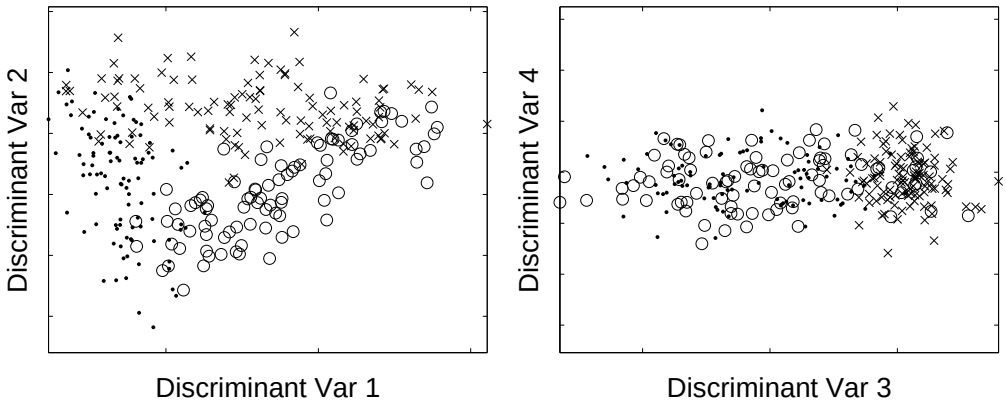


Figure IV.3: Some two dimensional views of waveform dataset projected onto the four basis vectors obtained using SDR-MM

guishes 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 (figure IV.4). In the corresponding projections obtained using MDA, shown in figure IV.5, the third and fourth discriminants do not significantly discriminate between the classes.

IV.G.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 [49] 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 and 16 binary features from this data-set.

The goal is to extract and understand features that predict whether a

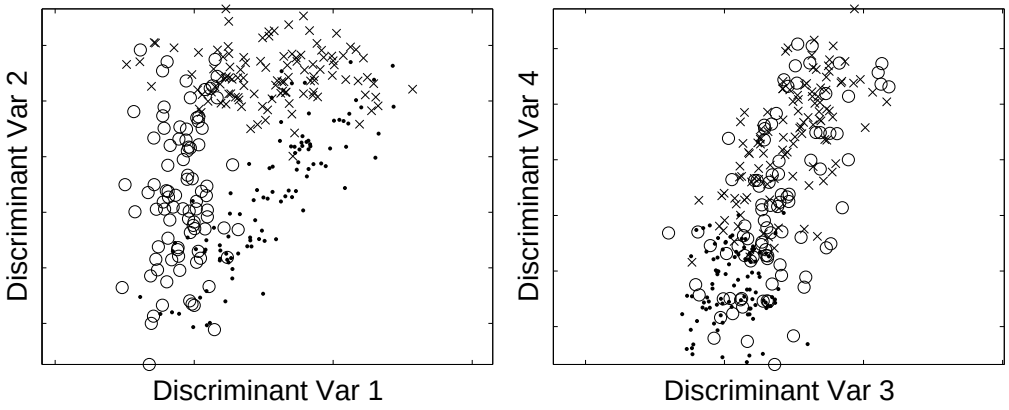


Figure IV.4: Some two dimensional views of waveform dataset projected onto the four basis vectors obtained using kernel dimensionality reduction

patient will leave the ICU alive. The features considered include presence 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. IV.6. 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.

The projected data can be visually divided into five clusters (figure IV.6). 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. 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.

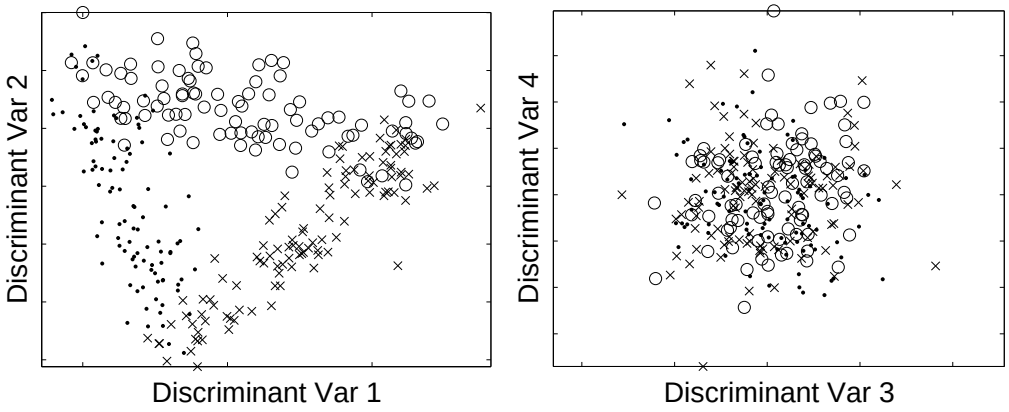


Figure IV.5: Some two dimensional views of waveform dataset projected onto the four basis vectors obtained using mixture discriminant analysis

IV.H Acknowledgement

The material presented in this chapter has been published in the Proceedings of the International Conference on Machine Learning 2005. The dissertation author was the primary investigator and the first author of this paper.

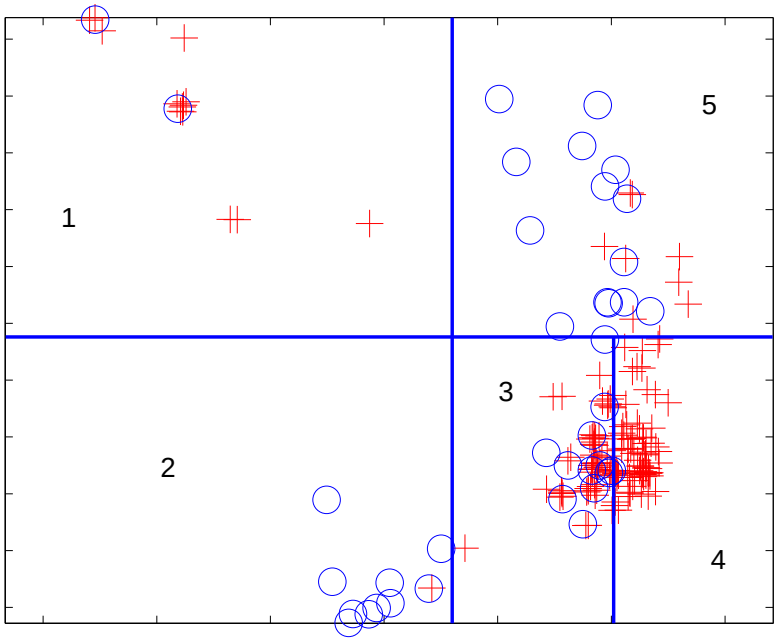


Figure IV.6: Two dimensional representation of binary data from the ICU data set : patients who left the ICU alive are shown by '+' and the patients who did not by '∘'.

Chapter V

Learning Distance metrics and its applications

In this chapter, we present a brief review of learning methods that use alternate distance metrics and methods that learn such metrics. The distance metric used to measure the similarity/dissimilarity between the data points is an essential part of machine learning algorithms. Most classification algorithms, for example, inherently use the assumption that two data points that are close to one another (according to some metric) are likely to belong to the same class. Similarly, k-means clustering assigns each data point to one of k clusters so as to minimize a measure of dispersion within the clusters and this measure of dispersion is measured in terms of distances between points.

V.A Alternative distance metrics

There are several instances when using a suitable distance measure is critical for the performance of learning algorithms. In Chapter II, we saw that several dimensionality reduction methods use a distance measure (specifically, Bregman distances) adapted to the data type, for example, a distance measure based on the Bernoulli probability model was found to be better suited for use with binary data. Another example of non-Euclidean distance metrics include use of cosine distance

(cosine of the angle between two vectors) in information retrieval for text [99] and audio [98]. Cosine distance improves performance of retrieval algorithms since it reflects similarity in terms of the relative distributions of components and is not influenced by one document being small compared to the other. Edit distance is used for measuring similarity between strings, for example genes and text data.

V.B Learning distance metrics

Over the last decade, a lot of research has been done on learning the distance or similarity measures are from data instead of assuming them apriori. Early work on learning distance metrics for classification used a modified distance metric close to the boundary of the classes where points that lie across the boundary should be considered more dissimilar than points on the same side of the boundary [101]. There has been much work on learning representations and distance functions in the supervised learning settings, and we briefly mention a few examples. [102] considers the problem of learning a Mahalanobis metric when the user provides the learning algorithm with sets of points that are similar or dissimilar. [103] optimize the metric with the goal that k-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. [104] presents a method for learning a distance metric from relative comparison such as A is closer to B than A is to C.

V.C Unsupervised learning of distance metrics and applications

Work in the area of unsupervised distance metric learning is concerned with adapting the distance to local data density. When data is available in clusters, two points in the same cluster are likely to be more similar to each other than points that belong to different clusters (see Figure V.1). Similarly, when data lies along a lower dimensional manifold, distance along the manifold is likely to be a better

measure of similarity than Euclidean distance (see Figure V.2). This notion of similarity based on data density is the subject of the second part of this thesis and has many applications including semi-supervised classification, clustering and interpolation. In the following subsections, we review literature relevant to these applications with an emphasis on methods based on this notion of similarity.

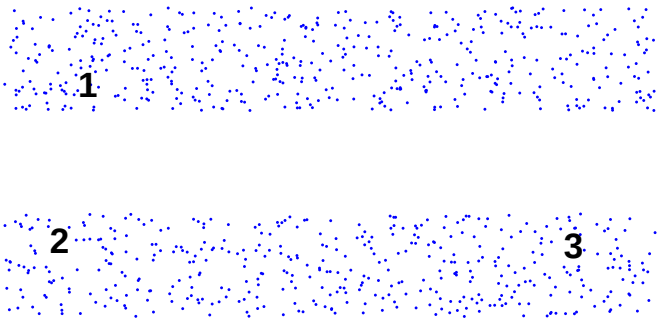


Figure V.1: Distance based on data density - the cluster case - point 2 is more similar to point 3 than to point 1

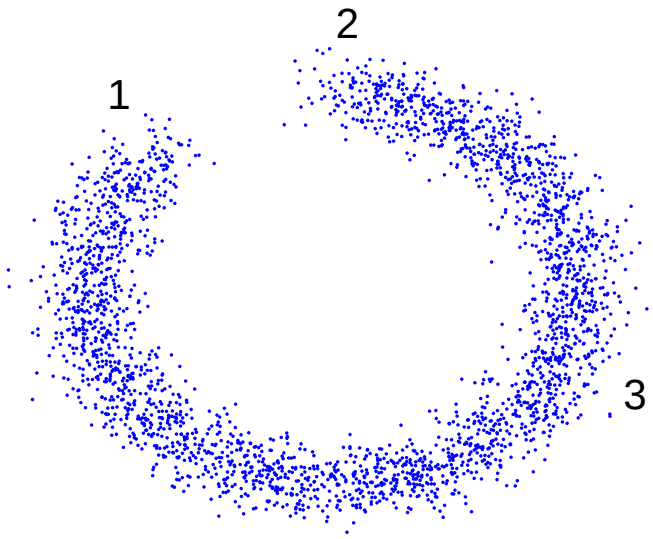


Figure V.2: Distance based on data density - the manifold case - point 2 is more similar to point 3 than to point 1

V.C.1 Semi-supervised learning

In the classical supervised learning setting, a rule for predicting the output y corresponding any given input x is learnt based on training data of the form $\{(x_i, y_i)\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y}$. Semi-supervised learning [117] deals with the problems where the labels y_i are expensive to obtain and hence only a small fraction of them are available. Expensive labeled data and inexpensive unlabeled data occurs in many important application areas including text classification, computer vision and biological research (genetic or proteomic).

The premise of semi-supervised learning is that the unlabeled samples x_i can be used in addition to the labeled samples in order to improve classification accuracy. Most semi-supervised learning algorithms use the unlabeled data to incorporate into the classification methods the prior knowledge (or assumption) of the cluster or manifold assumption regarding similarity between points as shown in Figures V.1 and V.2.

Many methods have been proposed to use the cluster assumption for semi-supervised learning and we will only mention a few examples. Some early methods were based on generative models [106, 72] which assumes that $p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$ where $p(\mathbf{x}|y)$ is an identifiable mixture distribution, for example Gaussian mixture models. The EM algorithm is then used to estimate this model where the fraction of y_i that are unavailable are treated as missing data. Most methods proposed for semi-supervised learning are discriminative. Several methods penalize changes in $p(y|\mathbf{x})$ in the regions of high $p(\mathbf{x})$ [75, 70]. [68] is a transductive method for semi-supervised learning using support vector machines. Another class of semi-supervised methods are based on regularization using graphs constructed on the data points [71, 66, 107, 108].

There is a group of methods that use a more direct approach to incorporating the cluster or manifold assumptions - they define density based distances and compute them before using these distances in various classification algorithms. [105] suggests using manifold distance for semi-supervised learning and presents

experiments with text and image data demonstrating improved classification accuracy. [100] present experiments in which using manifold distances revealed biologically relevant structures in microarray data.

In Chapter VI, we present our analysis of a density based distance that is based on a Riemannian manifold that is a function of the local data density. Several methods for semi-supervised learning work with such Riemannian metrics [74, 63, 69, 64].

There has also been work [96, 95] on density based distances that cannot be cast into the Riemannian manifold framework. These methods consider a fully connected graph constructed on the points, where the edges are weighted by the Euclidean distance between the two points (or a given dissimilarity, if the points do not belong to an Euclidean space). In [95], this definition of distance is modified ('softened') in order to avoid connection of otherwise separate clusters by single outliers. They demonstrate how this kernel could be used in transductive SVM for semi-supervised learning and present experimental results which show improvement over the standard implementation of transductive SVM.

V.C.2 Clustering

Clustering is the process of organizing objects into groups whose members are similar in some way. It is one of the most important unsupervised learning problems since it has wide ranging applications including Information retrieval, DNA analysis, marketing and insurance studies, computational linguistics and astronomical data. One of the main challenges of clustering algorithms is that their effectiveness depends critically on the definition of similarity or distance between the objects.

Standard partitioning clustering algorithms like K-means or K-medoid algorithms group together objects in such a way as to minimize intra-cluster variance which is measured on the basis of an assumed distance metric. These algorithms need the practitioner to pick the number of clusters which requires some apriori

domain knowledge and the clusters they produce are necessarily convex, which is quite restrictive.

A more recent class of clustering algorithms are density based [111, 110, 109]. The basic idea of density based partitioning algorithms is that a set of data points in Euclidean space should be divided into a set of components with two points belonging to the same component if it is possible to reach from the first point to the second one by a sufficiently small step. Density based spatial clustering of applicatons with noise (DBSCAN) [110, 109] is a very popular and successful density based clustering method and several improvements, extensions and applications have been proposed for DBSCAN in the data mining literature [112, 113, 114, 115].

Some work on density based clustering has been reported in the machine learning literature [63, 96]. [63] propose to use Riemannian metric based distances along with standard clustering algorithms for density based clustering. [96] propose using graphs constructed on the data points and using path lengths along this graph to measure distances between points. Here the length of a path is defined to be the maximum edge weight on the path and the effective density based distance between any two points is defined to be the smallest path-length among all paths connecting the two points. Using these distances, they show a robust and computationally feasible method for clustering elongated high density regions.

V.C.3 Nonlinear interpolation

Non-linear interpolation is a problem that arises in image, speech and signal processing [76]. One application is interpolating between two images from a video stream, intermittently obtained over a weak link in low-bandwidth tele-conferencing and video e-mail [116]. Such interpolation has also been used for audiovisual speech recognition [77, 76]. [77] models the space of valid lip poses within the image space and present interpolation techniques that can be used for both analysis and synthesis. [76] use a Riemannian metric induced by a model

of the data density that assigns smaller length to those paths that pass through those regions where data density is high. This Riemannian metric then assigns a length value to each path through the space (for example of images). Given any two points, the shortest path(s) according to this metric is used as an interpolant between the two points. They show that this interpolation can be done efficiently in high dimensions for Gaussian, Dirichlet and Mixture models.

Chapter VI

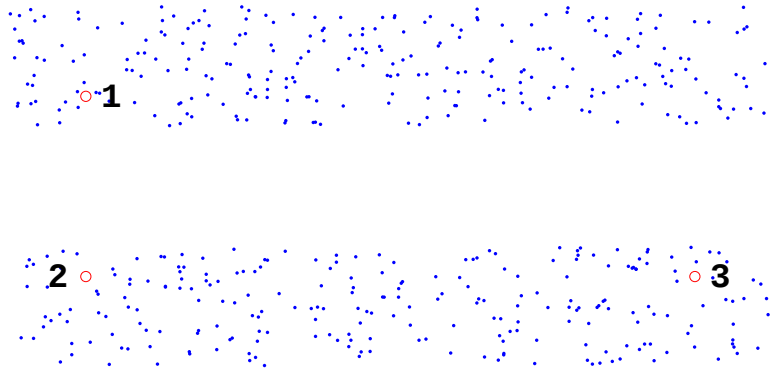
Estimating and computing density based distances

In this chapter, we analyze errors that occur while measuring a density based distance defined in terms of a minimum path length. We analyze and bound the errors that arise because of availability of finite sample and finite computational resources.

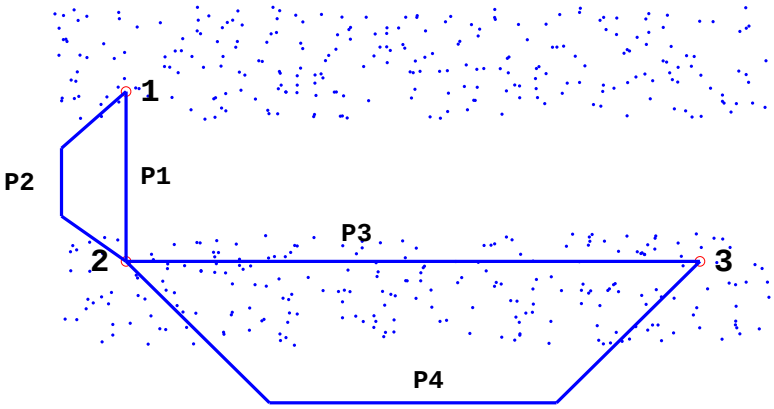
VI.A Motivation and overview

When data is in $\mathcal{R}^d$, the standard similarity measure used by learning algorithms is the Euclidean distance. Semi-supervised learning algorithms rely on the intuition that two data points are similar to each other if they are connected by a high density region. For example, based on this intuition, in the case of the two-dimensional data sample shown in Figure VI.1(a), point 2 is closer to point 3 than to point 1. In this chapter we consider measuring this density-based notion of similarity directly in the form of a distance metric between all pairs of points and then using this resulting metric in standard learning algorithms to perform semi-supervised classification.

To see how a density-based distance (DBD) metric can be defined, let us take a closer look at the two-strips example in Figure VI.1(b). Since there



(a) According to the semi-supervised smoothness assumption point 2 has greater similarity (is closer) to point 3 than to point 1



(b) This notion of similarity can be written in terms of property of paths between the points

Figure VI.1: A notion of similarity that is a function of data density

is a path between Points 2 and 3 that lies in a high density region (for example, P3), we assume them to be similar or 'closer'. Conversely, since none of the paths between points 1 and 2 (P1, P2, etc.) can avoid the low density regions, they are 'farther' according to the density based notion of distance.

This observation leads us to consider modifying the standard Euclidean definition of the length of paths and to use the shortest path length as the density-based distance metric. To make this definition work, those paths that leave the high-density regions should be assigned longer length than those that do not. Note that path length is defined as the sum of lengths of infinitesimally small path-segments. One way to define a density based path length would be to assign different lengths to path segments based on the data density at their location.

Hence, we use a modified definition of the path length Γ of a path γ in $\mathcal{X}$ which depends on the data density $p(\mathbf{x})$ and a suitably chosen *weighting* function $q : \mathcal{R}^+ \rightarrow \mathcal{R}^+$ via the relation

$$\Gamma(\gamma; p) \doteq \int_{t=0}^{LE(\gamma)} q(p(\gamma(t)))|\gamma'(t)|_2 dt$$

where $|.|_2$ is the L_2 norm on $\mathcal{R}^d$. We can assume, without loss of generality, that all paths are parametrized to have unit speed according to the standard Euclidean metric on $\mathcal{R}^d$ and hence that $LE(\gamma) =$ Euclidean length of curve γ and $|\gamma'(t)|_2 = 1$.

The DBD between two points $\mathbf{x}'$ and $\mathbf{x}''$ is defined to be

$$d(\mathbf{x}', \mathbf{x}''; p) = \inf_{\gamma}\{\Gamma(\gamma; p)\} \tag{VI.1}$$

where γ varies over the set of all paths from $\mathbf{x}'$ to $\mathbf{x}''$.

This DBD metric can be thought of as being induced by a corresponding Riemannian manifold structure. To specify a Riemannian manifold structure on $\mathcal{R}^d$ we need to specify the inner product on the space of tangent vectors at each point in $\mathcal{R}^d$. For $\mathcal{R}^d$ the tangent space at each point is just a copy of $\mathcal{R}^d$ itself. Hence the Riemannian structure at each point is determined by specifying the inner product between the d orthonormal unit vectors which span $\mathcal{R}^d$, i.e., $< \mathbf{e}_i, \mathbf{e}_j >$

$\forall\, i, j = 1, \ldots, d.$

$$< \mathbf{e}_i, \mathbf{e}_j > = \; q^2(p(\mathbf{x})) \times \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise.} \end{cases} \tag{VI.2}$$

Semi-supervised learning using density-based Riemannian metrics has been considered by [74, 63, 69, 64]. In particular, [63] proposed using $q(y) = \frac{1}{y^\alpha}$, $\exp(-\alpha y)$ and $\alpha - \log y$, where α is a parameter that controls the path-segment length in high density regions relative to the length in low-density regions. [69] proposed $q(y) = \frac{1}{\chi(y)}$ where χ is a strictly increasing function. In this chapter, following [64], we will assume that $q(y) : [0, \infty) \longrightarrow (0, \infty)$ is any monotonically decreasing, non zero function that is constant (=1 without loss of generality) for small y. The assumption that q is decreasing ensures that paths in high-density regions have smaller length and $q > 0$ ensures that paths are not assigned zero length. Assuming that $q(y)$ does not change rapidly for small y is necessary to have uniform bounds on approximation errors when using graph-based lengths to approximate path lengths. This is because the concentration of sample points in the regions with sufficiently low density (low-concentration regions change with sample size) is likely to be small. Hence using graph edges in these regions to approximate paths will lead to large approximation errors, unless q is relatively slowly changing in these regions.

Notice that all of these definitions of the Riemannian metric are non-parametric and hence the space of possible metrics is as large as the space of probability functions that we allow. A different approach was proposed by [74] who suggested picking a Riemannian metric from a parametric set of metrics based on an objective function which gives higher value to those metrics which reduce path lengths for paths passing through high density regions.

Shortest paths according to such density-based distance (DBD) metrics have been proposed for non-linear interpolation of speech and images [76, 77]. DBD metrics could also be used for clustering when the notion of clusters is of 'connected regions' of high density separated by 'boundaries' of low density [63].

[74] proposes picking a Riemannian metric from a parametric set of metrics based on an objective function which encourages metrics which reduce path lengths for paths passing through high density regions.

There has also been work on density based distances that cannot be cast into the Riemannian manifold framework ([96, 95]). These methods consider a fully connected graph constructed on the points, where the edges are weighted by the Euclidean distance between the two points (or a given dissimilarity, if the points do not belong to an Euclidean space). In [96], the length of a path is defined to be the maximum edge weight on the path and the effective density based distance between any two points is defined to be the smallest path-length among all paths connecting the two points. Using these distances, they show a robust and computationally feasible method for clustering elongated high density regions. In [95], this definition of distance is modified ('softened') in order to avoid connection of otherwise separate clusters by single outliers. They demonstrate how this kernel could be used in transductive SVM for semi-supervised learning and present experimental results which show improvement over the standard implementation of transductive SVM.

Errors in the knowledge of the DBD metric can arise from two sources, viz., estimation and computation. Estimation error arises because the underlying data density is not known a priori and the path length values need to be estimated from the finite data sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ according to the density. Even in the case when the data density is known, computing the Riemannian distance involves the variational problem of minimizing the Riemannian length over all paths between two points. Computation error arises since this minimization cannot be done perfectly when computational resources are limited.

This computation problem has been extensively studied (cf. [90]) and finds applications in computational geometry, fluid mechanics, computer vision and materials science. These methods involve building a grid in $\mathcal{R}^d$ whose size is exponential in d. This is inconvenient for the learning scenario where the data

dimension is usually high. It is therefore necessary to consider grids based on data points, in which case the computational complexity grows at a rate polynomial in sample size n. Heuristics for computing the minimum Riemannian distance using graphs constructed on data sample have been suggested by [63, 69, 64].

In the sections that follow, we present asymptotically consistent methods to estimate and compute these metrics and show bounds on the estimation and computation errors of these metrics ([64]). We also discuss the various ways in which density based metrics could be used for semi-supervised learning and present experimental results.

VI.B Estimating density based distance metrics

In this section we consider the error in our knowledge of DBD metrics that comes from the fact that we have a limited data sample, i.e., a set of d-dimensional data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ drawn *iid* from a probability density function $p(\mathbf{x})$. In other words, we are interested in the estimation of the path length function

$$\Gamma(\gamma; p) \doteq \int_{t=0}^{LE(\gamma)} q(p(\mathbf{x}))|\gamma'(t)|_2 dt$$

(see Section VI.A) for any given path γ. Note that for a fixed path γ, $\Gamma(\gamma; p)$ is a functional of the density $p(\mathbf{x})$. Several different ways of analyzing estimators of functionals of data density have been studied in the statistics literature. For bounding the error in estimating the DBD metric we borrow from the proof techniques used by [81, 80].

To characterize the estimators of the path lengths and hence the DBD metric, we use the definitions of upper and lower bounds on rate of convergence of estimators proposed by [81]. Let $\mathcal{W}$ denote the set to which p is known to belong.

Definition 3. *A convergence rate r is* achievable *if there is a sequence $\{\hat{\Gamma}_n(\gamma)\}$ of estimators such that*

$$\lim_{c\to\infty} \limsup_{n} \sup_{p\in\mathcal{W}} \Pr_{p}(|\hat{\Gamma}_n(\gamma) - \Gamma(\gamma; p)| > cn^{-r}) = 0$$

Definition 4. *A rate $r > 0$ is an* upper bound *to the rate of convergence if for every sequence $\hat{\Gamma}_n(\gamma)$ of estimators of $\Gamma(\gamma; p)$,*

$$\liminf_n \sup_{p \in \mathcal{W}} \Pr_p(|\hat{\Gamma}_n(\gamma) - \Gamma(\gamma; p)| > cn^{-r}) > 0 \qquad \forall c > 0 \tag{VI.3}$$

and

$$\lim_{c \to 0} \liminf_n \sup_{p \in \mathcal{W}} \Pr_p(|\hat{\Gamma}_n(\gamma) - \Gamma(\gamma; p)| > cn^{-r}) = 1 \tag{VI.4}$$

For statements in probability about random variables T_n, Q_n, whose distributions may depend on $p(\mathbf{x})$, we will use the notation $T_n = \mathcal{O}(Q_n)$ when $\lim_{c \to \infty} \limsup_n \sup_{f \in \mathcal{W}_s} P(|T_n| > c|Q_n|) = 0$.

VI.B.1 Achievability

We are trying to understand the limits on rate at which the estimation error can converge to zero as sample size n increases. Lower bounds on the achievable rate of convergence can be shown by considering particular estimators and analyzing their performance. This is the basic idea which leads to the first Theorem in this section where we consider the plug-in estimators, $\hat{\Gamma}_n$ for the path length Γ, i.e.,

$$\hat{\Gamma}_n(\gamma) = \Gamma(\gamma; \hat{p}_n).$$

This estimator is obtained by plugging in the kernel density estimator $\hat{p}_n$ for data density in place of actual density $p(\mathbf{x})$ into the expression for path length Γ. The kernel density estimator is given by

$$\hat{p}_n(\mathbf{x}) = \frac{1}{n\, h_n^d} \sum_{i=1}^{n} \mathcal{K}\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

where h_n is the *width* parameter of the kernel which is chosen to be a function of sample size n and $\mathcal{K}(\mathbf{x})$ is a d-dimensional kernel function. To bound how far this plug-in estimator is from the true path length, we consider the 'gradient' of the path length functional with respect to variations in density $p(\mathbf{x})$. We can then use the results on bias and variance of the kernel density estimators to derive a lower bound on the rate of convergence of the estimation error.

To define an estimator for the DBD metric between two points in the support of $p(\mathbf{x})$, we could take the shortest estimated path length among all possible paths between the points. However, this is a large space of paths that contains redundant paths like those that loop over themselves, etc. In order to prove a lower bound on the rate of convergence of the DBD metric, we consider a smaller set of paths, Sp, that nevertheless contains the shortest Riemannian paths between all pairs of points in the support of $p(\mathbf{x})$. Let the maximum Euclidean distance between two points in the support of $p(\mathbf{x})$ be L. Define,

$$Sp = \left\{ \gamma \, \hat{\Gamma}_n(\gamma) \leq L + c \right\},$$

where c is any strictly positive constant.

To see why it is sufficient to look within the set Sp, note that the straight line joining any two such points has length less than L according to this density based Riemannian metric (because we have defined the the weighting function q to be less than or equal to 1). Hence, the shortest Riemannian path between any two points will have length less than or equal to L. By the proof of Theorem 7, for sufficiently large n, all paths of length $\Gamma \leq L$ will have estimated path length $\hat{\Gamma} \leq L + c$. Hence for sufficiently large n, Sp will almost surely contain the shortest Riemannian paths between all pairs of points in the support of $p(\mathbf{x})$.

Given the estimator $\hat{\Gamma}_n$ for the lengths of paths, and the set of paths to consider, Sp, we define the estimator $\hat{d}_n(\mathbf{x}', \mathbf{x}'')$ of the DBD metric $d(\mathbf{x}', \mathbf{x}''; p(\mathbf{x}))$ to be

$$\hat{d}_n(\mathbf{x}', \mathbf{x}'') = \inf_{\gamma \in Sp} \{\hat{\Gamma}_n(\gamma)\}.$$

For proving these bounds, the function q that controls the path length is assumed to have the following properties

[A1] q is monotonically decreasing function

[A2] $\inf_y q(y) > 0$

[A3] q has bounded first and second derivatives

One feature of the kernel density estimator is that, when the true data density can be assumed to be smooth (have a certain number of derivatives), its bias can be reduced by choosing an appropriate kernel. Let us denote by $\mathcal{W}_s$, the set of functions which have s or more continuous derivatives. We assume that $p(\mathbf{x})$ has the following properties —

1. $p(\mathbf{x}) \in \mathcal{W}_s$
2. $p(\mathbf{x})$ has bounded support
3. $\exists\ C_1$ such that $|| \bigtriangledown p(\mathbf{x})|| \leq C_1 \quad \forall \mathbf{x}$

The smoothness parameter s measures the complexity of the class of underlying distributions. Given that $p(\mathbf{x})$ belongs to $\mathcal{W}_s$, we base the density estimate on the d-dimensional kernel $\mathcal{K}(\mathbf{x}) = \Pi_{j=1}^{d} k(x_j)$. Here k is a one-dimensional kernel with the following properties

$$k(x) = k(-x), \qquad \int k(x)dx = 1, \qquad \sup_{-\infty<x<\infty} |k(x)| \leq A < \infty,$$

$$\int x^m k(x)dx = 0,\ m = 1, \ldots, s-1 \qquad \text{and} \qquad 0 \neq \int x^s k(x)dx < \infty.$$

We use the following two Lemmas about well known (cf. [62]) properties of the kernel density estimators.

Lemma 5 (Bias of the kernel density estimator). *Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$ be a d-dimensional vector with $\mu_i \geq 0$, and let $\mathbf{u} = (u_1, \ldots, u_d$ denote a vector in $\mathcal{R}^d$. Let $|\boldsymbol{\mu}| = \sum_{j=1}^{d} \mu_j$, $\boldsymbol{\mu}! = \mu_1! \ldots \mu_d!$, $\mathbf{u}^{\mu} = u_1^{\mu_1} \ldots u_d^{\mu_d}$ and $D^{\boldsymbol{\mu}} = \frac{\partial^{\mu_1}}{\partial u_1^{\mu_1}} \ldots \frac{\partial^{\mu_d}}{\partial u_d^{\mu_d}}$. Then, $\forall \mathbf{x}$, the bias*

$$\mathbf{E}_p\left[\hat{p}_n(\mathbf{x})\right] - p(\mathbf{x}) = s h_n^s \int_{\mathbf{u}\in\mathcal{R}^d} F(\mathbf{u}, \mathbf{x}) \mathcal{K}(\mathbf{u}) du_1 ... du_d,$$

where

$$F(\mathbf{u}, \mathbf{x}) = \sum_{|\boldsymbol{\mu}|=s} \frac{\mathbf{u}^{\boldsymbol{\mu}}}{\boldsymbol{\mu}!} \int_{T=0}^{1} (1-T)^{s-1} D^{\boldsymbol{\mu}} p(\mathbf{x} + T\mathbf{u}) dT.$$

Lemma 6 (Variance of the kernel density estimator). $\forall \mathbf{x}$, $\forall \epsilon \geq 0$, *for* $n \geq N(\epsilon)$ *(where* $N(\epsilon)$ *is sufficiently large), the variance*

$$\mathbf{E}_p\left[(\hat{p}_n(\mathbf{x}) - \mathbf{E}_p\left[\hat{p}_n(\mathbf{x})\right])^2\right] \leq \frac{(1+\epsilon)p(\mathbf{x})}{nh_n^d}\int_{\mathbf{u}\in\mathcal{R}^d}\mathcal{K}^2(\mathbf{u})d\mathbf{u}.$$

Theorem 7 (Achievability). *Uniformly over all pairs of points* $\mathbf{x}'$ *and* $\mathbf{x}'' \in$ *the support of* $p(\mathbf{x})$*, the plug-in estimator* $\hat{d}_n(\mathbf{x}', \mathbf{x}'')$ *that uses the kernel density estimator* $\hat{p}_n$*, achieves the rate of convergence* $r = \frac{s}{2s+d}$ *where the width of the kernel density estimators* $h_n = \frac{c}{n^{\frac{1}{2s+d}}}$*, where* c *is a constant.*

Proof. We begin by defining the derivative T of the functional $\Gamma(\gamma; p)$ with respect to changes $\delta p(\mathbf{x})$ in $p(\mathbf{x})$ to be

$$T(\delta p; p) \doteq \int_{t=0}^{LE(\gamma)} q'(p(\gamma(t)))\delta p(\gamma(t))|\gamma'(t)|_2 dt$$

Hence, we can write

$$|\Gamma(\gamma;\hat{p}_n) - \Gamma(\gamma;p) - T(\hat{p}_n - p; p)| = \left|\int_{t=0}^{LE(\gamma)}\left[q(\hat{p}_n) - q(p) - (\hat{p}_n - p)q'(p)\right]|\gamma'(t)|_2 dt\right|,$$

where p and $\hat{p}_n$ are evaluated at $\gamma(t)$. By a proof similar to intermediate value theorem, we know that $q(y+\delta y) - q(y) - \delta y q'(y) = \frac{q''(\beta)}{2!}\delta y^2$ for some β in the domain of q. Hence, for some constant C,

$$|\Gamma(\gamma;\hat{p}_n) - \Gamma(\gamma;p) - T(\hat{p}_n - p; p)| \leq \mathrm{C}\int_{t=0}^{LE(\gamma)}\{\hat{p}_n(\gamma(t)) - p(\gamma(t))\}^2|\gamma'(t)|_2 dt.$$

Therefore,

$$\begin{aligned}|\Gamma(\gamma;\hat{p}_n) - \Gamma(\gamma;p)| \leq\ & |T(\hat{p}_n - \mathbf{E}_p\left[\hat{p}_n\right]; p)| + |T(\mathbf{E}_p\left[\hat{p}_n\right] - p; p)| \\ & + \left|\mathrm{C}\int_{t=0}^{LE(\gamma)}\{\hat{p}_n(\gamma(t)) - p(\gamma(t))\}^2|\gamma'(t)|_2 dt\right|.\end{aligned}$$

We now bound each of these three terms in turn. The variance of the

first term is bounded as follows

$$\begin{aligned}
&\mathbf{E}_p\left[\left(\int_t q'(p(\gamma(t)))\left\{\hat{p}_n - \mathbf{E}_p\left[\hat{p}_n\right]\right\}|\gamma'(t)|_2 dt\right)^2\right] \\
&\leq L\left(\max_\beta q'(\beta)\right)^2 \mathbf{E}_p\left[\int_t \left\{\hat{p}_n - \mathbf{E}_p\left[\hat{p}_n\right]\right\}^2 |\gamma'(t)|_2 dt\right] \\
&= L\left(\max_\beta q'(\beta)\right)^2 \int_t \mathbf{E}_p\left[\left(\hat{p}_n - \mathbf{E}_p\left[\hat{p}_n\right]\right)^2\right]|\gamma'(t)|_2 dt \\
&\leq \frac{(1+\epsilon_1)L^2}{nh_n^d}\left(\max_\beta q'(\beta)\right)^2\left(\max_{\mathbf{x}} p(\mathbf{x})\right)\int_{\mathcal{R}^d}\mathcal{K}^2(\mathbf{u})d\mathbf{u}.
\end{aligned}$$

The first inequality follows from the Cauchy-Schwarz inequality, and the second equality follows from Fubini's theorem. The third inequality is true for sufficiently large n by Lemma 6. The constant L is the maximum Euclidean length of the paths that we are considering and hence also upper bounds the length of these paths according to the density based Riemannian metric. Since the variance of $T(\hat{p}_n - E\hat{p}_n; p)$ is bounded as above for sufficiently large n, we can conclude that

$$T(\hat{p}_n - \mathbf{E}_p\left[\hat{p}_n\right]; p) = \mathcal{O}\left(\frac{1}{(nh_n^d)^{1/2}}\right).$$

The second term $T(\mathbf{E}_p\left[\hat{p}_n\right] - p; p)$ can be bounded in terms of the partial derivatives of $p(\mathbf{x})$ —

$$\begin{aligned}
&T(\mathbf{E}_p\left[\hat{p}_n\right] - p; p) = \int_t q'(p(\gamma(t)))(\mathbf{E}_p\left[\hat{p}_n\right] - p)|\gamma'(t)|_2 dt \\
&\leq (\max q'(\beta))h_n^s \int_t\left[\int_{\mathbf{u}}\left\{\sum_{|\boldsymbol{\mu}|=s}\frac{\mathbf{u}^{\boldsymbol{\mu}}}{\boldsymbol{\mu}!}\{D^{\boldsymbol{\mu}}p(\gamma(t)) + \epsilon_2\}\right\}\mathcal{K}(\mathbf{u})d\mathbf{u}\right]|\gamma'(t)|_2 dt \\
&= \mathcal{O}(h_n^s)
\end{aligned}$$

Here, we have used Lemma 5 and the inequality follows from uniform continuity of $D^{\boldsymbol{\mu}}p$ and holds for sufficiently large n.

The third term, $\frac{1}{2}\left(\max_\beta |q''(\beta)|\right)\int_t\{\hat{p}_n(\gamma(t)) - p(\gamma(t))\}^2|\gamma'(t)|_2 dt$ can be bounded by bounding the expectation of $\int_t\{\hat{p}_n(\gamma(t)) - p(\gamma(t))\}^2|\gamma'(t)|_2 dt$ and then using Markov's inequality.

$$\mathbf{E}_p\left[\int_t \{\hat{p}_n(\gamma(t)) - p(\gamma(t))\}^2 |\gamma'(t)|_2 dt\right]$$
$$= \int_t \mathbf{E}_p\left[(\hat{p}_n - f)^2\right] |\gamma'(t)|_2 dt$$
$$= \int_t (\mathbf{E}_p[\hat{p}_n] - p)^2 |\gamma'(t)|_2 dt + \int_{t=0}^{LE(\gamma)} \mathbf{E}_p\left[(\hat{p}_n - \mathbf{E}_p[\hat{p}_n])^2\right] |\gamma'(t)|_2 dt$$

Using Lemma 5, we can conclude that

$$\int_t (\mathbf{E}_p[\hat{p}_n] - p)^2 p |\gamma'(t)|_2 dt = \mathcal{O}(h_n^{2s})$$

It follows from Lemma 6 that

$$\int_t \mathbf{E}_p\left[(\hat{p}_n - \mathbf{E}_p[\hat{p}_n])^2\right] |\gamma'(t)|_2 dt = \mathcal{O}(\frac{1}{nh_n^d})$$

Collecting the three terms and assuming that $h_n = \frac{c}{n^{\frac{1}{2s+d}}}$, we conclude

$$|\Gamma(\gamma; \hat{p}_n) - \Gamma(\gamma; p)| = \mathcal{O}(\frac{1}{(nh_n^d)^{1/2}} + h_n^s + \frac{1}{nh_n^d} + h_n^{2s}) = \mathcal{O}(\frac{1}{n^{\frac{s}{2s+d}}}).$$

□

VI.B.2 Upper bound

An upper bound on the rate of convergence, is a reflection of the inherent difficulty of our estimation problem, since it states that you cannot do better than this limit no matter what estimator you may come up with in the future. In the second Theorem in this section, we show an upper bound by showing the existence of a density function $p_0(\mathbf{x})$ and a sequence of densities $\{p_n(\mathbf{x}), n \in \mathbb{N}\}$ with two opposing properties that hold at the same time. The first property is that $p_n(\mathbf{x})$ and $p_0(\mathbf{x})$ are close enough that they cannot be distinguished from one another on the basis of n samples and the second is that $p_n(\mathbf{x})$ and $p_0(\mathbf{x})$ are far enough away from one another that the DBD metric between two fixed points according to the two densities goes to zero slower than the rate given by the upper bound.

Theorem 8 (Upper bound). *No estimator of the DBD metric can converge at a rate faster than* $r = \frac{1}{2}$.

Proof. To prove this result, we show that there is a density function $p(\mathbf{x})$ and a shortest path between two points γ for which $\hat{\Gamma}(\gamma)$ cannot converge to $\Gamma(\gamma; p)$ faster than the rate r, irrespective of which estimator is used to obtain $\hat{\Gamma}(\gamma)$. The technique, termed 'the classification argument' was used by [81].

Consider a density function $p_0(\mathbf{x})$ with the property that the set $\{\mathbf{x} : p_0(\mathbf{x}) > \alpha\}$ contains an open ball in $\mathcal{R}^d$ over which $p_0(\mathbf{x})$ is constant. Let γ be any line segment contained in this open ball, let $\mathbf{x}_m$ be any point in the relative interior of γ and let $\mathbf{x}_0$ be any point in the ball which does not lie on the path γ. Since $p_0(\mathbf{x})$ is constant over the ball, any line segment including γ is the shortest path between its two end points. Let ψ be a non-negative, infinitely differentiable C^∞ function with compact support (for an example called 'the blimp' see [89]). Define

$$w_n(\mathbf{x}) \doteq \delta N n^{-\frac{1}{2}} \left\{\psi(\mathbf{x} - \mathbf{x}_m) - b_n \psi(\mathbf{x} - \mathbf{x}_0)\right\}.$$

Here, b_n is chosen such that $\int w_n p_0 d\mathbf{x} = 0$. We define a sequence of densities $p_n = p_0(1 + w_n)$. From the assumption [Ag1] that q is a monotonically decreasing function and from the definition of p_n, it follows that the straight line γ is the shortest path between its end points under the Riemannian metric specified by $p_n \; \forall \; n$. Since b_n is a constant, it remains bounded as $n \longrightarrow \infty$.

Now by the classification argument of [81] (details are given below), to prove our result it is sufficient to show the following two inequalities,

$$\limsup_n \; n \, \mathbf{E}_{p_0} \left[w_n^2(X)\right] < \infty \tag{VI.5}$$

$$\frac{\Gamma(\gamma; p_n) - \Gamma(\gamma; p_0)}{2} \geq C\delta N \left(n^{-\frac{1}{2}}\right), \tag{VI.6}$$

where C is some positive constant.

$$n\mathbf{E}_{p_0} \left[w_n^2(X)\right] = \frac{n\delta^2 N^2}{n} \int p_0(\mathbf{x}) \left\{\psi(\mathbf{y}) - b_n \psi(\mathbf{y} + (\mathbf{x}_p - \mathbf{x}_0))\right\}^2 d\mathbf{x} < \infty$$

For sufficiently large n, we have $\Gamma(\gamma; p_n) - \Gamma(\gamma; p_0) = T(p_n - p_0; p_0) + \mathcal{O}(\int_{t=0}^{LE(\gamma)} (p_n - p_0)^2 |\gamma'(t)|_2 dt) \geq (\delta N n^{-\frac{1}{2}})$, since $\int_{t=0}^{LE(\gamma)} (p_n - p_0)^2 |\gamma'(t)|_2 dt = \mathcal{O}(n^{-1})$.

Now, we show that using Equations VI.5 and VI.6, we can prove the two conditions (Equations VI.3 and VI.4) needed to show that 1/2 is an upper bound on the rate of convergence. Note that this part of the proof follows closely the proof in [81] and we are restating it here in detail for completeness. Let μ_n and ν_n denote the joint distribution of the iid random variables $\mathcal{X}_1, \ldots, \mathcal{X}_n$ under density functions p_0 and p_n respectively. Let L_n denote the Radon-Nikodym derivative $d\nu_n / d\mu_n$ and set $l_n = \log_e L_n$.

$$l_n = \sum_{i=1}^{n} \log(1 + w_n(\mathcal{X}_i))$$

Using the Taylor expansion $\log(1+z) = z - \frac{z^2}{2} + \frac{z^3}{3} - + \ldots$ and the fact that $|w_n(\mathbf{x})| \leq 0.5$ for n sufficiently large,

$$\left| l_n - \sum_{i=1}^{n} w_n(\mathcal{X}_i) \right| \leq \sum_{i=1}^{n} w_n^2(\mathcal{X}_i) \Rightarrow |l_n| \leq \left| \sum_{i=1}^{n} w_n(\mathcal{X}_i) \right| + \sum_{i=1}^{n} w_n^2(\mathcal{X}_i)$$

Now, since we choose b_n such that $\mathbf{E}_{p_0}[w_n(\mathcal{X})] = 0$,

$$\mathbf{E}_{p_0}\left[(\sum_{i=1}^{n} w_n(\mathcal{X}_i))^2 \right] = n \mathbf{E}_{p_0}\left[w_n^2(\mathcal{X}) \right]$$

By Schwarz's inequality

$$\left(\mathbf{E}_{p_0}\left[\sum_{i=1}^{n} w_n(\mathcal{X}_i) \right] \right)^2 \leq \mathbf{E}_{p_0}\left[\left(\sum_{i=1}^{n} w_n(\mathcal{X}_i) \right)^2 \right] = \left(n \mathbf{E}_{p_0}\left[w_n^2(\mathcal{X}) \right] \right)^{\frac{1}{2}}$$

Hence $\mathbf{E}_{p_0}[|l_n|] \leq (n \mathbf{E}_{p_0}[w_n^2(\mathcal{X})])^{\frac{1}{2}} + n \mathbf{E}_{p_0}[w_n^2(\mathcal{X})]$. This combined with Eqn VI.5 yields

$$\limsup_n \mathbf{E}_{p_0}[|l_n|] < \infty \qquad \text{and} \qquad \lim_{\delta \longrightarrow 0} \limsup_n \mathbf{E}_{p_0}[|l_n|] = 0 \tag{VI.7}$$

Hence, there is a finite, positive M such that $\limsup_n \mathbf{E}_{p_0}[|\log L_n|] < M$. Choose $\epsilon > 0$ such that if $L_n > (1-\epsilon)/\epsilon$ or $L_n < \epsilon/(1+\epsilon)$, then $|\log L_n| \geq 2M$.

By the Markov inequality

$$\liminf_n \ \mu_n \left(\frac{\epsilon}{1-\epsilon} \leq L_n \leq \frac{1-\epsilon}{\epsilon} \right) > \frac{1}{2} \, .$$

Let n be sufficiently large so that

$$\mu_n \left(\frac{\epsilon}{1-\epsilon} \leq L_n \leq \frac{1-\epsilon}{\epsilon} \right) > \frac{1}{2} \, .$$

Put prior probabilities $1/2$ each on p_0 and p_n. Then

$$P\left\{p = p_n | \mathcal{X}_1, \ldots, \mathcal{X}_n\right\} = \frac{L_n/2}{L_n/2 + 1/2} = \frac{L_n}{L+n+1}$$

and hence

$$\begin{aligned} P\left\{\epsilon \leq P\left\{p = p_n | \mathcal{X}_1, \ldots, \mathcal{X}_n\right\} \leq 1-\epsilon\right\} \\ &= P\left\{\epsilon \leq \frac{L_n}{L+n+1} \leq 1-\epsilon\right\} = P\left\{\frac{\epsilon}{1-\epsilon} \leq L_n \leq \frac{1-\epsilon}{\epsilon}\right\} \\ &\geq \frac{1}{2}\mu_n \left(\frac{\epsilon}{1-\epsilon} \leq L_n \leq \frac{1-\epsilon}{\epsilon} \right) \geq \frac{1}{4} \end{aligned}$$

Therefore any method of deciding between p_0 and p_n based on $\mathcal{X}_1, \ldots, \mathcal{X}_n$ must have overall error probability at least $\epsilon/4$. Apply this result to the classifier $\bar{p}_n$ defined by

$$\bar{p}_n = \begin{cases} p_0 & \text{if} \quad \hat{\Gamma}_n(\gamma) \leq \frac{\Gamma(\gamma;p_0)+\Gamma(\gamma;p_n)}{2} \\ 0 & \text{otherwise} \end{cases} ,$$

It follows that

$$\begin{aligned} \frac{1}{2} P_{p_0} \left(|\hat{\Gamma}_n(\gamma) - \Gamma(\gamma;p_0)| \geq \frac{\Gamma(\gamma;p_n) - \Gamma(\gamma;p_0)}{2} \right) \\ + \frac{1}{2} P_{p_0} \left\{ |\hat{\Gamma}_n(\gamma) - \Gamma(\gamma;p_n)| \geq \frac{\Gamma(\gamma;p_n) - \Gamma(\gamma;p_0)}{2} \right\} \geq \frac{\epsilon}{4} \end{aligned}$$

consequently,

$$\sup_{p \in \mathcal{W}_s} P_p \left\{ |\hat{\Gamma}_n(\gamma) - \Gamma(\gamma;p)| \geq \frac{\Gamma(\gamma;p_n) - \Gamma(\gamma;p_0)}{2} \right\} \geq \frac{\epsilon}{4} \, .$$

and hence

$$\liminf_n \sup_{p \in \mathcal{W}_s} P_p \left\{ |\hat{\Gamma}_n(\gamma) - \Gamma(\gamma;p)| \geq \frac{\Gamma(\gamma;p_n) - \Gamma(\gamma;p_0)}{2} \right\} > 0.$$

This along with Equation VI.6 proves the first requirement (Equation VI.3) for 1/2 to be an upper bound on the rate of convergence .

To prove the second part of the upper bound definition, we choose a positive integer $i_o \geq 2$ and put prior probability i_o^{-1} on each of the i_o points

$$p_{ni} = p_0 + \frac{i-1}{i_o - 1}(p_n - p_0)$$

Now, $\exists \delta > 0$ such that for sufficiently large n, any method of classifying $p \in \{p_{n1}, \ldots, p_{ni_o}\}$ based on $\mathcal{X}_1, \ldots, \mathcal{X}_n$ must have overall probability of error $1 - 2/i_o$. This is because

$$P\{p = p_{ni} | \mathcal{X}_1, \ldots, \mathcal{X}_n\} = \frac{1 + \left(\frac{i-1}{i_o - 1}\right)(L_n - 1)}{\left(\frac{L_n + 1}{2}\right) i_o}$$

and the optimum classifier to choose between p_{ni} is

$$\bar{p}_n = \begin{cases} p_0 & \text{if} \quad L_n < 1 \\ p_n & \text{otherwise} \end{cases}$$

which produces an error whenever one of $p_{n2}, \ldots, p_{n(i_o - 1)}$ are chosen in the random draw among the p_{ni}.

Note that

$$(p_{ni} + p_{n(i+1)}) - (p_{n(i-1)} + p_{n(i)}) = \left(p_0 + \frac{1}{2(i_o - 1)}(p_n - p_0)\right)$$

So, considering the classifier

$$\hat{p} = \left\{ p_{ni} \quad \text{if} \quad |\hat{\Gamma}_n - \Gamma(\gamma; p_{ni})| \leq \frac{|\Gamma(\gamma; p_n) - \Gamma(\gamma; p_0)|}{2(i_o - 1)} \right\},$$

we get

$$\sum_i \frac{1}{i_o} P_{p_{ni}} \left\{ |\hat{\Gamma}_n - \Gamma(\gamma; p_{ni})| \geq \frac{|\Gamma(\gamma; p_n) - \Gamma(\gamma; p_0)|}{2(i_o - 1)} \right\} \geq 1 - \frac{2}{i_o}$$

Consequently,

$$\sup_p P_p \left\{ |\hat{\Gamma}_n - \Gamma(\gamma; p)| \geq \frac{|\Gamma(\gamma; p_n) - \Gamma(\gamma; p_0)|}{2(i_o - 1)} \right\} \geq 1 - \frac{2}{i_o}$$

$$\lim_{i_o \longrightarrow \infty} \liminf_{n} \sup_{p \in \mathcal{W}_s} P_p \left\{ |\hat{\Gamma}_n - \Gamma(\gamma; p)| \geq \frac{|\Gamma(\gamma; p_n) - \Gamma(\gamma; p_0)|}{2(i_o - 1)} \right\} = 1.$$

This along with Equation VI.6 proves the second requirement (Equation VI.4) for 1/2 to be an upper bound on the rate of convergence .

□

VI.C Computing density based distance metrics

In Section VI.B, we analyzed the effect of using an estimate of the density function in place of the density function itself. However, even if the density were known, computing the Riemannian metric between two points is not an easy task. This is a variational minimization problem since the distance is defined as the infimum of path lengths over all paths joining the points (Equation VI.1). Isomap ([79, 78]) uses paths along a neighborhood graph to approximate paths along a manifold embedded in $\mathcal{R}^d$. [63, 69, 95] propose graph based methods to compute density based metrics for use in semi-supervised learning. However, these heuristics for approximating DBD metrics are not guaranteed to lead to a consistent distance measure, i.e., they do not guarantee convergence of the graph shortest path length to the Riemannian metric with increasing sample size. In this section we present upper and lower bounds on the rate at which approximation error can converge to zero when a particular graph construction is used for computing the Riemannian metric.

VI.C.1 Achievability

We show that the rate $1/2d$ is achievable, i.e., we present a graph construction method which produces graphs such that with high probability the difference between the shortest distance along the graph and the DBD metric is smaller than $c/n^{1/2d}$, for some constant c and for large enough n. In the proof, we use some techniques from [79, 78].

We first describe the method for constructing the graph and assigning

weights to the graph edges. In addition to the three assumptions made about the weighting function q in Section VI.B, we assume that

[A4] $\quad q(y) = 1 \quad \forall\, y \leq \alpha.$

Note that this is not overly restrictive since we can choose α to be small. As discussed in Section VI.A, it is necessary to assume that $q(y)$ does not change rapidly for small y in order to have uniform bounds on approximation errors when using graph-based lengths to approximate path lengths. Let $C_p(\alpha) \doteq \{\mathbf{x} : \hat{p}(\mathbf{x}) \geq \alpha\}$ and let $C_p(\alpha;\epsilon) \doteq \bigcup_{\mathbf{x}\in C_p(\alpha)} B(\mathbf{x},\epsilon)$ where $B(\mathbf{x},\epsilon)$ is a d-dimensional ball of radius ϵ centered at $\mathbf{x}$.

A point $\mathbf{x} \in \mathcal{R}^d$ is *high density* if $\mathbf{x} \in C_p(\alpha;\epsilon)$. A maximal connected set of high-density points is a *high-density component*. Since the density $p(\mathbf{x})$ has bounded support and integrates to one, it can be shown that there will be only finitely many high-density components and hence $C_p(\alpha;\epsilon)$ will be partitioned into finitely many high-density components $R_1, \ldots, R_k$. Note that these are high density components with respect to the estimated distribution $\hat{p}(\mathbf{x})$ and not the 'true' distribution $p(\mathbf{x})$. $C_p(\alpha;\epsilon)$ is being defined as a way to mollify the difficult properties of $C_p(\alpha)$ which can have complex boundaries (e.g., dendrils defined in [65]) and can have an infinite number of disjoint, maximally connected components.

The graph g is defined as follows. Its vertices are the observed data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Two nodes $\mathbf{x}_i$, $\mathbf{x}_j$ are connected if at least one of the following holds:

1. The Euclidean distance between two nodes is at most ϵ. The weight of such an edge is $w(\mathbf{x}_i, \mathbf{x}_j) = q(p((\mathbf{x}_i + \mathbf{x}_j)/2))|\mathbf{x}_i - \mathbf{x}_j|_2$.

2. At most one of the nodes is high-density, they are at least ϵ apart and the straight line joining the two nodes leaves $C_p(\alpha;\epsilon)$. The weight of such an edge is $w(\mathbf{x}_i, \mathbf{x}_j) = |\mathbf{x}_i - \mathbf{x}_j|_2$.

We use three distance metrics between data points $\mathbf{x}$ and $\mathbf{y}$, namely, the

DBD metric

$$d_M(\mathbf{x},\mathbf{y}) \doteq d(\mathbf{x},\mathbf{y};\hat{p}_n) = \inf_{\gamma}\{\Gamma(\gamma;\hat{p}_n)\},$$

the graph distance

$$d_{\mathsf{g}}(\mathbf{x},\mathbf{y}) \doteq \min_{P}\left(w(\mathbf{x}_0,\mathbf{x}_1)+\ldots+w(\mathbf{x}_{m-1},\mathbf{x}_m)\right),$$

and an intermediate distance

$$d_S(\mathbf{x},\mathbf{y}) \doteq \min_{P}\left(d_M(\mathbf{x}_0,\mathbf{x}_1)+\ldots+d_M(\mathbf{x}_{m-1},\mathbf{x}_m)\right)$$

where $P=(\mathbf{x}_0,\ldots,\mathbf{x}_m)$ varies over all paths along the edges of g connecting $\mathbf{x}=\mathbf{x}_0$ to $\mathbf{y}=\mathbf{x}_m$.

To lower bound the rate of convergence of the shortest path along graph g to the DBD metric, we bound the difference between the graph distance and DBD metric in Theorem 12. For this purpose we show the DBD metric and the intermediate distance are close to each other in Lemma 9. Lemmas 10 and 11 state that the graph and intermediate distances are close.

Lemma 9 (Difference between DBD metric and intermediate distance). *If* $\forall \mathbf{x}\in C_p(\alpha;2\epsilon)$ $\exists$ *some data point* $\mathbf{x}_i$ *for which* $d_M(\mathbf{x},\mathbf{x}_i)\le\delta$ *and if* $4\delta<\epsilon$, *then* $\forall$ *pairs of data points* $\mathbf{x}$ *and* $\mathbf{y}$,

$$d_M(\mathbf{x},\mathbf{y}) \le d_S(\mathbf{x},\mathbf{y}) \le \left(1+\frac{6\delta}{\epsilon}+\frac{8\delta^2}{\epsilon^2}\right) d_M(\mathbf{x},\mathbf{y})$$

Proof. The first inequality $d_M(\mathbf{x},\mathbf{y})\le d_S(\mathbf{x},\mathbf{y})$ is true by the definition of d_M and d_S. Let γ be any piecewise-smooth path connecting $\mathbf{x}$ to $\mathbf{y}$ with length l. If we are able to find a path from $\mathbf{x}$ to $\mathbf{y}$ along edges of g whose length $d_M(\mathbf{x}_0,\mathbf{x}_1)+\ldots+d_M(\mathbf{x}_{m-1},\mathbf{x}_m)$ is less than $\left(1+\frac{6\delta}{\epsilon}+\frac{8\delta^2}{\epsilon^2}\right)d_M(\mathbf{x},\mathbf{y})$, then the right hand inequality would follow by taking infimum over γ.

Note that it is sufficient to consider only those γ for which contiguous segments outside $C_p(\alpha;\epsilon)$ are straight lines. This is because, given any γ without

this property, we can define a path γ' such that the length of γ' is less than the length of γ by just replacing wiggly segment of γ outside $C_p(\alpha;\epsilon)$ by straight lines (recall that the density-based Riemannian metric has been defined to be constant Euclidean in the region outside $C_p(\alpha)$). We consider different cases based on the regions the path γ passes through.

Case (a) : *γ is wholly contained in one of the sub-regions R_k of $C_p(\alpha)$.* We use an argument similar to the one used in Isomap ([79, 78]). If $l \leq \epsilon - 2\delta$, then $\mathbf{x}$,$\mathbf{y}$ are connected by an edge which we can use as the path through the graph. If $l > \epsilon - 2\delta$, we write $l = l_0 + (l_1 + l_1 + \ldots + l_1) + l_0$ where $l_1 = \epsilon - 2\delta$ and $(\epsilon - 2\delta)/2 \leq l_0 \leq \epsilon - 2\delta$. Now, cut up the arc γ into pieces in accordance with this decomposition giving a sequence of points $r_0 = \mathbf{x}, r_1, \ldots, r_p = \mathbf{y}$, where each point r_i lies within a distance δ of a sample point $\mathbf{x}_i$. Using this construction, we can write

$$d_M(\mathbf{x}_i, \mathbf{x}_{i+1}) \leq d_M(\mathbf{x}_i, r_i) + d_M(r_i, r_{i+1}) + d_M(r_{i+1}, \mathbf{x}_{i+1}) \leq \frac{l_1 \epsilon}{\epsilon - 2\delta}.$$

Similarly,

$$d_M(\mathbf{x}, \mathbf{x}_1) \leq l_0 \frac{\epsilon}{\epsilon - 2\delta} \quad \& \quad d_M(\mathbf{x}_{p-1}, \mathbf{y}) \leq l_0 \frac{\epsilon}{\epsilon - 2\delta}$$

Since $l_0 \frac{\epsilon}{\epsilon - 2\delta} \leq \epsilon$, we find that each edge has manifold length $\leq \epsilon$ and hence belongs to g. Hence,

$$d_S(\mathbf{x}, \mathbf{y}) \leq l \frac{\epsilon}{\epsilon - 2\delta} < l \left(1 + \frac{4\delta}{\epsilon}\right)$$

Case (b) : *All segments of γ that lie outside $C_p(\alpha;\epsilon)$ have length $\geq \epsilon - 2\delta$.* We consider the case when both the initial and final points, $\mathbf{x}$ and $\mathbf{y}$ lie in $C_p(\alpha;\epsilon)$. The case when one or both of the end-points lies outside can be similarly handled. We divide the path γ into $2k+1$ sections, where k is the number of times γ goes outside $C_p(\alpha;\epsilon)$ i.e., — $\mathbf{x} \ldots r_{o1} \ldots r_{m1} \ldots r_{o2} \ldots r_{m2} \ldots r_{ok} \ldots r_{mk} \ldots \mathbf{y}$ where the sections $r_{oi} - - r_{mi}$ lie outside $C_p(\alpha)$. The d_S and d_M lengths of the interior segments

are related exactly as in Case (a) and hence we can write

$$d_S(\mathbf{x},\mathbf{y}) \leq \frac{\epsilon}{\epsilon-2\delta}\left\{d_M(\mathbf{x},r_{o1}) + d_M(r_{m1},r_{o2}) + \ldots + d_M(r_{mk},\mathbf{y})\right\} + \left\{2\delta + d_M(r_{o1},r_{m1})\right\} + \ldots + \left\{2\delta + d_M(r_{ok},r_{mk})\right\}.$$

Since each outside segment has a minimum length $\epsilon - 2\delta$, $d_M(\mathbf{x},\mathbf{y}) \geq (\epsilon-2\delta)k$. Hence $2\delta k \leq 2\delta/(\epsilon-2\delta)d_M(\mathbf{x},\mathbf{y})$ and

$$d_S(\mathbf{x},\mathbf{y}) \leq \left(1 + \frac{6\delta}{\epsilon} + \frac{8\delta^2}{\epsilon^2}\right) d_M(\mathbf{x},\mathbf{y}).$$

□

Lemma 10 (Difference between intermediate and graph distances - 1). *For all pairs of data points* $\mathbf{x}_i$,$\mathbf{x}_j$ *connected by an edge in* $\mathtt{g}$ *with* $|\mathbf{x}_i - \mathbf{x}_j|_2 \leq \epsilon$,

$$(1-\lambda_1)d_{\mathtt{g}}(\mathbf{x}_i,\mathbf{x}_j) \leq d_S(\mathbf{x}_i,\mathbf{x}_j) \leq (1+\lambda_1)d_{\mathtt{g}}(\mathbf{x}_i,\mathbf{x}_j)$$

where

$$\lambda_1 = 2\frac{\max|\bigtriangledown_{\mathbf{x}} q(p(\mathbf{x}))|_2\epsilon}{\min_{\mathbf{x}} q(p(\mathbf{x}))}$$

Proof. Let $\epsilon_2 = d_M(\mathbf{x}_i,\mathbf{x}_j)/2$ and let $B(\text{line}(\mathbf{x}_i,\mathbf{x}_j),\epsilon_2) = \bigcup_{\mathbf{x}\in\text{line}(\mathbf{x}_i,\mathbf{x}_j)} B(\mathbf{x},\epsilon_2)$.

$$R_{\min} = \min_{\mathbf{x}\in B(\text{line}(\mathbf{x}_i,\mathbf{x}_j),\epsilon_2)} q(p(\mathbf{x}))) \qquad R_{\max} = \min_{\mathbf{x}\in\text{line}(\mathbf{x}_i,\mathbf{x}_j)} q(p(\mathbf{x}))$$

Now,

$$R_{\min}|\mathbf{x}_i - \mathbf{x}_j|_2 \leq d_M(\mathbf{x}_i,\mathbf{x}_j) \leq R_{\max}|\mathbf{x}_i - \mathbf{x}_j|_2$$

and

$$d_{\mathtt{g}}(\mathbf{x}_i,\mathbf{x}_j) = |\mathbf{x}_i - \mathbf{x}_j|_2 q\left(p\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}\right)\right).$$

We use the fact that the gradient of q is bounded, we can write

$$R_{\max} \leq (1+\lambda_1)q\left(p\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}\right)\right) \qquad \forall\lambda_1 > 2\frac{\max|\bigtriangledown_{\mathbf{x}} q(p(\mathbf{x}))|_2\epsilon}{\min_{\mathbf{x}} q(p(\mathbf{x}))}$$

Hence,

$$(1-\lambda_1)d_{\mathtt{g}}(\mathbf{x}_i,\mathbf{x}_j) \leq d_M(\mathbf{x}_i,\mathbf{x}_j) \leq (1+\lambda_1)d_{\mathtt{g}}(\mathbf{x}_i,\mathbf{x}_j)$$

□

Lemma 11 (Difference between intermediate and graph distances - 2). *For all pairs of data points* $\mathbf{x}_i$,$\mathbf{x}_j$ *connected by an edge in* $\mathbf{g}$ *with* $|\mathbf{x}_i - \mathbf{x}_j|_2 > \epsilon$,

$$(1 - \lambda_2) d_{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_j) \leq d_S(\mathbf{x}_i, \mathbf{x}_j) \leq (1 + \lambda_2) d_{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_j)$$

where

$$\lambda_2 = \frac{2\delta^2 \max |\bigtriangledown\, q(p(\mathbf{x}))|_2}{\epsilon}$$

Proof. Since $q \leq 1$, $d_M(\mathbf{x}_i, \mathbf{x}_j) \leq |\mathbf{x}_i - \mathbf{x}_j|_2$. Among the exterior edges, we only need to consider those between nodes which are within δ of the boundary of $C_p(\alpha)$ or outside $C_p(\alpha)$. This is because of the way we approximate paths which leave $C_p(\alpha)$ in Theorem 9.

$$R_{\min} \geq 1 - \max_{\mathbf{x}} |\bigtriangledown\, q(p(\mathbf{x}))|_2 \delta$$

Since for exterior edges $d_{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_j) = |\mathbf{x}_i - \mathbf{x}_j|_2$, we can write

$$\begin{aligned} d_M(\mathbf{x}_i, \mathbf{x}_j) &\geq 2\delta \left(1 - \max |\bigtriangledown\, q(p(\mathbf{x}))|_2 \delta\right) + |\mathbf{x}_i - \mathbf{x}_j|_2 - 2\delta \\ &\geq |\mathbf{x}_i - \mathbf{x}_j|_2 \left(1 - \frac{2\delta^2 \max |\bigtriangledown\, q(p(\mathbf{x}))|_2}{|\mathbf{x}_i - \mathbf{x}_j|_2}\right) \\ &\geq d_{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_j) \left(1 - \frac{2\delta^2 \max |\bigtriangledown\, q(p(\mathbf{x}))|_2}{\epsilon}\right) \end{aligned}$$

Hence,

$$(1 - \lambda_2) d_{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_j) \leq d_M(\mathbf{x}_i, \mathbf{x}_j) \leq d_{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_j) \qquad \forall \lambda_2 \geq \frac{2\delta^2 \max |\bigtriangledown\, q(p(\mathbf{x}))|_2}{\epsilon}$$

□

Theorem 12 (Lower bound on the computing error). $\forall \zeta < 1/2d$, *a computing error (uniform over all pairs of points* $\mathbf{x}$, $\mathbf{y}$*) of*

$$(1 - \lambda) d_M(\mathbf{x}, \mathbf{y}) \leq d_{\mathbf{g}}(\mathbf{x}, \mathbf{y}) \leq (1 + \lambda) d_M(\mathbf{x}, \mathbf{y})$$

with $\lambda = cn^{-\zeta}$ *can be achieved with probability* $\geq \delta'$ *for sufficiently large data sample* $n \geq N(\delta')$ *(c is a constant).*

Proof. We show that the shortest path along the graph is within λ of the DBD metric, by considering two cases based on the properties of the shortest path. We define a new graph $\mathbf{g}_2$ on the data points which contains only a subset of the edges in $\mathbf{g}$. $\mathbf{g}_2$ contains all edges in $\mathbf{g}$ where $|\mathbf{x}_i - \mathbf{x}_j|_2 \leq \epsilon$. In addition, it contains edges in $\mathbf{g}$ that leave $C_p(\alpha;\epsilon)$ and whose endpoints, $\mathbf{x}_i$ and $\mathbf{x}_j$, lie within δ of the boundary of $C_p(\alpha;\epsilon)$. Note that $\mathbf{g}_2$ is sufficient to approximate all shortest paths between data points. However, it is difficult to compute/generate and hence we define a more dense graph $\mathbf{g}$ with the property that the extra edges are most likely not going to be used in the shortest path unless they form a good approximation to the shortest path along $\mathbf{g}_2$.

Case (a) : *The shortest path along* $\mathbf{g}$ *lies entirely within the subset* $\mathbf{g}_2$. Using the Theorem from [87], we can conclude that the choice in Section VI.B of kernel width, $h_n = \frac{1}{n^{\frac{1}{2s+d}}}$ and other properties assumed about $p(\mathbf{x})$ ensure that almost surely,

$$\max_{\mathbf{x}} |p_n(\mathbf{x}) - p(\mathbf{x})| = \mathcal{O}\left(\sqrt{\left(\frac{(2s+d)\log(n)}{n^{\frac{2s}{2s+d}}}\right)}\right)$$

This means that for sufficiently large n, $\forall$ points $\mathbf{y}$ in $C_p(\alpha;2\epsilon)$ have the property that $p(\mathbf{y}) \geq \alpha - \alpha_1$ for arbitrarily small α_1. Using this fact and the δ-sampling condition (see [79, 78]), we know that the requirement for Lemma 9 is satisfied when $n = \Omega\left(\left(\frac{1}{\delta}\right)^d \log\frac{1}{\delta}\right)$. This condition is satisfied with a choice of $\zeta < 1/2d$ and letting $\delta = c_1 n^{-2\zeta}$ and $\epsilon = c_2 n^{-\zeta}$) ($c_1$ and c_2 are constants). Let $\lambda_3 = max(\lambda_1, \lambda_2)$, where λ_1 and λ_2 are defined in Lemma 10 and 11 respectively. Hence we can use Lemmas 9, 10 and 11 to conclude that

$$(1 - 2\lambda_3)d_M(\mathbf{x},\mathbf{y}) \leq d_{\mathbf{g}}(\mathbf{x},\mathbf{y}) \leq (1 + 2\lambda_3)\left(1 + \frac{6\delta}{\epsilon} + \frac{8\delta^2}{\epsilon^2}\right) d_M(\mathbf{x},\mathbf{y})$$

which implies that

$$(1 - \lambda_4)d_M(\mathbf{x},\mathbf{y}) \leq d_{\mathbf{g}}(\mathbf{x},\mathbf{y}) \leq (1 + \lambda_4)d_M(\mathbf{x},\mathbf{y}),$$

where

$$\lambda_4 = \mathcal{O}(\epsilon + \frac{\delta}{\epsilon}) = \mathcal{O}(n^{-\zeta}).$$

Case (b) : *The shortest path, P, along* $\mathbf{g}$ *uses some edges that are not part of* $\mathbf{g}_2$. Consider any edge E connecting $\mathbf{x}_l$ and $\mathbf{x}_m$ in the shortest path along $\mathbf{g}$ that is not in $\mathbf{g}_2$. We will show that there is a path through $\mathbf{g}_2$ that can closely approximate this edge E and hence this shortest path. We consider the case when only one section near the end point $\mathbf{x}_m$ is more than δ in $C_p(\alpha;\epsilon)$. The case when more sections of E are in $C_p(\alpha;\epsilon)$ can be similarly handled. Consider the boundary point r_b where the straight line starting at $\mathbf{x}_m$ toward $\mathbf{x}_l$ first touches the edge of $C_p(\alpha;\epsilon)$. By the δ-sampling condition, there is a data point $\mathbf{x}_k$ within δ of r_b. Consider the path consisting of the edge $\mathbf{x}_l$–$\mathbf{x}_k$ and the shortest path, P_2, between $\mathbf{x}_k$ and $\mathbf{x}_m$ through those edges of $\mathbf{g}$ that connect nodes within ϵ of one another. Let $d'_{\mathbf{g}2}$ be the length of a path that follows P except when it comes to edges not in $\mathbf{g}_2$ in which case it follows paths P_2 constructed to pass through $\mathbf{g}_2$. Let $d_{\mathbf{g}2}$ be the length of shortest path along graph $\mathbf{g}_2$. From proof of case (a), we know that

$$(1-\lambda_4)d_M(\mathbf{x},\mathbf{y}) \leq d_{\mathbf{g}2}(\mathbf{x},\mathbf{y}) \leq (1+\lambda_4)d_M(\mathbf{x},\mathbf{y}),$$

$$d_{\mathbf{g}}(\mathbf{x},\mathbf{y}) \leq d'_{\mathbf{g}2}(\mathbf{x},\mathbf{y}) \leq (1+\lambda_4)d_{\mathbf{g}}(\mathbf{x},\mathbf{y})$$

$$\text{and} \quad d_{\mathbf{g}}(\mathbf{x},\mathbf{y}) \leq d_{\mathbf{g}2}(\mathbf{x},\mathbf{y}) \leq d'_{\mathbf{g}2}(\mathbf{x},\mathbf{y}).$$

Hence,

$$(1-2\lambda_4)d_M(\mathbf{x},\mathbf{y}) \leq d_{\mathbf{g}}(\mathbf{x},\mathbf{y}) \leq (1+\lambda_4)d_M(\mathbf{x},\mathbf{y})$$

□

VI.C.2 Upper bound

In Theorem 12, we showed that we can construct a neighborhood-based graph on the data sample which can be used to approximately compute DBD metrics with a rate of convergence of $1/n^{1/2d}$. This is a very slow rate of convergence, especially when data dimension, d, is high. The natural question that follows this analysis is whether this dependence of the rate on the data dimension is because of curse of dimensionality or whether it is merely because of the way the graph

was constructed and analyzed. Theorem 13 shows that dimension does limit how much we can reduce the approximation error, regardless of the particular graph construction method we use, so long as we choose to use a neighborhood-based graph. This result is true even when data lies along a manifold, but is noisy and hence does not lie perfectly on the manifold, i.e., curse of dimensionality cannot be overcome in the case of approximation error when using neighborhood based graphs, even when the intrinsic dimension of data is small. For this reason, this result provides a lower bound on the approximation error of the ISOMAP algorithm [79] as well.

Theorem 13 (Upper bound on the computing error). *The computing error, when using an ϵ-neighborhood based graph on the data sample, cannot converge to zero faster than $\frac{1}{n^{\frac{1}{d-1}}}$ with probability $\geq \delta'$ for sufficiently large data sample $n \geq N(\delta')$.*

Proof. This result is shown using an example for which the approximation error when using the graph converges at rate $1/n^{\frac{1}{d-1}}$. Consider the case when data density is uniform over any convex set. (Note that all continuous density functions can be approximated by a constant function in a small enough neighborhood.) In this case the graph construction method described at the beginning of this section reduces to an ϵ- neighborhood graph (with high probability). Consider any two points $\mathbf{x}'$, $\mathbf{x}''$ in the interior of the support of the density. The shortest path between $\mathbf{x}'$ and $\mathbf{x}''$ is the straight line joining them. Consider a d-dimensional cuboid which circumscribes a cylinder of radius $\delta/2$ around this line. If none of the points in the data sample lie in this cuboid, the approximation error in measuring the length of this line along the graph edges will be at least of order δ. The probability of this happening, $(1 - c\delta^{d-1})^n$, can be lower bounded by a constant if δ is chosen to be of order $1/n^{\frac{1}{d-1}}$. □

VI.D Approximating minimal geodesics

The support of $p(\mathbf{x})$ with Riemannian metric defined by $g(p(\mathbf{x}))$ is a closed and bounded manifold, i.e., it is complete. Hence by Hopf-Rinow theorem [92] any two points in the manifold are connected by a minimal geodesic. This minimal geodesic may not be unique. An example with infinitely many minimal geodesics between two points is the uniform distribution on the surface of a sphere in which case each longitude is a minimal geodesic connecting the north and south poles.

We will show that the graph of the shortest path between any two points, $\mathbf{x}'$ and $\mathbf{x}''$, along the graph G constructed as described in Section VI.C converges to the set of minimal geodesics connecting the points. To prove convergence of the paths, we need to define a topology (or metric) on the space of paths. We use a commonly used metric which leads to the compact-open topology [93] on the space of paths (the proof can be modified for other metrics). Let S_p be the space of paths of unit speed between $\mathbf{x}'$ and $\mathbf{x}''$.

Definition 14. *The distance between two unit speed paths, p_1 and p_2, is defined to be*

$$d(p_1, p_2) = \sup_{t \in \mathcal{R}^+} |p_1(t) - p_2(t)|_2.$$

Let T be the quotient topology obtained using the compact-open topology and the equivalence relationship between paths given by $p_1 \sim p_2$ if lengths of p_1 and p_2 are equal.

Theorem 15. *The graph of the shortest path along graph G converges to the set of minimal geodesics in the topology T (see section VI.C for definition of the graph).*

Proof. Consider path-length as a map into positive reals from the space of all shortest paths along G_n between the points for all sample sizes n. Since we have already shown that the length of the shortest path along G converges to the length of a minimal geodesic, it is sufficient to show that length is a continuous function on this subset of S_p with the quotient topology T. Let p_2 be a minimal geodesic

path and p_1 be a shortest path along G_n for some n. Given that $d(p_1, p_2) \leq \delta$, the difference between their lengths is

$$\left| \int \left| \frac{dp_1}{dt} \right|_2 dt - \int \left| \frac{dp_2}{dt} \right|_2 dt \right| \leq 3\delta$$

for all n large enough such that the δ-sampling condition is satisfied with high probability. Note that length is a continuous function only on this subset and not on all of S_p. □

VI.E Applications and experiments

VI.E.1 Semi-supervised learning using density based metrics

Given a density-based distance metric, any of the nearest neighbor based methods (K-nearest neighbors, weighted K-nearest neighbors with various weights) can be used for classification in a semi-supervised learning scenario. Let y_i be the label of $\mathbf{x}_i$ and let classifier be $sign(h(\mathbf{x}_i))$. Let l_M denote the Lipschitz constant according to the manifold specified by $q(p(\mathbf{x}))$. In this manifold, the lengths scale locally as $q(p(\mathbf{x}))$, hence it can be verified that for any function h on $\mathcal{R}^d$

$$|l_M h|_2 = \sup_{\mathbf{x}} \frac{1}{q(p(\mathbf{x}))} |\bigtriangledown_{\mathbf{x}} h|_2.$$

[97] have shown that the 1-nearest neighbor classifier corresponds to a large-margin classifier. In the case of the DBD metric, 1-NN is equivalent to (using the modified Lipschitz constant according the density-based manifold), the optimization problem

$$\arg\min_h \sup_{\mathbf{x}} \left[\frac{1}{q(p(\mathbf{x}))} |\bigtriangledown_{\mathbf{x}} h|_2 \right] \text{ under constraints } y_i h(\mathbf{x}_i) \geq 1.$$

As $p(\mathbf{x})$ increases, $\frac{1}{q(p(\mathbf{x}))}$ also increases and hence this optimization problem corresponds to penalizing the gradient of the classifier function h in high density regions and allowing h to change in the low density regions. This agrees with the intuition that data points in the same high density region are likely to have similar labels.

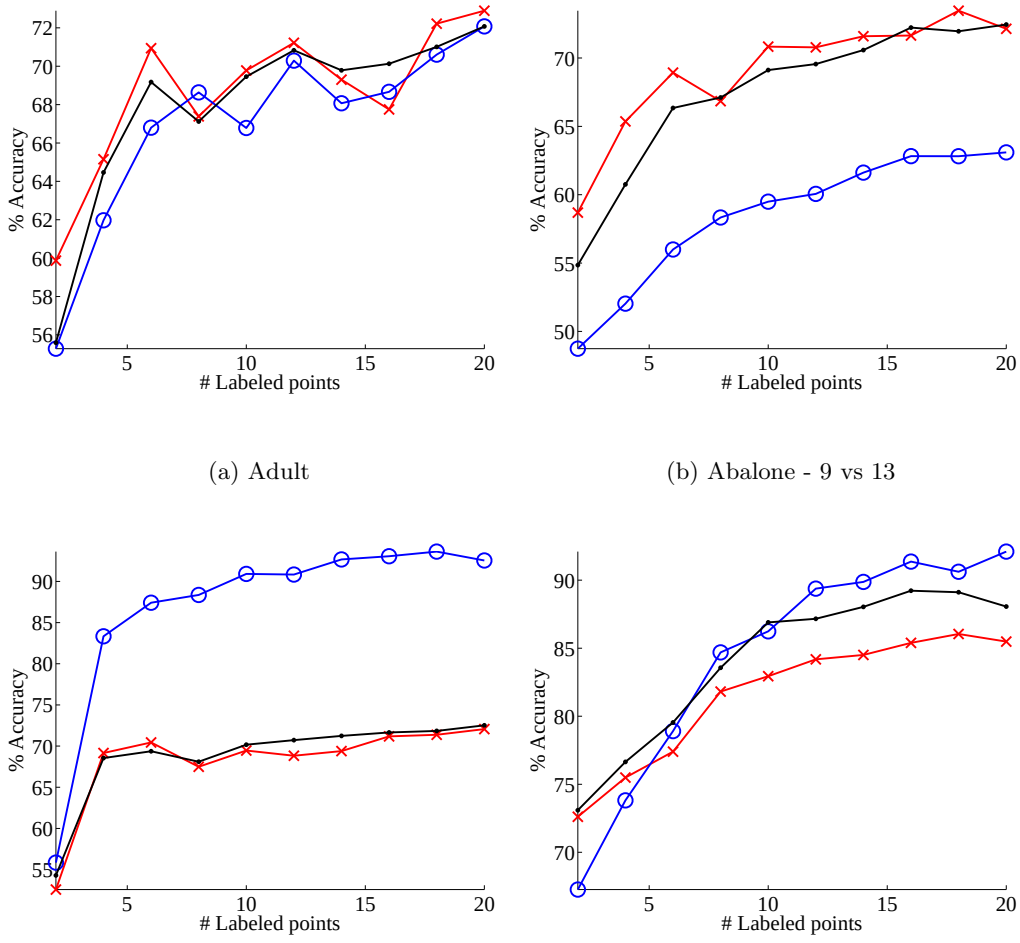
Please see [69] for a discussion on regularization appropriate for semi-supervised learning and its relationship to modifying geometry based on the data density.

In this section, we present experimental results on data from the UCI machine learning repository, summarized in Table VI.1. The three methods we compare are standard 1-nearest neighbor, DBD metric based 1-nearest neighbor and the randomized min-cut method ([66]). The randomized min-cut method involves averaging over results obtained from several min-cuts and it is suggested by [66] that those min-cuts which lead to a very unbalanced classification are to be rejected. However, there is no clear way to choose this cut-off ratio. For the results presented here we choose the cutoff to be slightly smaller than the 'true' ratio between the classes in the dataset. For the DBD based 1-NN implementation, we chose the function q to fall exponentially with increase in density beyond α which in turn was chosen to be smaller than the estimated density at all sample points.

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

Data set	data dimension	data set size	class ratio
Adult	6	1000	0.30
Abalone - 9 vs 13	7	892	0.29
Abalone - 5 vs 9	7	804	0.17
Digits - 1 vs 2	256	2200	1.00

We performed experiments for labeled set size varying between 2 and 20 and the accuracy results are shown in Fig. VI.2. We observed that DBD based 1-NN performed better than or similar to the standard 1-NN algorithm for all datasets with small dimension. We conjecture that the reason DBD based 1-NN performed worse than 1-NN for the digits example is because of difficulty in density estimation in very high dimensions. DBD-based 1-NN algorithm performed better than the other two when the number of labeled examples was very small, except in the case of digits example. One interesting result was that of the two abalone



(a) Adult

(b) Abalone - 9 vs 13

(c) Abalone - 5 vs 9

(d) Digits - 1 vs 2

Figure VI.2: Classification results comparing 1-NN ('.'), DBD based 1-NN ('x') and Randomized Mincut ('o') algorithms

data examples, in which the randomized min-cut algorithm performed much better than both NN algorithms in one case and much worse in the other.

VI.E.2 Non-linear interpolation

In density-based interpolation, given two points, our task is to find a path that respects the statistical model of the data. In particular, the desired interpolant should not pass through regions of space to which the modelled density

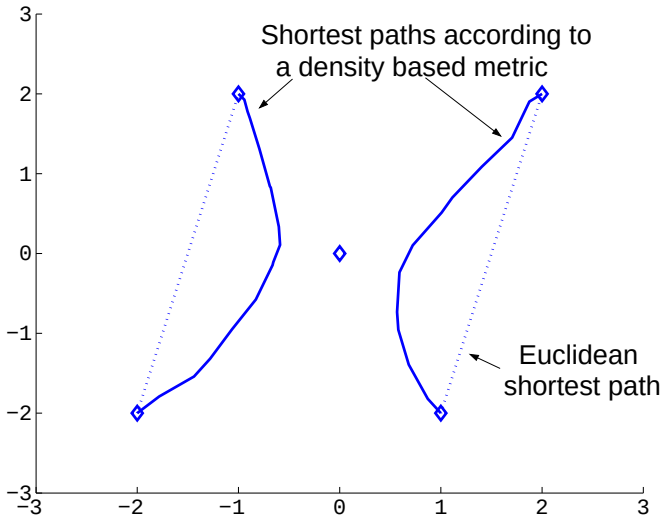


Figure VI.3: Density-based non-linear interpolation using 1000 iid samples drawn from a spherical, unit variance, zero mean Gaussian distribution.

$f(\mathbf{x})$ assigns low probability [76]. Given a sample of points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we can find an approximation to such a path by computing the weighted graph G as described in section VI.C and tracing the shortest path between the two points through the graph. We illustrate this using a simple example where data is drawn from a single spherical Gaussian distribution with mean at $(0, 0)$ and variance one in each direction. The shortest path according to a DBD metric with $g = \exp(-\frac{f(\mathbf{x})-\alpha}{f_{\max}-\alpha})$ and based on 1000 data samples drawn from the Gaussian distribution is shown in figure VI.3.

VI.F Acknowledgement

The material presented in this chapter has been published in the Proceedings of the International Conference on Machine Learning 2005 and as a chapter in the book 'Semi-supervised Learning', MIT press 2006. The dissertation author was the primary investigator and the first author of these publications.

Chapter VII

Conclusion

This dissertation is concerned with non-parametric techniques adapted to various characteristics of the data-sets including their high dimensionality, large volume, different data types (for example binary or integer), partially available data etc.

In Chapter III, we presented an unsupervised dimensionality reduction method that is applicable to binary, integer and other data types. This method models data as consisting of a parametric noise that is added to an arbitrary (non-parametric) distribution in a lower dimensional space. Unlike previous methods, this algorithm is guaranteed to be asymptotically consistent (modulo identifiability, see Section III.E) in finding the lower dimensional subspace. We demonstrated using experiments that that this method recovers the true subspace when other competing methods fail to do so. Using simulations on standard text and image datasets, we demonstrated that it is effective in separating different populations, in projecting similar observed data points close to one another in the representation space and in generalization to unseen samples.

In Chapter IV, we argued that use of maximum conditional likelihood estimation is a natural way to utilize mixture models in a supervised setting. We presented an efficient, iterative EM-like algorithm to compute the best lower dimensional subspace that contains maximum discriminating information. Experiments

with data-sets containing class labels demonstrate the potential of this method to learn transformations that lead to competitive classification accuracy results and for supervised visualization of high dimensional data.

Despite a large amount of literature over several decades, dimensionality reduction remains an active area for research. Our dimensionality reduction method performs well while estimating the non-parametric prior and the lower dimensional subspace from small data-sets (see Sections III.G.2 and III.G.3) and this is critically dependent on the fact that we are estimating a linear subspace. This is because of the strong regularization effect of the linearity of the subspace that is being estimated. While the same theoretical results of consistency hold when the lower dimensional signal subspace is not linear, a challenging problem for future work would be making such a non-linear dimension reduction method with non-parametric prior practically applicable.

While working with various data-sets for the simulation results presented in Chapter III, we found that the objective function being optimized was very non-linear for binary data and even more so when data was integer valued. Hence, an open area for further work is devising approximations to this objective function that are easily optimized or devising more effective optimization algorithms for this function. Similar comments apply for the supervised dimensionality reduction method presented in Chapter IV. 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.

In Chapter VI showed that density-based distance metrics which satisfy certain properties can be estimated consistently using an estimator obtained by

plugging in the kernel density estimate of the data distribution. In terms of s, a smoothness parameter that corresponds to how many times data density is known to be differentiable and d, the data dimension, we showed that the rate of convergence of such an estimator is $\frac{s}{2s+d}$. We showed that no estimator can converge at a rate faster than $\frac{1}{2}$. This contains both good and bad news. The knowledge that we have consistent estimation is useful when applying the method to voluminous data (for example web pages). However, we expect d to be high for many machine learning applications and we might not be able to assume that the smoothness parameter, s, is very high. Hence, when using the plug-in estimator, the convergence rate can be very slow for high-dimensional data.

We showed a graph construction method that can be used for consistent computation of DBD metrics and shown that with high probability, the approximation error when using this graph goes to zero faster than $1/n^{1/2d}$ with high probability. We also showed that shortest distance along a nearest-neighborhood based graph on the data cannot converge to true distance faster than $1/n^{1/(d-1)}$ with high probability. We presented semi-supervised classification results that demonstrate that using DBD metrics can sometimes improve performance over using simple Euclidean distance, when data density can be estimated with reasonable reliability.

While we have given a theoretical understanding of DBD metrics, further experimental investigation of their use for semi-supervised learning in needed to make them a practically viable choice. While several papers have considered DBD metrics, the only papers that present experimental results with real world data use the 1-nearest neighbor algorithm ([74, 64]). Experiments using these metrics with other classification algorithms, using parametric density estimation in place of the kernel density estimator and studying alternative graph construction and weighting methods for more accurate and efficient computation will be of practical value.

Bibliography

[1] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 2nd edition, 2002.

[2] Lazarsfeld, Paul F. and Neil W. Henry. *Latent Structure Analysis*. Houghton Mifflin, Boston, 1968.

[3] C. Radhakrishna Rao. *Linear statistical inference and its applications*. John Wiley and Sons, New York, 1965.

[4] Richard O. Duda, Peter E. Hart and David G. Stork. *Pattern Classification*. Wiley Interscience, 2nd edition, 2000.

[5] Trevor Hastie, Robert Tibshirani and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in Statistics, 2001.

[6] K. V. Mardia, J. T. Kent and J. M. Bibby. *Multivariate analysis*. Academic Press, New York, 1979.

[7] Brian D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, New York, 1996.

[8] W.N. Venables and B.D. Ripley. *Modern applied statistics with S-PLUS*. Springer, New York, 3rd edition, 1999.

[9] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 3rd edition, 1961.

[10] D. L. Donoho. *High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality*. Lecture on August 8, 2000, to the American Mathematical Society "Math Challenges of the 21st Century", Available from http://www-stat.stanford.edu/ donoho/, 2000.

[11] I. Ibragimov and R. Z. Khasminskii. *Statistical estimation: asymptotic theory*. Springer, New York, 1981.

[12] A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 8(3): 930-945, 1993.

[13] 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.

[14] Nathan Srebro and Tommi Jaakkola. Linear dependent dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, 2003.

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

[16] Sajama and A. Orlitsky. Semi-parametric exponential family PCA. In *Advances in Neural Information Processing Systems 17*, 2005.

[17] Christopher M. Bishop, Markus Svensén and Christopher K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215-318, 1998.

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

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

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

[21] 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, 2001.

[22] 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.

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

[24] Sam Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems, 10*, 1998.

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

[26] 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.

[27] B. G. Lindsay. The geometry of mixture likelihoods : A general theory. *The Annals of Statistics*, 11(1):86-104, 1983.

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

[29] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1983.

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

[31] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1-38, 1977.

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

[33] Francis Bach. The kernel-ica package. http://www.cs.berkeley.edu/∼fbach/kernel-ica/index.htm, 2002.

[34] 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.

[35] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:158-176, 1996.

[36] Trevor Hastie, Robert Tibshirani and Andreas Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89(428):1255-1270, 1994.

[37] N. Kumar and A. Andreou. Generalization of linear discriminant analysis in a maximum likelihood framework. In *Proceedings of the Joint Meeting of the American Statistical Association*,1996.

[38] Kari Torkkola and William M. Campbell. Mutual information in learning feature transformations. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[39] Aldebaro Klautau, Nikola Jevtic and Alon Orlitsky. Discriminative gaussian mixture models: A comparison with kernel classifiers. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[40] T. Jebara and A. Pentland. Maximum conditional likelihood via bound maximization and the CEM Algorithm. In *Advances in Neural Information Processing Systems 11*, 1998.

[41] L. K. Saul and D. D. Lee. Multiplicative updates for classification by mixture models. In *Advances in Neural Information Processing Systems 14*, 2002.

[42] S. M. Omohundro. Efficient algorithms with neural networks behaviour. *Complex Systems*, 1(2):273-347, 1987.

[43] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[44] Katy S. Azoury and Manfred K. Warmuth. Relative loss bounds for online density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211-246, 2001.

[45] Jurgen Forster and Manfred K. Warmuth. Relative expected instantaneous loss bounds. In *Computational Learing Theory*, 2000.

[46] J. F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization*. Springer Verlag, 2003.

[47] B. G. Lindsay. The geometry of mixture likelihoods : A general theory. *The Annals of Statistics*, 11(1):86-104, 1983.

[48] J. Park and L. W. Sandberg. Universal approximation using radial basis function networks. *Neural Computation*, 3:246-257, 1991.

[49] 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.

[50] Sajama and Alon Orlitsky. Semi-parametric exponential family PCA. In *Advances in Neural Information Processing Systems 17*, 2004.

[51] Jerome H. Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the american statistical association*, 76:817-823, 1981.

[52] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical Science*, 1:297-318, 1986.

[53] Ker-Chau Li. Sliced inverse regression for dimension reduction (with discussion). *Journal of american statistical association*, 86: 316-342, 1991.

[54] 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.

[55] Amir Globerson and Naftali Tishby. Sufficient dimensionality reduction. *Journal of Machine Learning Research*, 3:1307-1331, 2003.

[56] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537-550, 1994.

[57] Isabelle Guyon and Andr Elisseeff. An Introduction to variable and feature selection. *Journal of Machine Learning Research*, 3: 1157-1182, 2003.

[58] Leo Breiman and Jerome H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80:580-598, 1985.

[59] P.A. Devijver and J. Kittler. *Pattern recognition: A statistical approach*. Prentice Hall, London, 1982.

[60] X. Guorong, C. Peiqi and W. Minhui. Bhattacharyya distance feature selection. In *Proceedings of the 13th International conference on Pattern recognition*, volume 2, pages 25-29, 1996.

[61] G. Saon and M. Padmanabhan. Minimum Bayes error feature selection for continuous speech recognition. In *Advances in Neural Information Processing Systems 13*, 2001.

[62] E. A. Nadaraya, *Nonparametric estimation of probability densities and regression curves*. Kluwer Academic Publishers, 1989.

[63] P. Vincent and Y. Bengio. Density-Sensitive Metrics and Kernels. In *Workshop on Advances in Machine Learning*, Montreal, Quebec, Canada, 2003.

[64] Sajama and Alon Orlitsky. Estimating and computing density based distance metrics. In *Proceedings of the 22th International Conference on Machine Learning*, 2005.

[65] Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[66] Avrim Blum, John D. Lafferty, Mugizi Robert Rwebangira and Rajashekar Reddy. Semi-supervised learning using randomized mincuts. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[67] X. Zhu, Z. Ghahramani and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[68] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, 1999.

[69] O. Bousquet, O. Chapelle and M. Hein. Measure based regularization. In *Advances in Neural Information Processing Systems 16*, 2004.

[70] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems 17*, 2005.

[71] C. Kemp, T. L. Griffiths, S. Stromsten and J. B. Tenenbaum. Semi-supervised learning with trees. In *Advances in Neural Information Processing Systems 16*, 2003.

[72] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2):103-134, 2000.

[73] O. Chapelle, J. Weston and B. Schoelkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems 15*, 2003.

[74] G. Lebanon. Learning Riemannian metrics. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003.

[75] Adrian Corduneanu and Tommi Jaakkola. On Information regularization. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*, 2003.

[76] L. K. Saul and M. I. Jordan. A variational principle for model-based morphing. In *Advances in Neural Information Processing Systems 9*, 1997.

[77] C. Bregler and S. Omohundro. Nonlinear image interpolation using manifold learning. In *Advances in Neural Information Processing Systems 7*, 1995.

[78] M. Bernstein, V. de Silva, J. C. Langford and J. B. Tenenbaum. *Graph approximations to geodesics on embedded manifolds*. Manuscript, 2000.

[79] J. B. Tenenbaum, V. de Silva and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319-2323, 2000.

[80] Larry Goldstein and Karen Messer. Optimal plug-in estimators for nonparametric functional estimation. *The annals of statistics*, 20(3):1306-1328, 1992.

[81] Charles J. Stone. Optimal rates of convergence for nonparametric estimators. *The annals of statistics*, 8(6):1348-1360, 1980.

[82] Luc Deveroye and Laszlo Gyorfi. *Nonparametric density estimation : The L1 view*. John Wiley, New York, 1985.

[83] H. G. Muller. Smooth optimum kernel estimators of regression curves, densities and modes. *Annals of Statistics*, 12:766-774, 1984.

[84] J. Lafferty and G. Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129-163, 2005.

[85] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56:209-239, 2004.

[86] F. J. Narcowich. Generalized Hermite interpolation and positive definite kernels on a Riemannian manifold. *Journal of Mathematical Analysis and Applications*, 190:165-193, 1995.

[87] E. Gine and A. Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, 38(6):907-921, November 2002.

[88] Bruce E. Hansen. *Exact Mean Integrated Squared Error of Higher-Order Kernels*, Unpublished manuscript, http://www.ssc.wisc.edu/∼ bhansen, June 2003.

[89] Robert Strichartz. *The Way of Analysis*. Jones and Bartlett, 1995.

[90] James A. Sethian. *Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1999.

[91] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew K. McCallum, Tom M. Mitchell, Kamal Nigam and Seán Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the 15th Conference of the American Association for Artificial Intelligence*, 1998.

[92] John W. Milnor. *Morse Theory*. Princeton University Press, 1963.

[93] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[94] Steven Rosenberg. *The Laplacian on a Riemannian manifold*. Cambridge University Press, 1997.

[95] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Tenth international workshop on artificial intelligence and statistics*, 2005.

[96] Bernd Fischer, Volker Roth and Joachim M. Buhmann. Clustering with the connectivity kernel. In *Advances in Neural Information Processing Systems 16*, 2004.

[97] U. von Luxburg and O. Bousquet. Distance-based classification with Lipschitz functions. *Journal for Machine Learning Research*, 5:669-695, 2004.

[98] Jonathan Foote. Content-based retrieval of music and audio. *Multimedia Storage and Archiving Systems II, Proceedings of SPIE* 3229:138-147, 1997.

[99] David A. Grossman. *Information retrieval : algorithms and heuristics*. Kluwer, Boston, 1998.

[100] J. Nilsson, T. Fioretos, M. Hoglund and M. Fontes. Approximate geodesic distances reveal biologically relevant structures in microarray data. *Bioinformatics*, 20(6):874-880, 2004.

[101] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification and regression. In *Advances in Neural Information Processing Systems 8*, 1996.

[102] E. Xing, A. Ng, M. Jordan and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, 2003.

[103] Kilian Q. Weinberger, John Blitzer and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18*, 2006.

[104] Matthew Schultz and Thorsten Joachims. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 15*, 2003.

[105] M. Belkin and P. Niyogi. Using manifold structure for partially labeled classification. In *Advances in Neural Information Processing Systems*, 2002.

[106] B. Shahshahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions On Geoscience and Remote Sensing* 32:1087-1095, 1994.

[107] X. Zhu, J. Kandola, Z. Ghahramani and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2005.

[108] V. Sindhwani, P. Niyogi and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[109] Jorg Sander, Martin Ester, Hans-Peter Kriegel and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, 2(2):169-194, 1998.

[110] Martin Ester, Hans-Peter Kriegel, Jrg Sander and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[111] J. A. Garca, J. Fdez-Valdivia, F. J. Cortijo and R. Molina. A dynamic approach for clustering data. *Signal Processing*, 44(2):181-196, 1994.

[112] Yasser El-Sonbaty, M. A. Ismail and Mohamed Farouk. An efficient density based clustering algorithm for large databases. In *16th IEEE International Conference on Tools with Artificial Intelligence*, 2004.

[113] Jae-Joon Hwang, Kyu-Young Whang, Yang-Sae Moon and Byung-Suk Lee. A top-down approach for density-based clustering using multidimensional indexes. *Journal of Systems and Software*, 73(1):169-180, 2004.

[114] Daoying Ma and Aidong Zhang. An adaptive density-based clustering algorithm for spatial database with noise. In *Fourth IEEE International Conference on Data Mining*, 2004.

[115] M. Emre Celebi, Y. Alp Aslandogan and Paul R. Bergstresser. Mining biomedical images with density-based clustering. In *International Conference on Information Technology: Coding and Computing*, 2005.

[116] D. Beymer, A. Shashua and T. Poggio. *Example based image analysis and synthesis*. Technical report, MIT AI Lab, AIM-1431, 1993.

[117] O. Chapelle, B. Schölkopf and A. Zien, *Semi-Supervised Learning*. MIT Press, Cambridge, 2006.