Dimensionality Reduction on Statistical Manifolds

by

Kevin Michael Carter

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering-Systems)
in The University of Michigan
2009

Doctoral Committee:

Professor Alfred O. Hero III, Chair
Professor William G. Finn
Assistant Professor Elizaveta Levina
Assistant Professor Clayton D. Scott
Assistant Professor Raviv Raich, Oregon State University

To Him through which all things are possible

ACKNOWLEDGEMENTS

This work could not have been possible without the support of many individuals, and I would be remiss if I did not take the opportunity to thank them. To start, I give the utmost thanks to my advisor, Professor Alfred Hero. He not only took me under his wing as a research assistant, but was a major contributor to my professional development. While his otherworldly knowledge base was critical towards my maturation as a researcher, his motivation, mentorship, and words of advice kept me going during difficult and stressful times. I would also like to thank Professor Raviv Raich, who has worked side-by-side with me throughout my entire research experience. Whenever I came upon a road block, I knew I could count on Raviv to have the patience and wherewithal to guide me through.

My ability to progress so quickly throughout this process was due in large part to my amazing research project. I owe this entirely to Dr. William Finn and the Department of Pathology at the University of Michigan, who came to us with an idea and a lot of data. Dr. Finn was always available for discussion and insight into the process of flow cytometry, and throughout my development he has shown a genuine excitement for all of the work I have done. Without his knowledge, support, and enthusiasm, none of this work would have been completed.

This work has also benefited from discussions with the remainder of my committee members. A special thanks goes to Professor Elizaveta Levina and Professor Clayton Scott for their input and support. Their level of expertise in many of the areas

directly coinciding to my research topics was very beneficial, and the third-party review strengthened my work substantially. This work was funded by the National Science Foundation, grant No. CCR-0325571, a Rackham Merit Fellowship, and the Department of Pathology at the University of Michigan.

While the University of Michigan has offered me an outstanding opportunity to study under the leaders and best, it has also given me the chance to work with exceptional individuals I have the honor of calling my peers. I would first like to thank Eran Bashan, without whom I may not have made it past my first year and qualifying exams. I would also like to thank all of those in Professor Hero's research group that have provided assistance and thought provoking debate. While too numerous to name in entirety, I want to specifically mention Ami Weisel, Mark Kliger, Neal Patwari, Jose Costa, Arvind Rao, Patrick Harrington, Sung Jin Hwang, Kumar Sricharan, Christine Kim, Kevin Xu, and Kyle Herrity.

In addition to those who have helped me develop academically, I would like to thank those who have provided a support network for me. Special thanks goes to SMES-G and all of its members, as well as Pastor Mark J. Lyons and the membership of SBC. Most importantly I would like to thank the guys that have kept me grounded and humbled throughout the years: Korey, Ced, Brad, and Jason.

Finally, and most of all, I have to thank my family for their constant love and support. No matter what issues I may have encountered, my parents were always there to offer a loving comment, a faithful word of advice, and a sincere "I'm proud of you". My sister would always check in on her baby brother to make sure I was doing alright. My grandparents, aunts, uncles, and cousins would offer overwhelming praise for all of my endeavors. Words cannot express how thankful I am that I was blessed with such a wonderful family. I love you all.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Dimensionality Reduction on Statistical Manifolds

by

Kevin Michael Carter

Chair: Alfred O. Hero III

This thesis concerns the problem of dimensionality reduction through information geometric methods on statistical manifolds. While there has been considerable work recently presented regarding dimensionality reduction for the purposes of learning tasks such as classification, clustering, and visualization, these methods have focused primarily on Riemannian sub-manifolds in Euclidean space. While sufficient for many applications, there are many high-dimensional signals which have no straightforward and meaningful Euclidean representation. In these cases, signals may be more appropriately represented as a realization of some distribution lying on a *statistical* manifold, or a manifold of probability density functions (PDFs). These manifolds are often intrinsically lower dimensional than the domain of the data realization.

We begin by first discussing local intrinsic dimension estimation and its applications. There has been much work done on estimating the global dimension of a data set, typically for the purposes of dimensionality reduction. We show that by estimating dimension locally, we are able to extend the uses of dimension estimation to statistical manifolds as well as many applications which are not possible with global

dimension estimation. We illustrate independent benefit of dimension estimation on complex problems such as anomaly detection, clustering, and image segmentation.

We then discuss two methods of dimensionality reduction on statistical manifolds. First, we propose a method for statistical manifold reconstruction that utilizes the principals of information geometry and Euclidean manifold learning to embed PDFs into a low-dimensional Euclidean space. This embedding enables comparative analysis of multiple high-dimensional data sets using standard Euclidean methods. Our second algorithm proposes a linear projection method which creates a dimension reduced subspace which preserves the high-dimensional relationships between multiple signals. Defining this information preserving projection contributes to both feature extraction and visualization of high-dimensional data.

Finally, we illustrate these techniques toward their original motivating problem of clinical flow cytometric analysis. These methods of dimensionality reduction approach the problems of diagnosis, visualization, and verification of flow cytometric data in a manner which has not been given significant consideration in the past. The tools we propose are illustrated for several case studies on actual patient data sets.

CHAPTER I

Introduction

1.1 The Curse of Dimensionality

In the recent past, sensing and media storage capabilities have enabled the generation of enormous amounts of information, often in the form of high-dimensional data. This is easily viewed within sensor networks, imaging, and biomedical applications such as flow cytometry and gene micro-arrays. While this vast amount of retrieved data has opened a wealth of opportunities for data analysis, the problem of the *curse of dimensionality* has become more substantial. High-dimensional data is inherently difficult to analyze for a multitude of reasons. As the dimensionality increases, it becomes much more computationally complex for learning algorithms to effectively perform. Additionally, higher dimensions require significantly more points to fill the space. As an example, the probability that a point sampled from a uniform distribution on a hypercube in $\mathbb{R}^m$ will lie within some distance ϵ from a boundary on the cube approaches 1 as $m \to \infty$ and $\epsilon \to 0$. This is detailed in Proposition 3.1, but intuitively as the dimensionality of a data set increases, all of the sample points tend to lie near the boundaries of the space. This causes significant problems with many learning algorithms, often resulting in over-fitting and unreliable models.

The high-dimensional nature of data is often simply a product of its representa-

tion. In many instances data dimensions are redundant and entirely correlated with some combination of other dimensions within the same data set. In these instances, although the retrieved data seems to exhibit a naturally high dimension, it is actually constrained to a lower dimensional subset – manifold – of the measurement space. This allows for significant dimensionality reduction with minor or no loss of information. This focus of manifold learning, which is a subset of machine learning, aims at the high dimension regime, in which examples are governed by geometric constraints effectively reducing the dimension of the problem from a high extrinsic dimension to a low intrinsic dimension. There has been much research done in the area of manifold learning in Euclidean space [5, 71, 74], providing algorithms which reconstruct manifolds based on the geometric properties of samples within a data set.

Often data does not exhibit a low intrinsic dimension in the data domain as one would have in Euclidean manifold learning. For example, data generated by a multivariate Gaussian distribution with i.i.d. dimensions $\mathcal{N}(\mu, \sigma^2 I)$, $\mu \in \mathbb{R}^d$, is not constrained to lie on any low dimensional Euclidean manifold in $\mathbb{R}^m$, $m < d$. A straightforward strategy is to express the data in terms of a low-dimensional feature vector to alleviate the dimensionality issue. This initial processing of data as real-valued feature vectors in Euclidean space, which is often carried out in an ad hoc manner, has been called the "dirty laundry" of machine learning [30]. This procedure is highly dependent on having a good model for the data, and in the absence of such a model may be highly suboptimal. This problem is even more prevalent when there is no straight-forward Euclidean representation of the data, which is the reality in many practical applications such as document classification, flow cytometry analysis, face recognition, and shape analysis.

The aim of this thesis is the development of nonparametric methods of dimensionality reduction which work on statistical models of retrieved data. We intend to discover low-dimensional representations of data which maintain the information-geometric structure of the original high-dimensional data.

1.2 Background and Previous Work

1.2.1 Dimension Estimation

When the intrinsic dimension is assumed constant over the data set, several algorithms [10, 24, 50, 58] have been proposed to estimate the dimensionality of the manifold. In many problems of practical interest, however, data will exhibit varying dimensionality, as there may be multiple manifolds of varying dimension supporting the data. In these situations, the local intrinsic dimension may be of more importance than the global dimension. In previous work we illustrated the process of local dimension estimation [12], in which a dimension estimate is obtained for each sample within the data, rather than a single dimension estimate for the entire set.

While dimension estimation has typically been utilized for the purposes of inferring an appropriate projection or embedding dimension, there have also been novel uses presented in which dimensionality reduction is not the final goal. We have presented the ability to use dimension as a means of anomaly detection in router networks [12]. Significant work has been shown using dimension estimation for image and texture segmentation [22, 65], although to our knowledge all methods focus on the usage of fractal dimensions, which in itself requires a specific model assumption [59].

1.2.2 Learning on Statistical Manifolds

Informally, a statistical manifold may be considered as a manifold whose elements consist of probability density functions (PDFs). This information-theoretic construct does not exist in Euclidean space, and is defined by its own properties and metrics [3]. As such, problems which are not easily represented in Euclidean space are often better described with statistical manifolds. Applications of statistical manifolds have been presented in the cases of document classification [53, 55], face recognition [4], texture segmentation [56], image analysis [73], clustering [72], and shape analysis [52]; including our own work on flow cytometry analysis [15, 16, 34] and dimensionality reduction for document classification [19] and object recognition [13].

A common theme to all of the problems is that the model from which the data is generated is unknown, and each paper proposes alternatives to using Euclidean geometry for data modeling. However, outside of our own work, these methods focus on clustering and classification, and do not explicitly address the problems of dimensionality reduction and visualization. Additionally, many focus on parameter estimation as a necessity for their methods.

Recent work by Lee *et al.* [57] similar to our own [14,18,19] has demonstrated the use of statistical manifolds for dimensionality reduction. Their work focuses on the specific case of image segmentation, which consists of modeling images as multinomial distributions which lie on an n-simplex (or projected onto an $(n+1)$-dimensional sphere). By framing their problem as such, they are able to exploit the properties of such a manifold: using the cosine distance as an exact computation of the Fisher information distance, and using linear methods of dimensionality reduction.

1.2.3 Data Projections

There has been much work presented in which high-dimensional data is projected into a low-dimensional space to aid in various learning tasks such as classification and visualization. Much of the work done for unsupervised dimensionality reduction [5, 71, 74] operates in a non-linear framework, which requires re-processing the low-dimensional space whenever new data is available, which may be noticeably different than previous spaces (e.g. scaled or rotated differently). This has been approached with out-of-sample extension methods [6], but it is still a relatively open problem. Linear methods which have been presented often focus on optimizing an objective function – such as variance with principal component analysis (PCA) [37] or independence with independent component analysis (ICA) [46] – of the low-dimensional components.

A common theme to all methods, however, is that the dimensionality reduction is typically performed based on the properties of a single data set. This often causes difficulties when one wishes to find a single low-dimensional projection for multiple related data sets. This setting fits into the supervised Fisher's linear discriminate analysis (LDA) [35, 42, 63] framework by assigning each set a unique class label. However LDA and other supervised methods are designed to separate classes, which is not ideal when attempting to preserve the geometry and similarities between multiple sets.

1.2.4 Flow Cytometry

In flow cytometry, pathologists gather readings of fluorescent markers and light scatter off of individual blood cells from a patient sample, leading to a characteristic multidimensional distribution that, depending on the panel of markers selected, may

be distinct for a specific disease entity. The data from clinical flow cytometry can be considered multidimensional both from the standpoint of multiple characteristics measured for each cell, and from the standpoint of thousands of cells analyzed per sample. Historically, however, clinical flow cytometry analysis has been a step-by-step process of 2-dimensional histogram analysis, and the multidimensional nature of flow cytometry is routinely underutilized in clinical practice.

There have been previous attempts at using machine learning to aid in flow cytometry diagnosis. Some have focused on clustering in the high-dimensional space [78,79], while others have utilized information geometry to identify differences in sample subsets and between data sets [69, 70]. These methods have not satisfied the problem because they do not significantly approach the aspect of visualization for 'human in the loop' diagnosis, and the ones that do [60,61] only apply dimensionality reduction to a single set at a time.

1.3 Contributions of Thesis

The focus of the work presented in this thesis is to apply learning techniques to complex problems using the properties of statistical manifolds and dimensionality reduction. By combining these interest areas, we are able to effectively analyze problems which presented significant difficulties in the past due to Euclidean assumptions, and improve upon those which have shown promise using statistical manifolds but ignoring the dimensionality effects. In order to reduce the dimension of a data set, it is first necessary to know the intrinsic dimension of the data, and we present methods for obtaining this knowledge in Chapter III. Rather than obtaining the global dimension of a data set, which assumes all data is generated by a single manifold, we focus on the local dimensionality, which allows for the recognition of multiple man-

ifolds within a single set. This is useful not only as a precursor for dimensionality reduction, but the information garnered by intrinsic dimension can be applied towards many practical application. We will present several novel problem areas which capitalize on the changes in dimensionality in a data set, such as anomaly detection, clustering, and image segmentation.

Given the intrinsic dimensionality of a statistical manifold, we can effectively recreate said manifold through samples from it. These samples are probability density functions $p(x)$, or realizations thereof, and can be used to find a low-dimensional embedding of the original statistical manifold in Euclidean space $A : p(x) \rightarrow y$, where $y \in \mathbb{R}^m$. In Chapter IV we present an information-geometric framework for determining this embedding, which includes a characterization of data sets in terms of a nonparametric statistical model, a geodesic approximation of the Fisher information distance as a metric for evaluating similarities between data sets, and a dimensionality reduction procedure to obtain a low-dimensional Euclidean embedding of the original collection of high-dimensional data sets for the purposes of both classification and visualization. We present two algorithms to obtain this embedding – which we refer to as *Fisher Information Nonparametric Embedding* (FINE) and *Spherical Laplacian Information Maps* (SLIM).

While FINE/SLIM jointly embeds a collection of high-dimensional data sets in order to recreate the underlying statistical manifold, we are often interested in reducing the dimension in the data domain of each individual set. However, unlike traditional methods of dimensionality reduction and manifold learning, which operate only on an individual data set, there are applications in which the dimensionality reduction is desired to preserve the similarities between multiple sets. Rather than finding the low-dimensional representation which best describes an individual data

set $A : x \to y$ – where once again $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$, $m < d$ – we would like to find the low-dimensional representation which best maintains the relationships within a collection of sets. This is done in a statistical sense by finding the optimal transformation for $A : p(x) \to p(y)$. In Chapter V we present a novel method of dimensionality reduction – *Information Preserving Component Analysis* (IPCA) – whose objective is to preserve the Fisher information distances between the data PDFs in the high-dimensional space when projecting into a low-dimensional space. We will show the usages of IPCA for both supervised and unsupervised problems.

The majority of the work presented here was motivated by an application to clinical flow cytometry. In Chapter VI, we apply our methods to accomplish both clustering and visualization to help with flow cytometry diagnosis and analysis. By characterizing each patient data set as a realization of some generative model, in which different disease classes have different characterizations, we are able to fit the problem directly into the framework we present. Each patient is viewed as a realization of some PDF lying on a statistical manifold, which enables both information based embedding with FINE/SLIM and information preserving linear projections with IPCA. We analyze these techniques for the ultimate goal of diagnosis on several different case studies which range from well defined within the pathology community to open problems.

1.4 List of Relevant Publications

The following publications were produced based on the research presented in this thesis:

(1) K. M. Carter, R. Raich, W. G. Finn and A. O. Hero. Information preserving component analysis: data projections for flow cytometry analysis. To appear in *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Digital*

Image Processing Techniques for Oncology, Feb. 2009.

(2) W. G. Finn, K. M. Carter, R. Raich, and A. O. Hero. Analysis of flow cytometric immunophenotyping data by clustering on statistical manifolds: treating flow cytometry data as high-dimensional objects. To appear in *Cytometry Part B: Clinical Cytometry*, vol. 76B, no. 1, Jan. 2009.

(3) K. M. Carter, C. Kyung-min Kim, R. Raich, and A. O. Hero III. Information preserving embeddings for discriminiation. To appear in *Proc. of IEEE Signal Processing Society DSP Workshop*, Jan. 2009.

(4) K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero III. Dimensionality reduction of flow cytometric data through information preservation. In *Proc. of IEEE Machine Learning for Signal Processing Workshop*, Oct. 2008.

(5) K. M. Carter, R. Raich, and A. O. Hero. Fine: Information embedding for document classification. In *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 1861-1864, April 2008.

(6) K. M. Carter and A. O. Hero. Variance reduction with neighborhood smoothing for local dimension estimation. In *Proc. IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3917-3920, April 2008.

(7) K. M. Carter, A. O. Hero, and R. Raich. De-biasing for intrinsic dimension estimation. In *Proc. IEEE Statistical Signal Processing Workshop*, pages 601-605, August 2007.

(8) K. M. Carter, R. Raich, and A. O. Hero. Learning on statistical manifolds for clustering and visualization. In *Proc. of 45th Annual Allerton Conf. on Communication, Control, and Computing*, pages 526-533, September 2007.

(9) K. M. Carter, R. Raich, W. G. Finn and A. O. Hero. Fine: Fisher information nonparametric embedding. In review for *IEEE Transactions on Pattern Recognition and Machine Learning*.

(10) K. M. Carter, R. Raich, and A. O. Hero. On local dimension estimation and its applications. In review for *IEEE Transactions on Signal Processing*.

(11) K. M. Carter, R. Raich, and A. O. Hero. An information geometric approach to supervised dimensionality reduction. To appear in *IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* April, 2009.

CHAPTER II

Background on Information Geometry

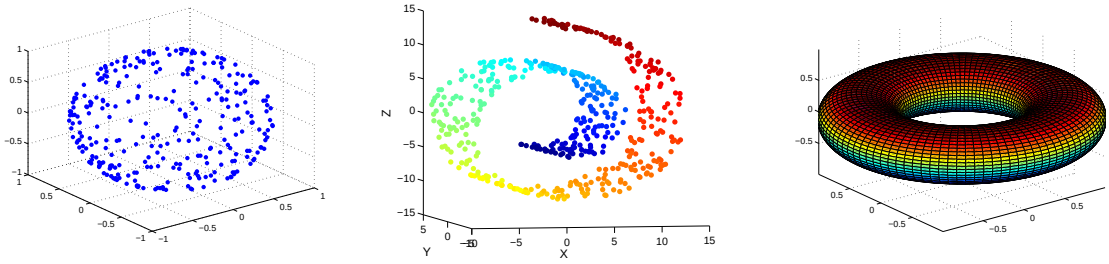
Information geometry is a field that has emerged from the study of geometrical constructs on manifolds of probability distributions. These investigations analyze probability distributions as geometrical structures in a Riemannian space. Using tools and methods deriving from differential geometry, information geometry is applicable to information theory, probability theory, and statistics. The field of information theory is largely based on the works of Shun'ichi Amari [2] and has been used for analysis in such fields as statistical inference, neural networks, and control systems. In this chapter, we will give a brief background on the methods of information geometry utilized throughout the rest of this thesis. For a more thorough introduction to information geometry, we suggest [3, 49].

2.1 Differential Manifolds

The concept of a differential manifold is similar to that of a smooth curve or surface lying in a high-dimensional space. A manifold $\mathcal{M}$ can be intuitively thought of as a set of points with a coordinate system. These points can be from a variety of constructs, such as Euclidean coordinates, linear systems, images, or probability distributions. Regardless of the definition of the points on the manifold $\mathcal{M}$, there exists a coordinate system with a one-to-one mapping from $\mathcal{M}$ to $\mathbb{R}^d$, hence d is

known as the dimension of $\mathcal{M}$.

For reference, we will refer to the coordinate system on $\mathcal{M}$ as $\psi : \mathcal{M} \to \mathbb{R}^d$. If ψ has $\mathcal{M}$ as its domain, we call it a global coordinate system [3]. In this situation, ψ is a one-to-one mapping onto $\mathbb{R}^d$ for all points in $\mathcal{M}$. A manifold is differentiable if the coordinate system mapping ψ is differentiable over its entire domain. If ψ is infinitely differentiable, the manifold is said to be 'smooth' [49].



(a) Sphere (b) Swiss Roll (c) Torus

Figure 2.1: Examples of manifolds in which no global coordinate system exists.

In many cases there does not exist a global coordinate system. Examples of such manifolds include the surface of a sphere, the "swiss roll", and the torus (see Fig. 2.1). For these manifolds, there are only local coordinate systems. Intuitively, a local coordinate system acts as a global coordinate system for a local neighborhood of the manifold, and there may be many local coordinate systems for a particular manifold. Fortunately, since a local coordinate system contains the same properties as a global coordinate system (only on a local level), analysis is consistent between the two. As such, we shall focus solely on manifolds with a global coordinate system.

2.1.1 Statistical Manifolds

Let us now present the notion of statistical manifolds, or a set $\mathcal{M}$ whose elements are probability distributions. A probability density function (PDF) on a set $\mathcal{X}$ is

defined as a function $p : \mathcal{X} \to \mathbb{R}$ in which

$$p(x) \geqslant 0, \forall x \in \mathcal{X}$$
$$\int p(x)\, dx = 1. \tag{2.1}$$

We describe only the case for continuum on the set $\mathcal{X}$, however if $\mathcal{X}$ was discrete valued, equation (2.1) will still apply by switching $\int p(x)\, dx = 1$ with $\sum p(x) = 1$. If we consider $\mathcal{M}$ to be a family of PDFs on the set $\mathcal{X}$, in which each element of $\mathcal{M}$ is a PDF which can be parameterized by $\theta = \left[\theta^1, \ldots, \theta^d\right]$, then $\mathcal{M}$ is known as a statistical model on $\mathcal{X}$. Specifically, let

$$\mathcal{M} = \{p(x \mid \theta) \mid \theta \in \Theta \subseteq \mathbb{R}^d\}, \tag{2.2}$$

with $p(x \mid \theta)$ satisfying the equations in (2.1). Additionally, there exists a one-to-one mapping between θ and $p(x \mid \theta)$.

Given certain properties of the parameterization of $\mathcal{M}$, such as differentiability and C^∞ diffeomorphism (details of which are described in [3]), the parameterization θ is also a coordinate system of $\mathcal{M}$. In this case, $\mathcal{M}$ is known as a statistical manifold. For the remainder of this thesis, we will use the terms 'manifold' and 'statistical manifold' interchangeably. When referring to standard Euclidean differentiable manifolds, we will make this clear.

2.2 Distances on Manifolds

2.2.1 Euclidean Distance

In Euclidean space, the distance between two points x and y is defined as the length of a straight line between the points and is calculated with the L_2-norm

$$D(x, y) = \|x - y\|.$$

On a manifold, however, one can measure distance by a trace of the shortest path between the points along the manifold. This path is called a geodesic, and the length of the path is the geodesic distance. In local regions about the manifold the strict Euclidean distance converges to the geodesic distance as the radius of the region decreases. Given 'far' points on a well-sampled Euclidean manifold, the geodesic may be approximated through graphical methods [7].

2.2.2 Fisher Information Distance

The Fisher information metric measures the amount of information a random variable X contains in reference to an unknown parameter θ. For the single parameter case it is defined as

$$\mathcal{I}(\theta) = E\left[\left(\frac{\partial}{\partial\theta}\log f(X;\theta)\right)^2 |\theta\right].$$

If the condition $\int \frac{\partial^2}{\partial\theta^2} f(X;\theta)\, dX = 0$ is met, then the above equation can be written as

$$\mathcal{I}(\theta) = -E\left[\frac{\partial^2}{\partial\theta^2}\log f(X;\theta)\right].$$

For the case of multiple parameters $\theta = \left[\theta^1, \ldots, \theta^d\right]$, we define the Fisher information matrix $[\mathcal{I}(\theta)]$, whose elements consist of the Fisher information with respect to specified parameters, as

$$[\mathcal{I}(\theta)]_{ij} = \int f(X;\theta)\frac{\partial \log f(X;\theta)}{\partial\theta^i}\frac{\partial \log f(X;\theta)}{\partial\theta^j}\, dX. \tag{2.3}$$

For a parametric family of probability distributions, it is possible to define a Riemannian metric using the Fisher information matrix, known as the information metric. The information metric distance, or Fisher information distance, between two distributions $p(x;\theta_1)$ and $p(x;\theta_2)$ in a single parameter family is

$$D_F(\theta_1,\theta_2) = \int_{\theta_1}^{\theta_2} \mathcal{I}(\theta)^{1/2} d\theta, \tag{2.4}$$

where θ_1 and θ_2 are parameter values corresponding to the two PDFs and $\mathcal{I}(\theta)$ is the Fisher information for the parameter θ. Extending to the multi-parameter case, we obtain:

$$D_F(\theta_1, \theta_2) = \min_{\substack{\theta(\cdot): \\ \theta(0)=\theta_1 \\ \theta(1)=\theta_2}} \int_0^1 \sqrt{\left(\frac{d\theta}{dt}\right)^T [\mathcal{I}(\theta)] \left(\frac{d\theta}{dt}\right)}\, dt. \tag{2.5}$$

where $\theta = \theta(t)$ is the parameter path along the manifold. Note that the coordinate system of a statistical manifold is the same as the parameterization of the PDFs (i.e. θ). Essentially, (2.5) amounts to finding the length of the shortest path – the geodesic – on $\mathcal{M}$ connecting coordinates θ_1 and θ_2.

Example

Here we present a derivation of a geodesic distance between univariate Gaussian densities via the Fisher information metric for two reasons. First, we would like to illustrate how involved the process is for such a simple family of PDFs. Secondly, we present a process of deriving the Fisher information metric that is involved in computing the geodesic distance. Let us consider the family of univariate Gaussian distributions $\mathcal{P} = \{p_1, \ldots, p_N\}$, such that

$$p_i(x) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-(x-\mu_i)^2/2\sigma_i^2\right).$$

where (μ_i, σ_i) is respectively the mean and standard deviation of distribution p_i.

For the case of $\mathcal{P}$ parameterized by $\theta = \left(\frac{\mu}{\sqrt{2}}, \sigma\right)$, the resultant Fisher information matrix is

$$[\mathcal{I}(\theta)] = \begin{pmatrix} \frac{2}{\sigma^2} & 0 \\ 0 & \frac{2}{\sigma^2} \end{pmatrix}.$$

We omit the derivation, which can be found in [49] and is straight forward from (2.3).

We define the distance between two points on the manifold as the minimum length between all paths connecting the two points. Using the inner product associated with the Fisher information matrix

$$< \boldsymbol{u}, \boldsymbol{v} >_F = \boldsymbol{u}^T [\mathcal{I}(\theta)] \boldsymbol{v},$$

we define the length of the path P between two points parameterized by θ_1 and θ_2, on the manifold $\mathcal{M}$ as

$$\|\theta_1 - \theta_2\|_P = \sqrt{< \theta_1 - \theta_2, \theta_1 - \theta_2 >_F}.$$

Using the parameterization $\theta(t)$ such that $\theta(0) = \theta_1$ and $\theta(1) = \theta_2$, we obtain the length of P as

$$\|\theta_1 - \theta_2\|_P = \int_0^1 \sqrt{\left(\frac{d}{dt}\theta(t)\right)^T [\mathcal{I}(\theta(t))] \left(\frac{d}{dt}\theta(t)\right)}\, dt.$$

We are able to define the distance between points $p_1 = p(x; \theta_1)$ and $p_2 = p(x; \theta_2)$ as the minimum over all path lengths defined above,

$$D_F(p_1, p_2) = \min_{\theta(t)} \sqrt{2} \int_0^1 \sqrt{\frac{\frac{1}{\sqrt{2}}\dot{\mu}^2 + \dot{\sigma}^2}{\sigma(t)^2}}\, dt, \tag{2.6}$$

where $\dot{\mu} = \frac{d}{dt}\mu(t)$ and $\dot{\sigma} = \frac{d}{dt}\sigma(t)$.

The solution to (2.6) is the well known Poincaré hyperbolic distance, in which the shortest path between two points is the length of an arc on a circle in which both points are at a radius length from the circle's center. In the case of the univariate normal distribution, this arc is a straight line when the mean is held constant and the variance is changed.

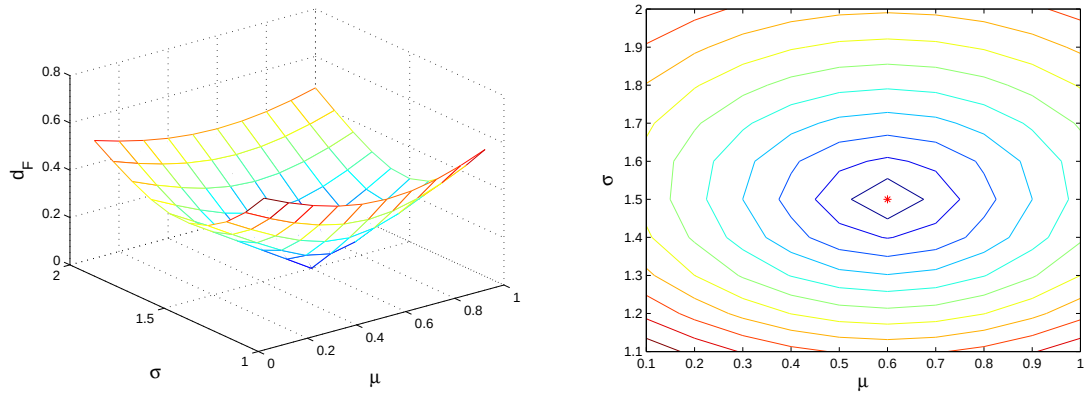
By changing variables and parameterizing σ as a function of μ, we obtain:

$$D_F(p_1, p_2) = \min_{\substack{\sigma(\mu): \\ \sigma(\mu_1)=\sigma_1 \\ \sigma(\mu_1)=\sigma_2}} \int_{\mu_1}^{\mu_2} \sqrt{\frac{1 + \dot{\sigma}^2}{\sigma(\mu)^2}}\, d\mu,$$

where $\dot{\sigma} = \frac{d}{d\mu}\sigma(\mu)$. It should be clear that this is a representation of (2.4). It should also be noted that there exists a one-to-one mapping $\sigma(\mu) : \mathbb{R} \to \mathbb{R}^+$ along the geodesic from $\sigma(\mu_1)$ to $\sigma(\mu_2)$, except for the case when $\mu_1 = \mu_2$.

Solving (2.6) becomes a problem of calculus of variations. For the univariate normal family of distributions, this has been calculated in a closed-form expression presented in [26], determining the Fisher information distance as:

$$D_F(p_1, p_2) = \sqrt{2}\log \frac{\left\|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2\right)\right\| + \left\|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2\right)\right\|}{\left\|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2\right)\right\| - \left\|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2\right)\right\|}. \tag{2.7}$$



(a) Mesh-grid (b) Contour plot

Figure 2.2: The Fisher information distance based on a grid of univariate normal densities, parameterized by (μ, σ). The reference point, p_i, is located at $(\mu_i, \sigma_i) = (0.6, 1.5)$ and is denoted by $\star$.

For visualization, let us define a set of probability densities $\mathcal{P} = \{p_i(x)\}$ on a grid, such that $p_i = p_{k,l}$ is parameterized by $(\mu_i, \sigma_i) = (\alpha k, 1 + \beta l)$, $k, l = 1 \ldots N$ and $\alpha, \beta \in \mathbb{R}$. Figure 2.2 shows a mesh-grid and contour plot of the Fisher information distance between the density defined by $(\mu_i, \sigma_i) = (0.6, 1.5)$ and the neighboring densities on the set $\mathcal{P}$ ($\alpha = \beta = 0.1$).

2.3 Approximation of Fisher Information Distance

The Fisher information distance is a consistent metric, regardless of the parameterization of the manifold [73]. This fact enables the approximation of the information distance when the specific parameterization of the manifold is unknown, and there have been many metrics developed for this approximation. An important class of such divergences is known as the f-divergence [28], in which $f(u)$ is a convex function on $u > 0$ and

$$D_f(p\|q) = \int p(x) f\left(\frac{q(x)}{p(x)}\right) dx.$$

A specific and important example of the f-divergence is the α-divergence, where $D^{(\alpha)} = D_{f^{(\alpha)}}$ for a real number α. The function $f^{(\alpha)}(u)$ is defined as

$$f^{(\alpha)}(u) = \begin{cases} \frac{4}{1-\alpha^2}\left(1 - u^{(1+\alpha)/2}\right) & \alpha \neq \pm 1 \\ u \log u & \alpha = 1 \\ -\log u & \alpha = -1 \end{cases}.$$

As such, the α-divergence can be evaluated as

$$D^{(\alpha)}(p\|q) = \frac{4}{1-\alpha^2}\left(1 - \int p(x)^{\frac{1-\alpha}{2}} q(x)^{\frac{1+\alpha}{2}} dx\right) \quad \alpha \neq \pm 1,$$

and

$$D^{(-1)}(p\|q) = D^{(1)}(q\|p) = \int p(x) \log \frac{p(x)}{q(x)} dx. \tag{2.8}$$

The α-divergence is the basis for many important and well known divergence metrics, such as the Kullback-Leibler divergence and the Hellinger distance.

2.3.1 Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence is defined as

$$KL(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx, \tag{2.9}$$

which is equal to $D^{(-1)}$ (2.8). The KL-divergence is a very important metric in information theory, and is commonly referred to as the relative entropy of one PDF to another. Kass and Vos show [49] the relation between the Kullback-Leibler divergence and the Fisher information distance is

$$\sqrt{2KL(p\|q)} \to D_F(p,q)$$

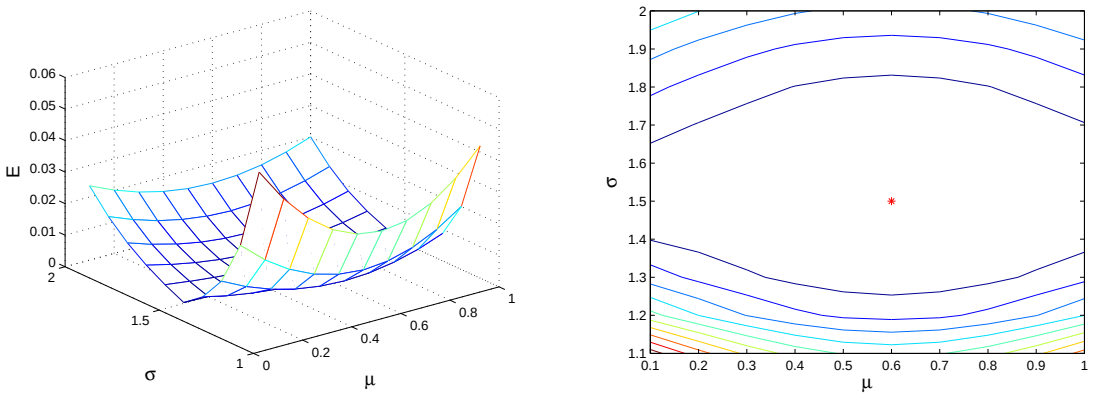
as $p \to q$. This allows for an approximation of the Fisher information distance, through the use of the available PDFs, without the need for the specific parameterization of the manifold. We approach the case where p and q are far apart, in which case the approximation becomes weak, in Chapter IV.

Returning to our illustration developed in Section 2.2.2, we have defined the data set $\mathcal{P}$ of univariate normal distributions, and presented an expression for the Fisher information distance on the resultant manifold (2.7). The Kullback-Leibler divergence between univariate normal distributions is also available in a closed-form expression:

$$KL(p_i\|p_j) = \frac{1}{2}\left(\log\left(\frac{\sigma_j^2}{\sigma_i^2}\right) + \frac{\sigma_i^2}{\sigma_j^2} + \left(\mu_j - \mu_i\right)^2/\sigma_j^2 - 1\right).$$

To compare the KL-divergence to the Fisher information distance, we define the error as $E = \left|\sqrt{2KL(p_i\|p_j)} - D_F(p_i,p_j)\right|$, where $p_{i,j} \in \mathcal{P}$. In Fig. 2.3 we display the mesh-grid and contour plots of E, where point p_i is held constant in the center of the grid defining $\mathcal{P}$, and p_j varies about the manifold. As described earlier, as the density $p_j \to p_i$, the error $E \to 0$. In Fig. 2.3(b), the reference point p_i is noted by the red star.

It should be noted that the KL-divergence is not a distance metric, as it does not satisfy the symmetry, $KL(p\|q) \neq KL(p\|q)$, or triangle inequality properties of a distance metric. To obtain symmetry, we will define the symmetric KL-divergence



(a) Mesh-grid

(b) Contour plot

Figure 2.3: The error between the KL-divergence and the Fisher information distance based on a grid of univariate normal densities, parameterized by (μ, σ). Note that $\sqrt{2KL} \to D_F$, where p_i is denoted by $\star$.

as:

$$D_{KL}(p, q) = KL(p\|q) + KL(q\|p), \tag{2.10}$$

which is symmetric, but still not a distance as it does not satisfy the triangle inequality. Since the Fisher information distance is a symmetric measure, we can relate the symmetric KL-divergence and approximate the Fisher information distance as

$$\sqrt{D_{KL}(p, q)} \to D_F(p, q), \tag{2.11}$$

as $p \to q$.

2.3.2 Hellinger Distance

Another important result of the α-divergence is the evaluation with $\alpha = 0$:

$$D^{(0)}(p\|q) = 2 \int \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2 dx,$$

which is closely related to the Hellinger distance,

$$D_H = \sqrt{\frac{1}{2} D^{(0)}},$$

which satisfies the axioms of distance – symmetry and the triangle inequality. The Hellinger distance is related to the information distance in the limit by

$$2D_H(p,q) \to D_F(p,q)$$

as $p \to q$ [49]. We note that the Hellinger distance is related to the Kullback-Leibler divergence, as in the limit $\sqrt{KL(p\|q)} \to D_H(p,q)$.

2.3.3 Other Fisher Approximations

There are other metrics which approximate the Fisher information distance, such as the cosine distance. When dealing with multinomial distributions, the approximation

$$D_C(p,q) = 2\arccos \int \sqrt{p(x)\cdot q(x)}\, dx \to D_F(p,q),$$

is the natural metric on the sphere in Euclidean space.

Throughout this thesis we restrict our analysis to that of the Kullback-Leibler divergence and the Hellinger distance. The KL-divergence is a great means of differentiating shapes of continuous PDFs. Analysis of (2.9) shows that as $p(x)/q(x) \to \infty$, $KL(p\|q) \to \infty$. These properties ensure that the KL-divergence will be amplified in regions where there is a significant difference in the probability distributions. While this property is positive in some applications, it can also be very unstable for the same reasons, especially when estimating PDFs from a finite sampling. For cases in which this instability may become an issue (e.g. multinomial PDFs because of divide-by-zero issues), the Hellinger distance is the desired metric, as it is entirely bounded by $\sqrt{2}$. Note that there also exists a monotonic transformation function $\psi : D_H \to D_C$ [49]. For additional measures of probabilistic distance, some of which approximate the Fisher information distance, and a means of calculating them between data sets, we refer the reader to [80]. We provide specific details on our

implementation of these information divergences in Appendix B.

CHAPTER III

Local Dimension Estimation

3.1 Introduction

Technological advances in both sensing and media storage have allowed for the generation of massive amounts of high dimensional data and information. Consider the class of applications which generate these high dimensional signals: e.g., digital cameras capture images at enormous resolutions; dozens of video cameras may be filming the exact same object from different angles; planes randomly drop hundreds of sensors into the same area to map the terrain. While this has opened a wealth of opportunities for data analysis, the problem of the *curse of dimensionality* has become more substantial, as many learning algorithms perform poorly in high dimensions. While the data in these applications may be represented in high dimensions, strictly based upon the immense capacity for data retrieval, it is typically concentrated on a lower dimensional manifold of the measurement space. The study of these low dimensional manifolds has led to a field of machine learning called manifold learning, and has yielded such renowned work as Isomap [74], Local Linear Embedding [71], and Laplacian Eigenmaps [5], among several other dimensionality reduction methods.

The point at which the data can be reduced with minimal loss of information is

related to the *intrinsic dimensionality* of the manifold supporting the data. When the intrinsic dimension is assumed constant over the data set, several algorithms [10, 24, 50, 58] have been proposed to estimate the dimensionality of the manifold. In several problems of practical interest, however, data will exhibit varying dimensionality, as there may lie multiple manifolds of varying dimension within the data. This is easily viewed in images with different textures or in classification tasks in which data from different classes is generated by unique PDFs. In these situations, the local intrinsic dimension may be of more importance than the global dimension. In previous work [12, 23] we illustrated the process of local dimension estimation, in which a dimension estimate is obtained for each sample within the data, rather than a single dimension estimate for the entire set.

In this chapter we focus on the applications of local dimension estimation [17]. One immediate benefit is using local dimension to estimate the global dimension of a data set. To our knowledge, every method of estimating intrinsic dimension has expressed an issue with a negative bias. While insufficient sampling is a common source of this bias, a significant portion is a result of samples near the boundaries or edges of a manifold. These regions appear to be low dimensional when sampled, and contribute a strong negative bias to the global estimate of dimension. Through the use of local dimension estimation, we will show that we can significantly remove this negative bias.

We continue by showing novel applications in which the exact dimension of the data is of no immediate concern, but rather the differences between the local dimensions. Dimensionality can be viewed as the number of *degrees of freedom* in a data set, and as such may be interpreted as a measure of data *complexity*. By comparing the local dimension of samples within a data set, we are able to identify different

subsets of the data for analysis. For example, in a time series data set, the intrinsic dimensionality may change as a function of time. By viewing each time step as a sample, we can identify changes in the system at specific time points. We illustrate this ability by finding anomalous activity in a router network. Additionally, the identification of subsets within the data allows for the immediate application of clustering and image segmentation. There has been much work presented on using fractal dimension estimation for image and texture segmentation [22,65]. We do not make the model assumption that textures may be represented as a collection of fractals [59], and instead segment images using a novel method based on Euclidean dimension. We show that by using a technique we developed termed 'neighborhood smoothing' [11] over the dimension estimates, we are able to find the regions which exhibit differing complexities, and use the smoothed dimension estimates as identifiers for the clusters/segments.

The rest of this chapter proceeds as follows: We give an overview of the two dimension estimation algorithms we will utilize in our simulations in Section 3.2. In Section 3.3, we describe the process of 'neighborhood smoothing' as a means of post-processing for local dimension estimation. We illustrate the various novel applications of local dimension estimation in Section 5.4, including de-biasing for global dimension estimation, network anomaly detection, clustering, and image segmentation. Finally, we offer a discussion and present areas for future work in Section 5.5.

3.2 Dimension Estimation

We will now present two algorithms for dimension estimation, the k-NN algorithm [24, 25] and the maximum likelihood estimation method [58]. While there are many algorithms available for dimension estimation, we focus on these two as a means for

illustrating the applications we later present. By utilizing two distinct methods, we hope to quell any concerns that our applications are algorithm dependent. Note that the applications in this chapter assume Euclidean space and therefore the algorithms are developed in a Euclidean fashion by using the L_2-distance norm. However, both algorithms are easily extended to statistical manifolds by substituting the Fisher information distance for the Euclidean distance, and will be utilized as such in later chapters.

3.2.1 The k-Nearest Neighbor Algorithm for Dimension Estimation

Let $\boldsymbol{X} = [x_1, \ldots, x_n]$ be n independent and identically distributed (i.i.d.) random vectors with values in a compact subset of $\mathbb{R}^d$. The (1-)nearest neighbor of x_i in $\boldsymbol{X}$ is given by

$$\arg \min_{x \in \boldsymbol{X} \setminus \{x_i\}} D(x, x_i)$$

where $D(x, x_i)$ is an appropriate distance measure between x and x_i. For a general integer $k \geqslant 1$, the k-nearest neighbor of a point is defined in a similar way. The k-NN graph assigns an edge between each point in $\boldsymbol{X}$ and its k-nearest neighbors. Let $\mathcal{N}_{k,i} = \mathcal{N}_{k,i}(\boldsymbol{X})$ be the set of k-nearest neighbors of x_i in $\boldsymbol{X}$. The total edge length of the k-NN graph is defined as:

$$L_{\gamma,k}(\boldsymbol{X}) = \sum_{i=1}^{n} \sum_{x \in \mathcal{N}_{k,i}} D(x, x_i)^{\gamma} \ , \tag{3.1}$$

where $\gamma > 0$ is a power weighting constant; generally $\gamma = 1$ for dimension estimation.

For many data sets of interest, the random vectors $\boldsymbol{X}$ are constrained to lie on an m-dimensional Riemannian submanifold $\mathcal{M}$ of $\mathbb{R}^d$ ($m < d$). Let us define $D(x, x_i) = \|x - x_i\|$ as the standard Euclidean distance. As described in [25], w.p.1

$$\lim_{n \to \infty} \frac{L_{\gamma,k}(\boldsymbol{X})}{n^{\alpha}} = \beta_{m,\gamma,k} \int_{\mathcal{M}} f^{\alpha}(x) \, \mu_{\mathcal{M}}(dx), \tag{3.2}$$

where f is the bounded density of $[x_1, \ldots, x_n]$ relative to the differential volume element over $\mathcal{M}$ ($\mu_{\mathcal{M}}(dx)$), $\beta_{m,\gamma,k}$ is a constant independent of f, and $\alpha = \alpha(m) = (m - \gamma)/m$. Under this framework, the asymptotic behavior of (3.1) is given as:

$$L_{\gamma,k}(\boldsymbol{X}) = n^{\alpha} c + \epsilon_n \tag{3.3}$$

where c is a constant with respect to α that depends on the Rényi entropy of the distribution on the manifold, and ϵ_n is an error residual [12].

The estimate of the intrinsic dimension $\hat{m}$ can be found using a non-linear least squares solution, by calculating graph lengths over varying values of n. In order to calculate graph lengths for differing sample sizes on the manifold, it is necessary to randomly subsample from the full set $\boldsymbol{X} = [x_1, \ldots, x_n]$, utilizing the non-overlapping block bootstrapping method [54]. Specifically, let $\boldsymbol{X}'_n = [x_{(1)}, \ldots, x_{(n)}]$ be a spatially or temporally sorted version of $\boldsymbol{X}$, and let w be an integer satisfying $w < n/Q$. Define the blocks $\mathcal{B}_i = (x_{((i-1)w+1)}, \ldots, x_{(iw)})$, $i = 1, \ldots, n/w$; we may now redefine $\boldsymbol{X}' = \{\mathcal{B}_1, \ldots, \mathcal{B}_{n/w}\}$. Let $\{p_1, \ldots, p_Q\}$ be Q integers such that $1 \leqslant p_1 < \ldots, < p_Q \leqslant n/w$. For each value of $p \in \{p_1, \ldots, p_Q\}$ randomly draw N bootstrap datasets $\boldsymbol{X}_p^j$, $j = 1, \ldots, N$, with replacement, where the p blocks of data points within each $\boldsymbol{X}_p^j$ are chosen from the entire data set $\boldsymbol{X}'_n$ independently. From these samples define $\boldsymbol{L}_n = \left\{L_{\gamma,k}(\boldsymbol{X}_p^1), \ldots, L_{\gamma,k}(\boldsymbol{X}_p^N)\right\}$, where $n = p\, w$.

Since c is dependent on m, it is necessary to solve for the minimum mean squared error, derived from (3.3), by minimizing over both c and integer values of $m \in \mathbb{Z}$.

$$\hat{m} = \arg\min_{m \in \mathbb{Z}} \left\{ \min_{c} \sum_{i=1}^{Q} \left\| \boldsymbol{L}_{n_i} - n_i^{\alpha(m)} c \mathbf{1} \right\|^2 \right\}, \tag{3.4}$$

where $n_i = p_i\, w$ and $\mathbf{1}$ is the vector of length n_i whose elements are all 1. We solve over integer values of m as we do not consider fractal dimensions for this algorithm. This improves accuracy by constraining the estimation space to discrete values, rather

than discretizing estimates in a continuous space. One can solve (3.4) in the general manner presented in Appendix 3-A. This non-linear least squares solution yields the dimension estimate $\hat{m}$ based on the k-NN graphs.

3.2.2 The Maximum Likelihood Estimator for Intrinsic Dimension

The maximum likelihood estimation (MLE) method [58] for dimension estimation estimates the intrinsic dimension $\hat{m}$ from a collection of i.i.d. observations $\boldsymbol{X} = [x_1, \ldots, x_n] \in \mathbb{R}^d$. Similar to the k-NN algorithm for dimension estimation, the MLE method assumes that close neighbors lie on the same manifold. The estimator proceeds as follows, letting k be a fixed number of nearest neighbors to sample x_i:

$$\hat{m}_k(x_i) = \left[\frac{1}{k-2} \sum_{j=1}^{k-1} \log \frac{T_k(x_i)}{T_j(x_i)} \right]^{-1}, \tag{3.5}$$

where $T_k(x_i)$ is the distance from point x_i to its k-th nearest neighbor in $\boldsymbol{X}$, measured with some suitable metric. The intrinsic dimension for the data set can then be estimated as the average over all observations:

$$\hat{m}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{m}_k(x_i).$$

The complete derivation and further details of the MLE algorithm can be found in [58].

3.2.3 Local Dimension Estimation

While the MLE method inherently generates local dimension estimates for each sample, $\hat{m}(x_i)$, the k-NN algorithm in itself is a global dimension estimator. We are able to adopt it (and any other dimension estimation algorithm) as a local dimension estimator by running the algorithm over a smaller neighborhood about each sample point. Define a set of n samples $\boldsymbol{X} = [x_1, \ldots, x_n]$ from the collection of manifolds $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_M\}$ such that each point x_i lies on manifold $\mathcal{M}_j$. Any small sphere

or data cluster of samples $\mathcal{C} \subseteq \boldsymbol{X}$ centered at point x_i, with $|\mathcal{C}| = n' \leqslant n$, will contain samples from $M' \leqslant M$ distinct manifolds. As $n' \to 1$, all of the points in $\mathcal{C}$ will lie on a single manifold (i.e. $M' \to 1$). Intuitively speaking, as the cluster about point x_i is reduced in size, the local neighborhood defined by said cluster can be viewed as its own data set confined to a single manifold. Hence, we can use a global dimension estimation algorithm on a local subset of the data to estimate the local intrinsic dimension of each sample point. This can be performed as described in Algorithm 3.1, where 'dimension($\mathcal{C}$)' refers to applying any method of dimension estimation to the data cluster $\mathcal{C}$.

Algorithm 3.1. Local Dimension Estimation

Input: Data set $\boldsymbol{X} = [x_1, \ldots, x_n]$

1: **for** $i = 1$ to n **do**
2: Initialize cluster $\mathcal{C} = x_i$
3: **for** $k = 1$ to n' **do**
4: Find the k-th NN, $x_{k,i}$, of x_i
5: $\mathcal{C} \leftarrow \mathcal{C} \cup x_{k,i}$
6: **end for**
7: $\hat{m}(x_i) = \text{dimension}(\mathcal{C})$
8: **end for**

Output: Local dimension estimates $\hat{m}$

One of the keys to local dimension estimation is defining a value of n'. There must be a significant number of samples in order to obtain a proper estimate, but it is also important to keep a small sample size as to (ideally) only include samples which lie on the same manifold. Currently we arbitrarily choose n' based on the size of the data set. However, a more definitive method of choosing n' is grounds for

future work.

3.3 Neighborhood Smoothing

For the problem of local dimension estimation, results are often highly variable, where nearby samples from the same manifold may estimate at different dimensions. This issue can be a result of a variety of reasons, such as variability due to random subsampling in the k-NN algorithm, or variability due to the neighborhood size in the MLE method. When constructing a global dimension estimate, this variance is relatively insignificant as the estimate is constructed as a function of the local estimates. For local dimension estimation, however, this variance is of significant concern, and we propose a variance reduction method known as 'neighborhood smoothing' [11] which improves estimation accuracy.

An initial intuition for manifold learning algorithms is that samples that are "close" tend to lie on the same manifold, which extends to the assumption that they therefore have the same dimension. With this assumption in place, it follows that filtering by majority vote over the dimension estimates of nearby samples should smooth the estimator and reduce variance. This voting strategy is similar to the methods of mode filtering, bagging [9] and learning by rule ensembles [36]. Smoothing simply looks at the distribution of dimension estimates within each sample point's local neighborhood, and re-assigns each sample a dimension estimate equal to that with the highest probability within its neighborhood. Specifically,

$$\hat{m} = \arg\max_{l} \mathrm{P}_{\mathcal{N}_i} \left[\hat{m} = l\right], \tag{3.6}$$

where $\mathrm{P}_{\mathcal{N}_\mathrm{i}}$ is the probability over the neighborhood of the current sample $\mathcal{N}_i$. Given

a finite number of samples $[x_1, \ldots, x_n]$, this may be empirically evaluated as

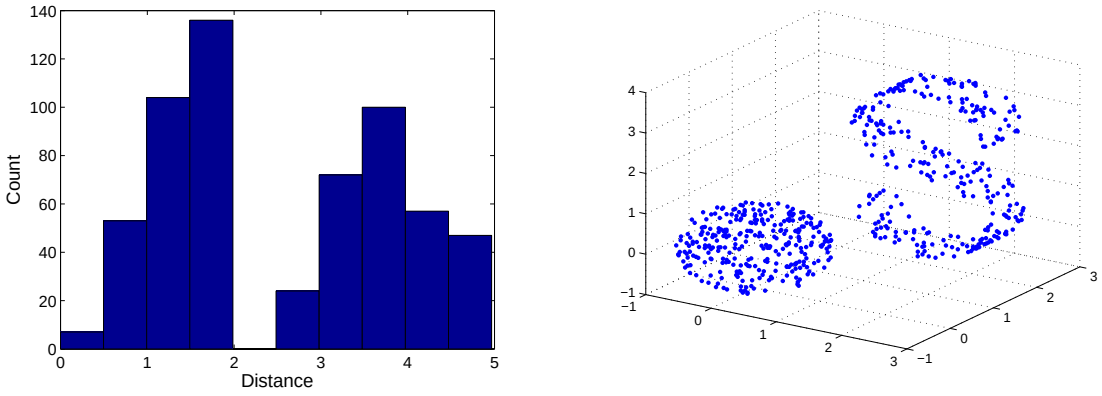
$$\hat{m}(x_i) = \arg\max_l \sum_{x_j \in \mathcal{N}_i} I(\hat{m}(x_j) = l)\,, \tag{3.7}$$

where $I(\cdot)$ is the standard indicator function. This process may then be iterated until each neighborhood converges to a consistent estimate. This has the effect of implicitly incorporating the neighbors of each sample's neighbors to some extent, as the dimension estimates within a local region may change through iterations.

Intuitively, neighborhood smoothing is similar to iteratively imposing a k-NN classifier on the local dimension estimates – under the guise that at each iteration, sample x_i is a test sample and all points x_j, $j \neq i$ are appropriately labeled training samples. Similarly to k-NN classification, the key factor to smoothing is defining the neighborhood, $\mathcal{N}_i$. If $\mathcal{N}_i$ is too large, oversmoothing will occur. The variance of the dimension estimates will drastically decrease, but there will be a strong bias which will remove the detection of coarsely sampled manifolds. As such, one cannot use a constant region about a point, but must adapt that region to the statistics of the sample.

3.3.1 Spherical Radius Selection

Since the number of sample points on each manifold of a data set is generally unknown, using a constant number of smoothing samples is not a viable option. Samples on a smaller manifold may have points from a disjoint manifold included in their smoothing neighborhood. Therefore, when using a spherical region as a smoothing neighborhood, it is important to adjust the radius of that region for each sample point. This can be done by looking at the distribution of the distances of each point from the current sample location. Often, one can quickly determine a reasonable neighborhood radius, as there tends to be a jump in the distance distri-



(a) Distance Histogram

(b) 3-D Plot of Data

Figure 3.1: Analysis of a histogram of distances from a point on the sphere suggests a spherical neighborhood radius of no larger than 2. This clearly distinguishes the distinct manifolds.

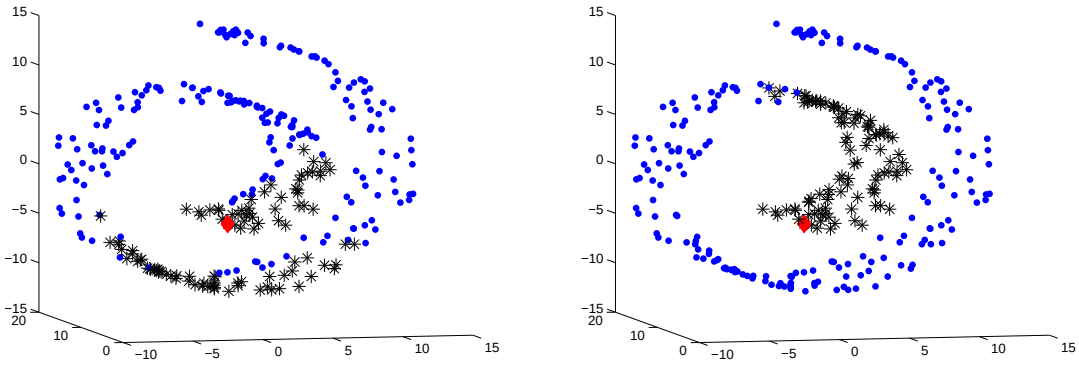
bution. This can be very apparent when there are multiple manifolds in the data set (see Fig. 3.1). When the difference is not as clear, one can decide the radius in various different ways (such as a function of the median distance).

By adjusting the spherical radius of the smoothing neighborhood for each sample point, we are efficiently adapting the smoothing algorithm to, ideally, smooth only over samples which lie on the same manifold. This method is feasible only when manifolds are disconnected and somewhat distant from one another.

3.3.2 Non-Spherical Neighborhoods

When distinct manifolds lie near one another, or potentially intersect, it is necessary to further adapt the smoothing neighborhood beyond a spherical region. This is due to the fact that points on a nearby or intersecting manifold may be as close (or closer) to a sample as others on its own manifold. A spherical region may smooth over different manifolds, and the results will lead to the dimension estimates 'leaking' from one manifold to another.

Rather than defining neighborhoods through Euclidean distance, which will form



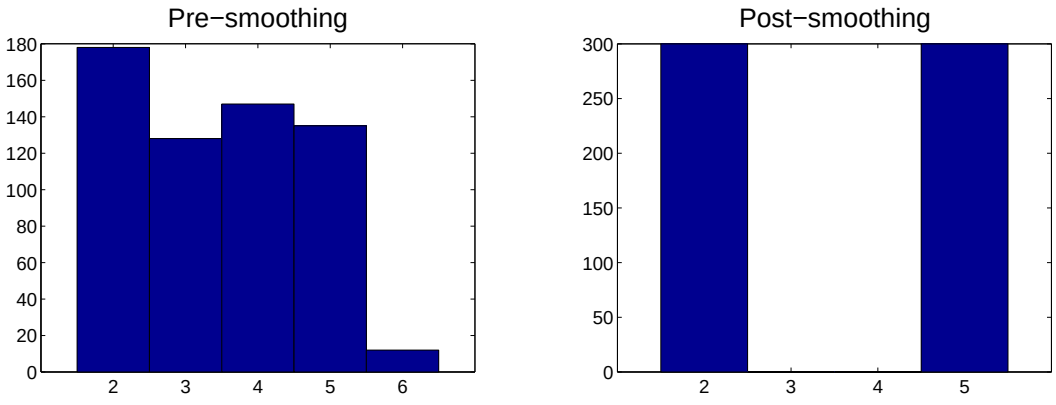
(a) Spherical Neighborhood

(b) Adaptive Neighborhood

Figure 3.2: Neighborhoods ($\star$) of the sample in question ($\diamond$) defined by a) Euclidean distance and b) geodesic distance.

only spherical regions about each sample point, we will define neighborhoods using a geodesic distance metric. This will adapt the neighborhood to the geometry of the manifold. For our purposes, the geodesic distance can be approximated by taking each point and creating an edge to its the k-NN. Then using Dijkstra's shortest path algorithm (or any other algorithm for computing the shortest path), approximate the geodesic distances to each pair of points in the graph. Any points that remain unconnected are considered to have an infinite geodesic distance.

To define a local neighborhood, we can now simply choose the closest n_g points for which the geodesic distance is not infinite. This forms a non-spherical neighborhood that adapts to the curvature of the manifold, performing much better than spherical neighborhoods. Figure 3.2 illustrates the difference in the neighborhoods (black stars) that are formed on the 'swiss roll' manifold when using different proximity metrics. The Euclidean distance (Fig. 3.2(a)) forms a spherical neighborhood, including points that are separated from the sample in question (red diamond). The geodesic distance (Fig. 3.2(b)), however, forms a neighborhood considering points only in close proximity along the actual manifold. While all points in this exam-



(a) (b)

Figure 3.3: Neighborhood smoothing applied to 7-dimensional data containing two spheres with intrinsic dimensions 2 and 5

ple do exist on the same manifold, it is clear that defining neighborhoods along the manifold rather than in simple spherical regions reduces the probability of including samples from a nearby distinct manifold.

Illustrating the effects of neighborhood smoothing, we create a 7-dimensional data set that includes 2 distinct spheres of intrinsic dimensions 2 and 5, each containing 300 uniformly sampled points intersecting in 3 common dimensions. Figure 3.3(a) shows the histogram of the local dimension estimates of each sample before any neighborhood smoothing was applied, while Fig. 3.3(b) shows the results after smoothing. One can clearly see that the wide histogram was correctly condensed to the proper local dimension estimates, even though the manifolds intersect. The use of the geodesic distance measure prevents smoothing across distinct manifolds which lie closely together in Euclidean space.

It is important to note that, as with any form of post-processing, neighborhood smoothing can only produce accurate results given sufficient input. The benefits of smoothing can be significantly diminished if the initial local dimension estimates are not sufficiently accurate. We note this explicitly because of the known issues

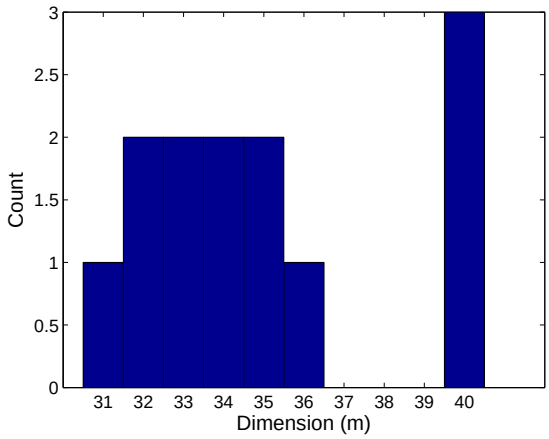


Figure 3.4: Issues arise with neighborhood smoothing when estimating very large dimensions, due to the variance of such estimates. In this example smoothing would assign a dimension estimate of 40, although the more appropriate estimate would be 33 or 34.

with estimating large dimensions (e.g. $m > 10$). Because of variance issues due to insufficient samples and boundary effects, it is difficult to accurately estimate very large dimensions, and often times the estimate can more appropriately be considered a measure of *complexity*, where the difference between m and $m + 1$ is rather insignificant. This is important because no single dimension may dominate a given local neighborhood, yet smoothing will still assign a dimension estimate equal to the most represented dimension, which may indeed be inconsistent with the rest. We demonstrate this scenario with the example shown in Fig. 3.4, where smoothing would assign a dimension estimate of $m = 40$, which is the most represented dimension in the neighborhood. However, a more accurate dimension estimate could be considered $m = 33$ or $m = 34$, as that would be more consistent with the majority of the samples. In these scenarios it may be more appropriate to smooth over a histogram with user defined bin sizes, corresponding to significant differences in complexity, rather than individual dimensions. This is an area for future work.

3.4 Applications

3.4.1 De-biasing Global Dimension Estimation

To our knowledge, a phenomenon common to all algorithms of intrinsic dimension estimation is a negative bias in the dimension estimate. It is believed that this is an effect of undersampling the high dimensional manifold. While the bias due to lack of sufficient samples is inherent, we offer that the sample size is not the only source of bias; a significant portion is related to the depth of the data. Specifically, as data samples approach the boundaries of the manifold, they exhibit a lower intrinsic dimension. This issue becomes more prevalent as the dimension of the manifold increases, and is directly related to the *curse of dimensionality*. Consider the m-dimensional unit hypercube $\mathcal{A} = [0,1]^m$. One can define the interior as the set $\mathcal{I} = \{x \mid \frac{\epsilon}{2} \leqslant x_i \leqslant 1 - \frac{\epsilon}{2}, \forall i = [1,m]\}$. The ϵ-boundary is therefore $\partial\mathcal{A} = \mathcal{A}/\mathcal{I}$. The following statement can be made:

Proposition 3.1. *With probability of at least $1-\delta$, a uniformly selected x from $\mathcal{A}$ is contained in the boundary $\partial\mathcal{A}$, i.e., $x \in \partial\mathcal{A}$ and $\epsilon = \frac{\log(1/\delta)}{m}$.*

Proof. Since x is uniform in $\mathcal{A}$, its components are *i.i.d.* uniform random variables $U[0,1]$. The probability of x being in the interior $\mathcal{I}$ is therefore given by the product

$$P(x \in \mathcal{I}) = \prod_{i=1}^{m} P\left(\frac{\epsilon}{2} \leqslant x_i \leqslant 1 - \frac{\epsilon}{2}\right) = (1-\epsilon)^m.$$

Therefore, the probability of $x \in \partial\mathcal{A}$ is

$$\begin{aligned} P(x \in \partial\mathcal{A}) &= 1 - (1-\epsilon)^m \\ &= 1 - \exp(m\log(1-\epsilon)). \end{aligned}$$

Since $\log(1+t) \leqslant t$, we have $\exp(m\log(1-\epsilon)) \leqslant \exp(-m\epsilon)$ and therefore

$$P(x \in \partial\mathcal{A}) \geqslant 1 - \exp(-m\epsilon).$$

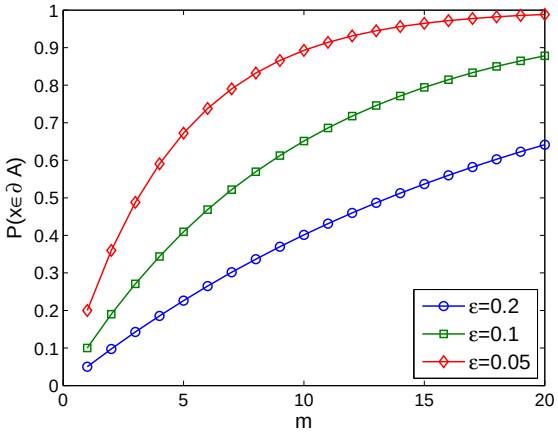


Figure 3.5: The probability of randomly selecting a point on the boundary of an m-dimensional hypercube for $\epsilon = 0.2$ ($\times$), $\epsilon = 0.1$ ($\circ$), and $\epsilon = 0.05$ ($\diamond$).

For $\epsilon = \frac{\log(1/\delta)}{m}$, we have

$$P(x \in \partial\mathcal{A}) \geqslant 1 - \exp\left(-m\frac{\log(1/\delta)}{m}\right) = 1 - \delta.$$

□

This result suggests that at least $1 - \delta$ of the entire points in the hypercube are concentrated in a boundary with $\epsilon \to 0$ as $m \to \infty$. Alternatively, for large m most points in a hypercube will concentrate on its boundary (see Fig. 3.5).

We proceed by suggesting that the boundary of the m-dimensional hypercube can be approximated as an $(m - c)$-dimensional manifold, where c is the number of connected boundaries (i.e. edges and corners), and hence should produce a lower dimension estimate. Clearly, a simple average of the local dimension estimates over the manifold will consider many more points $(1 - \delta)$ on a boundary with a lower dimension as compared with the number of points in the interior of the hypercube (δ), leading to a lower global dimension estimate.

We are able to further justify the effect of data depth on dimension estimation by calculating the depth of each sample and quantitatively analyzing the relationship

between depth and dimension. We utilize the L_1-data depth algorithm developed by Vardi and Zhang [77], which calculates depth $D_n(x)$ as the sum of all the unit vectors between the interested sample $x \in \boldsymbol{X}$ and the rest of the data set $\boldsymbol{X} = \{x_1, \ldots, x_n\} \backslash \{x\}$. Specifically,

$$D_n(x) = 1 - \max\Big(0, \Big\| \sum_{x_i \neq x} e(x_i - x)/n \Big\| - \sum_{x_i = x} \frac{1}{n}\Big) \tag{3.8}$$

where $e(x_i - x) = (x_i - x)/\|x_i - x\|$ is the unit vector in the direction of $(x_i - x)$. This depth metric assigns the most interior points in the data set a depth value approaching 1, while samples along the boundaries approach a depth of 0.

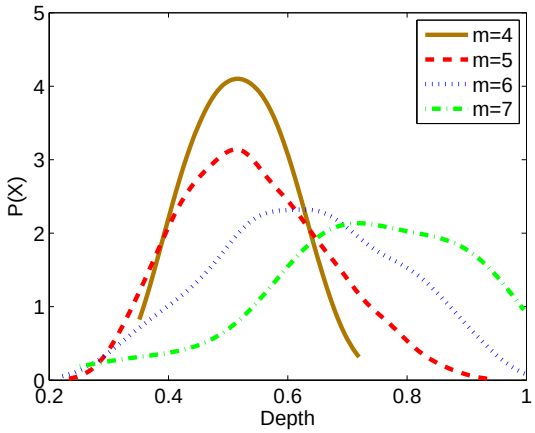


Figure 3.6: Analysis of the effect of data depth on local dimension estimation. Points with less depth estimate at a lower dimension, contributing to the overall negative bias.

Using this measure, we illustrate the effect of data depth on dimension estimation in Fig. 3.6. The data set used was of 3000 points uniformly sampled on a 6-dimensional hypercube. We utilize the MLE method for dimension estimation, and Fig. 3.6 illustrates the distribution of data depths for samples that estimate at different dimensions. It is clear that as the depth increased, so did the probability of estimating at a higher dimension, even to the point where the most deep points estimated at a dimension of 7 (although we note that there were very few points with this estimate).

When estimating the global dimension of a data set, one can substantially reduce the negative bias by placing more emphasis on the local dimension of those points away from the boundaries, as they are more indicative of the true dimension of the manifold. Specifically, let the global dimension be estimated as follows:

$$\hat{m} = \frac{1}{\sum_j W_j} \sum_i W_i \hat{m}(x_i), \tag{3.9}$$

where W_i is a weighting on each sample point. We offer two potential definitions of W_i, the first being a binary weighting:

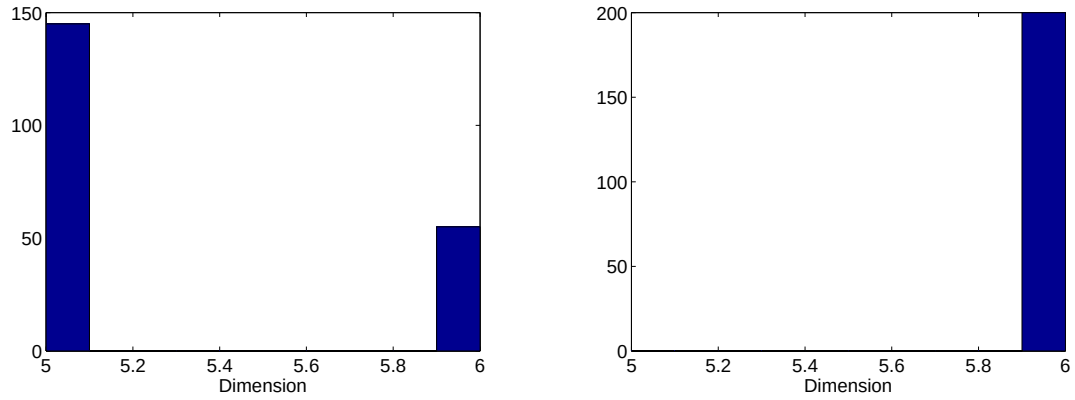
$$W_i = \begin{cases} 1, & D_n(x_i) \geqslant D_n(x_{(\alpha \times n)}) \\ 0, & \text{otherwise} \end{cases}, \tag{3.10}$$

where $0 \leqslant \alpha \leqslant 1$ and $D_n(x_{(\alpha \times n)})$ is the data depth of the $\alpha \times n$ deepest point. Essentially this binary weight amounts to de-biasing by averaging over the local dimension estimates of the deepest $\alpha \times 100\%$ of points, where the threshold α is user defined. This is worthwhile for potentially large data sets, where there are enough samples to ignore a large portion of them. When this is not the case, let us make the definition

$$W_i = \exp -(1 - D_n(x_i))/c, \tag{3.11}$$

where c is a user defined constant. This weighting may be viewed as a heat kernel, in which larger depths will yield higher weights. Unlike the binary weighting, which will ignore a large number of the data samples, this heat kernel weighting will utilize all samples (even those lying on a boundary), yet give preference to those with more depth in the manifold.

We now illustrate this de-biasing ability in Fig. 3.7, in which we estimated the global dimension of the 6-dimensional hypercube (3000 i.i.d. samples) over 200 unique



(a) Biased Results

(b) De-biased Results

Figure 3.7: Developing a de-biased global dimension estimate by averaging over the 50% of points with the greatest depth on the manifold

trials. Figure 3.7(a) shows the histogram of biased dimension estimates obtained by using the entire set for dimension estimation, while Fig. 3.7(b) estimates the correct global dimension each trial by using our de-biasing method (3.9) with the binary weighting function (3.10). We chose to set our threshold at $\alpha = 0.5$ by setting our boundary limit at $\epsilon = 0.1$. As previously calculated in Proposition 3.1 and shown in Fig. 3.5, the probability of a 6-dimensional point lying away from such a boundary is roughly 0.5. Although this suggests *a priori* knowledge of the true dimension, this knowledge is easily inferred from biased results as well; an estimate of $\hat{m} = 5$ would imply including the deepest $\sim 60\%$ of the points, and there is minimal consequence for including slightly fewer points in the final approximation.

It is important to note that our method of de-biasing is only applicable for data with a relatively low intrinsic dimension. When dealing with very high dimensional data, the probability of a sample lying near a boundary approaches 1 (see Fig. 3.5), and the value of the depth approximation becomes irrelevant. This is shown in Fig. 3.8 where the 'deepest' and most 'shallow' samples converge to the same depth value as the intrinsic dimension increases. When all samples estimate near the same depth,

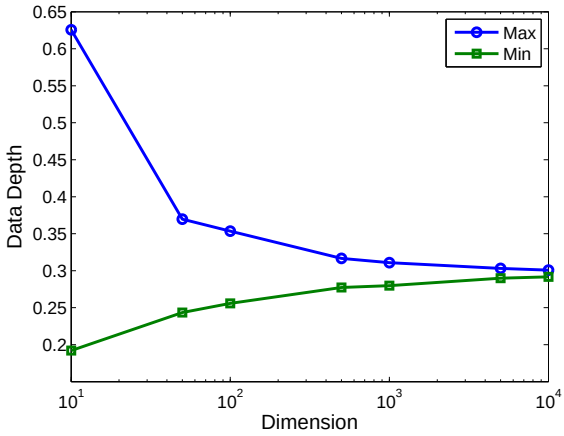


Figure 3.8: As the intrinsic dimension dimension increases, the maximum and minimum data depth of points in the set converge to the same value. This simulation was over a 5-fold cross-validation with 400 uniformly distributed points in the range [0,1].

it is clear that de-biasing based on depth will not have the intended effect.

3.4.2 Network Anomaly Detection

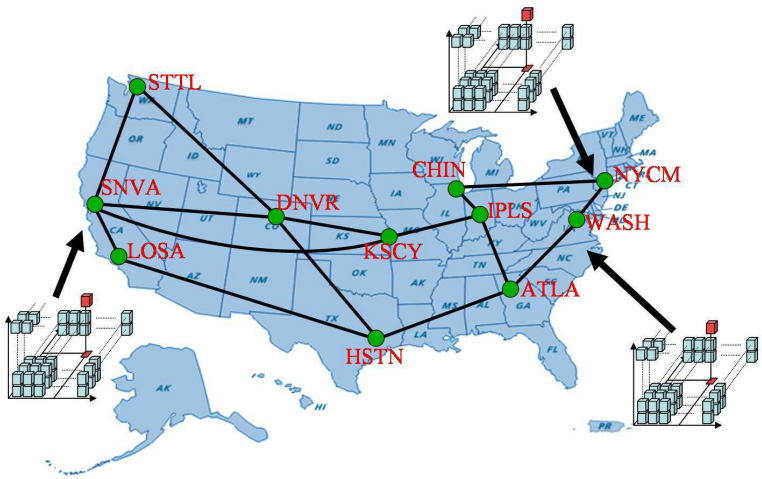
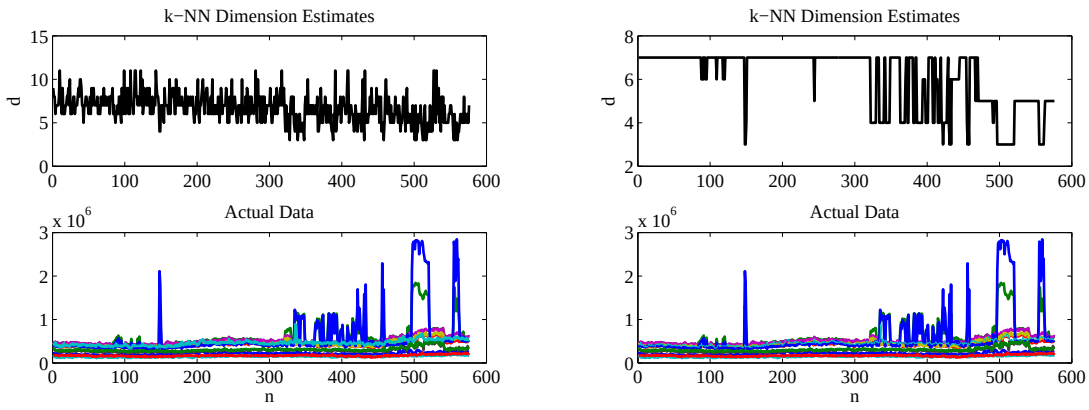


Figure 3.9: Map of Abilene router network

As illustrated in Figure 3.9, the Abilene Network is the set of routers which is the backbone of the '.edu' network. When an anomaly occurs on the network, there are changes in the correlation between traffic traces at different points in the network, imposing nonlinear constraints on the observed data. We have shown that anomalies can be detected in router networks through the use of local dimension estimation



(a) Before Smoothing

(b) After Smoothing

Figure 3.10: Neighborhood smoothing applied to Abilene Network traffic data dimension estimation results. Anomalous activity is preserved and more easily observed.

[11, 12]. Specifically, when only a few of the routers contribute disproportionably large amounts of traffic, the intrinsic dimension of the entire network decreases. Using neighborhood smoothing as a form of post-processing, we are better able to locate the traffic anomalies, as the variance of the estimates is reduced. Fig. 3.10 illustrates the results of k-NN algorithm for local dimension estimation for the purposes of anomaly detection. The data used is the number of packets counted on each of the 11 routers on the Abilene network, on January 1-2, 2005. Each sample is taken every 5 minutes, leading to 576 samples with an extrinsic dimension of $d = 11$.

Figure 3.10(b) illustrates that neighborhood smoothing is able to preserve both the visually obvious ($n = 148$, $n > 300$) and non-obvious ($n = 87 - 120$) changes in network complexity. A detailed investigation of time $n = 244$, for example, reveals that the Sunnyvale router (SNVA) showed increased contribution from a single IP address. Large percentages (over half) of the overall packets had both source and destination IP 128.223.216.xxx within port 119. This port showed increased activity on the Atlanta router as well. This change in dimensionality indicating anomalous activity would generally go unnoticed with the raw results of local dimension esti-

mation due to the high variance (Fig. 3.10(a)).

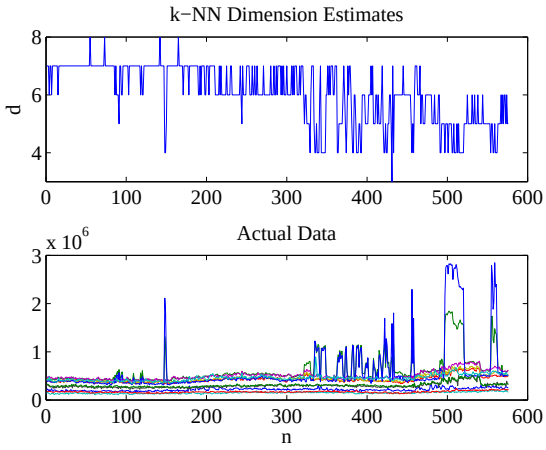


Figure 3.11: Using the k-NN algorithm with fully functional settings and no neighborhood smoothing still yields highly variable results on the Abilene data.

We note the results shown in Fig. 3.10 are performed using nominal settings within the k-NN algorithm which allows the algorithm to run quickly and accurately with neighborhood smoothing. We are able to generate results with much less variance than Fig. 3.10(a) by applying more averaging and bootstrapping as shown in Fig. 3.11, but this increases computation time by multiple orders of magnitude while still producing results with more variance than Fig. 3.10(b).

3.4.3 Clustering

As discussed previously, data sets often consist of multiple submanifolds of differing dimension. When the intrinsic dimension of these submanifolds becomes increasingly large, the value of the dimension may be interpreted as a measure of the *complexity* of the data. From this interpretation, we may use local dimension estimation to cluster data within a set by complexity. Specifically, we can define clusters through the use of recursive entropy estimation and neighborhood smoothing. As we increase the neighborhood size k, we incorporate more samples into our smoothing region, eventually oversmoothing between differing manifolds. By finding the point

in which the smoothing regions extend into multiple manifolds, we can define clusters in the data. This point of change can be located by analyzing the change in the entropy H of the dimension estimates as the region grows, such that

$$H = -\sum_j P_j \log P_j,$$

where $P_j = \frac{1}{n}\sum_i^n I(\hat{m}(i) = j)$ is the empirical probability a sample estimates at dimension j.

When the regions are stable within each cluster, H will be constant. As the smoothing neighborhood incorporates additional manifolds, the entropy will leave its constant state and eventually $H \to 0$ as $k \to \infty$ (i.e. the region includes every point). With *a priori* knowledge of the distribution of dimensionality, one may choose a neighborhood size which yields an appropriate value of entropy. Without this knowledge, the point at which H leaves its constant state can be used as a threshold for defining clusters based on dimension. This process is similar to dual-rooted diffusion kernels method of clustering [39], in which the authors used the jump in nearest neighbor distance as as means to differentiate clusters.

For example, let $\boldsymbol{X} = [x_1, \ldots, x_n]$, where $x_i \in \mathbb{R}^d$ is uniformly distributed in $[0,1]^{m_i}$ ($m_i \in M$, a discrete set of integer values) and constant elsewhere. Hence, m_i is the intrinsic dimension of x_i. For our simulation, let $d = 13$ and $M = \{2, 6, 10\}$, and there are $n = 200$ samples for each value in M. After obtaining local dimension estimates, we apply neighborhood smoothing to differing neighborhood sizes and measure the entropy of the local dimension estimates at each size. The results are shown in Fig. 3.12, where the entropy exhibits the same pattern we previously described; after initially decreasing, H remains constant as k approaches the region size of each manifold ($n = 200$). As the smoothing covers multiple manifolds, $k > 200$, the entropy decreases until the smoothing neighborhood eventually covers all

manifolds simultaneously and $H = 0$.

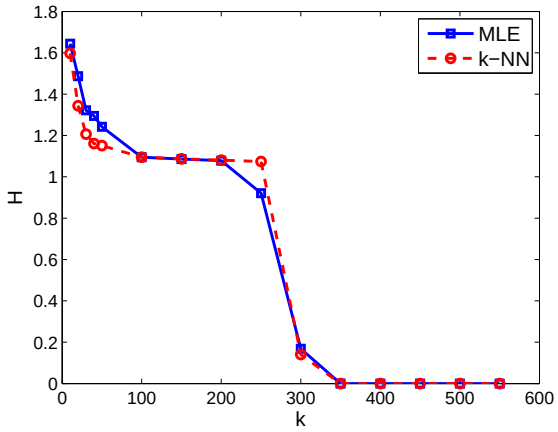
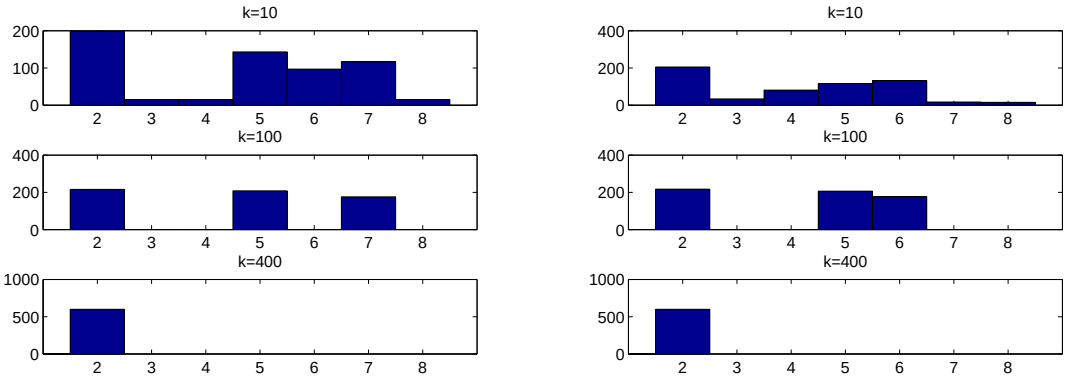


Figure 3.12: The entropy of the local dimension estimates changes as a function of neighborhood size k. As k increases to the size of the differing regions ($k = 200$ samples each), the entropy becomes constant and the data is properly clustered. As the neighborhood incorporates samples from differing manifolds, the entropy decrease until all points estimate at the same value ($k = 350$).

The histogram of local dimension estimates (with both k-NN and MLE methods), which is used to calculate the entropy, is shown in Fig. 3.13 to illustrate the evolution of the dimension estimates. It is clear that at $k = 100$ the 3 distinct clusters are represented, and this value also corresponds to the optimal entropy estimate given *a priori* knowledge that each dimension is represented with a constant probability of $P = \frac{1}{3}$, which yields the entropy value $H = 1.1$. Due to insufficient sampling, the actual value of the dimension estimates ($\{2, 5, 7\}$ for the k-NN algorithm and $\{2, 5, 6\}$ for the MLE method) differ from the true dimensions $\{2, 6, 10\}$. However, this is not of particular concern since the primary objective is to locate clusters of differing *complexity*. It is also worth noting that some samples are misidentified due to the overlapping nature of the 3 clusters (i.e. common dimensions share the same range), but the overall performance is respectable.

We note the dimension estimate obtained when smoothing over the entire set does not correspond to the global dimension of the data. Since we are using a majority



(a) k-NN Algorithm (b) MLE Method

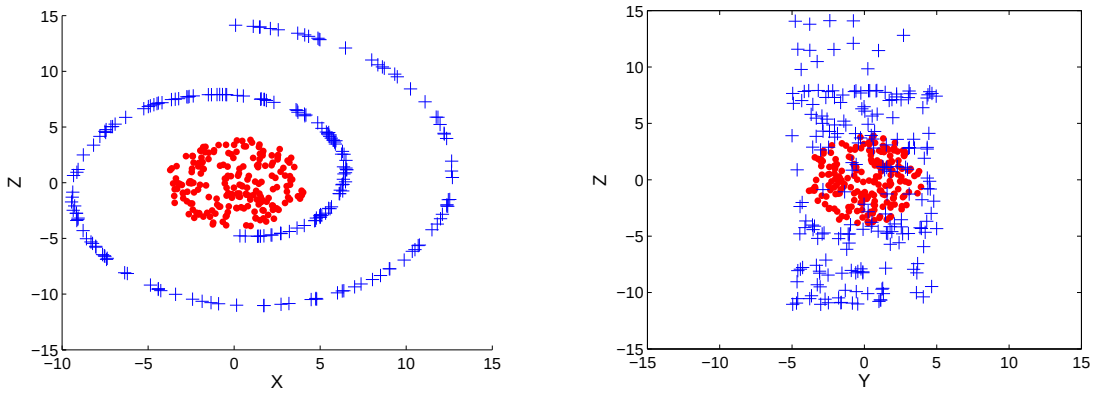
Figure 3.13: Comparing dimension histograms of dimension estimates at various neighborhood sizes, we see that samples are clustered very well at $k = 100$, which corresponds to constant point in the entropy plot shown in Fig. 3.12

.

voting method, the final value will be equal to the estimated dimension which is most represented (with simple tie-breaking rules). This is not necessarily equal to the global dimension, and is often not close to the dimension which best characterizes the entire data set (as in our example).

Let us now compare our clustering performance on a separate synthetic example. Consider the data set $\boldsymbol{X} = [x_1, \ldots, x_{400}]$ which consists of 200 points uniformly sampled on the 'swiss roll' manifold, and 200 points uniformly sampled on an intrinsically 3-dimensional hyper-sphere. Hence, each $x_i \in \mathbb{R}^4$ (points sampled from the 'swiss roll' have a constant value in the 4th dimension) and there are two distinct clusters formed. A visual representation of this set is illustrated in Fig. 3.14, and we compare our method of clustering by *complexity* using local dimension estimation with that of standard clustering methods – Fuzzy c-means [8] and K-means [41]. To demonstrate clustering performance we utilize the Jaccard index [48], which assesses the similarity between a predetermined set of class labels C and a clustering result K. Specifically,

$$J(C, K) = \frac{a}{a + b + c},$$



(a) Side (b) Front

Figure 3.14: Clustering based on local intrinsic dimensionality is useful for problems such as this, in which 3-dimensional hyper-sphere (•) is placed "inside" the 2-dimensional 'swiss roll' (+). Side and front angles of set shown.

where a is the number of pairs of points with the same class label in C and the same cluster label in K, b is the number of pairs which have the same label C but differ in K, and c is the number of pairs of points with the same cluster label in K but different class label in C. Essentially, the Jaccard index gives a rating in the range $[0, 1]$ for which 1 signifies complete agreement between the true labels C and the clustering results K.

Method	Mean Jaccard
Dimension Estimation	0.7834
K-Means	0.4224
Fuzzy c-means	0.3607

Table 3.1: Comparison of various clustering methods on data set consisting of 'swiss roll' and 3-dimensional hyper-sphere manifolds. Performance reported based on mean Jaccard index over a 20-fold cross validation.

We show the results in Table 3.1, over a 20-fold cross validation with i.i.d. realizations of $\boldsymbol{X}$. We see clustering by dimension estimation yields far superior performance to standard methods. While these methods aim to cluster by a variety of means, such as optimizing distances to centroids, dimension estimation simply assigns cluster labels based on the local dimensionality of each data point. In this

simulation we utilized a neighborhood size of $k = 25$ when smoothing, as larger values tended to incorporate both manifolds since they are so close to one another. We acknowledge that clustering by dimensionality is not applicable in many practical problems in which the different clusters exhibit the same dimensionality. However, in the realm of high-dimensional clustering, there may often exist an intrinsic difference in dimensionality, in which our method would be applicable.

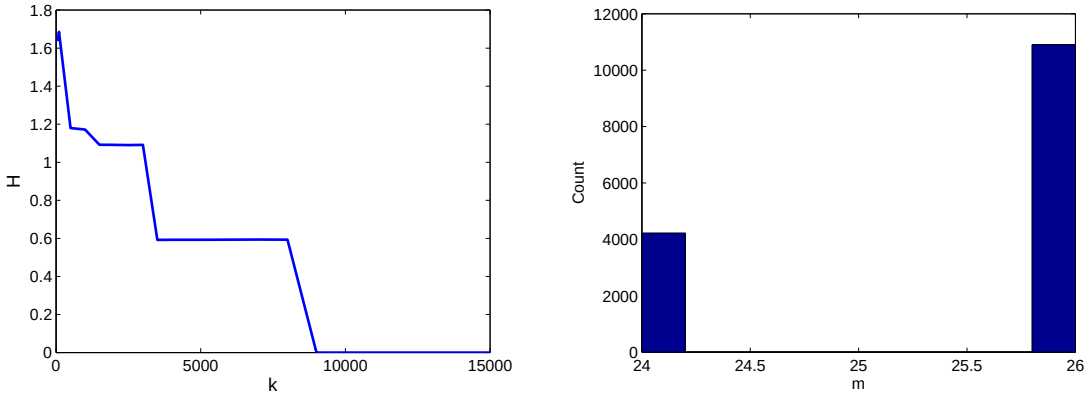
3.4.4 Image Segmentation

After showing the ability to use local dimension estimation for clustering data by complexity, a natural extension is to apply the methods for the problem of image segmentation. Differing textures in images can be considered to have different levels of complexity (e.g. a periodic texture is less complex than a random one). This has been well stated in [59], where natural images and textures are viewed as a collection of fractals. For our purposes we chose to ignore such model assumptions and see whether or not Euclidean dimension can be used as a means of image segmentation. In this case, the same framework as our clustering method applies.

Consider the satellite image of New York City[1] in Fig. 3.16(a), which has a resolution of 1452×1500. We wish to segment the image into land and water masses. To use local dimension estimation, we define $\boldsymbol{X} = [x_1, \ldots, x_n]$, where x_i is a 144-dimensional vector representing a rasterized 12×12 block of the image. After obtaining the local dimension estimates, we apply neighborhood smoothing and recursive entropy estimation as described above. The results, illustrated in Fig. 3.15(a), lead us to define an ideal neighborhood size of $k = 3500$. This allows us to segment the image into 2 regions, defined by the complexity estimates shown in Fig. 3.15(b). The final segmentation can be viewed in Fig. 3.16(b), where the water is well separated from

[1] `http://newsdesk.si.edu/photos/sites_earth_from_space.htm`

the land portions of the island of Manhattan and the surrounding burroughs. We note that this image is that of the smoothed local dimension estimates, uniformly scaled to the range $[0, 255]$.



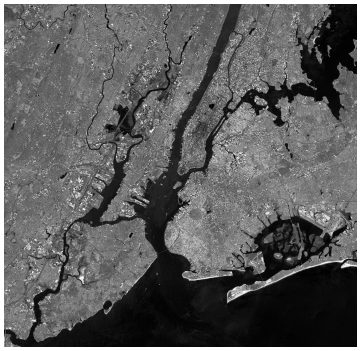
(a) Entropy vs. k

(b) Histogram of Dimension Estimates

Figure 3.15: Plotting the entropy of the dimension estimates suggests a neighborhood size of $k = 4000$, which yields 2 significant clusters in the dimension estimates.

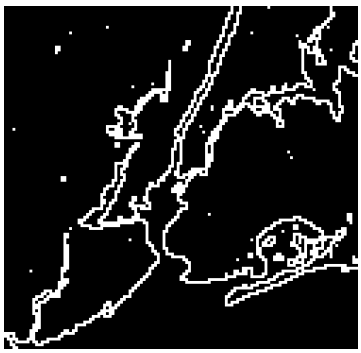
We notice there is a relatively low resolution in our segmentation image, due to the large 12×12 blocks used for estimation. We can correct this by using a smaller pixel blocks, however computational issues prevent us from estimating at much higher resolutions. We can alleviate this problem by estimating at a high resolution only in the areas which require such; this may be determined by using edge detection on the image of local dimension estimates as in Fig. 3.16(c). In the regions which are determined to contain edges, we re-segment at a higher resolution – using 4×4 pixel blocks – with the same recursive entropy estimation process. The results are shown in Fig. 3.16(d); it is clear that this segmentation appears significantly less digitized and more detailed.

While the previous task was simply to segment water from land in an image, we detailed the 'binary' task to demonstrate the process. The problem is easily extended to the multi-texture case, which we illustrate in Fig. 3.17 with images of local di-

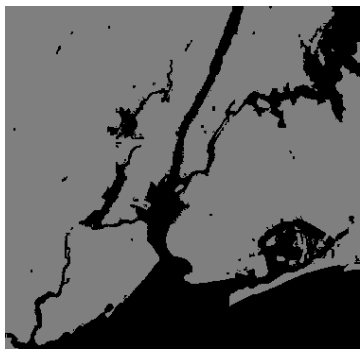


(a) New York City

(b) Low Resolution Segmentation



(c) Edges of Segmented Image



(d) High Resolution Segmentation

Figure 3.16: By using local dimension estimation, neighborhood smoothing, and entropy estimation, we are able to segment the satellite image of New York City into water and land regions. After segmenting the image at a low-resolution, we perform edge detection to find the regions which should be analyzed at a higher resolution, yielding a significantly more detailed segmentation.

mension estimates scaled to the range $[0, 255]$. In these cases we segmented images of a sloth bear[2] and a panda bear cub[3] using the same techniques as previously described, only we utilized a high-resolution segmentation over entire image along with small smoothing neighborhoods. This may give a finer segmentation than required (e.g. the bears are not segmented entirely as one object), but shows the potential segmentation power of local dimension estimation. If a coarser segmentation was desired, larger smoothing neighborhoods may be applied, similar to the previous

[2] `http://newsdesk.si.edu/photos/nzp_sloth_bear.htm`
[3] `http://newsdesk.si.edu/photos/nzp_panda_cub.htm`

case of New York City. We note that by no means are we suggesting that dimension alone is a superior means of image segmentation; we simply illustrate that there is a semblance of power to Euclidean dimension when segmenting natural images, and that dimension may be used in conjunction with other means for this complex task.

3.5 Conclusions and Future Work

We have shown the ability to use local intrinsic dimension estimation for a myriad of applications. The negative bias in global dimension estimation is strongly influenced by the data depth of the samples on the manifold. By developing a global dimension estimator based on the local dimension estimates of the deepest points, we have shown the issue of the negative bias can be significantly reduced. Typically, dimension estimation is used for the purposes of dimensionality reduction of Riemannian manifolds, and we will extend this to the problem of dimensionality reduction on statistical manifolds in the proceeding chapters..

By viewing dimension as a substitute for data complexity, we have applied local dimension estimation to problems which may not naturally be considered. Local dimension estimates can be used to find anomalous activity in router networks, as the overall complexity of the network is decreased when a few sources account for a disproportionate amount of traffic. We have also applied complexity estimation towards the problems of data clustering and image segmentation through the use of neighborhood smoothing. By finding the points in which entropy remains constant as the neighborhood size increases, we are able to optimally cluster the data.

Further analysis into the applications we have presented here is an area for future work. In terms of de-biasing global dimension estimation, the number of interior points decreases (holding total number of points constant) as the dimension increases.

As such, applying significant weight the interior points in averaging over local dimensions may result in large variance of the dimension estimate due to a small sample size. The bias-variance trade-off and its optimization is of great importance, and should be considered an area for future work. Additionally, we would like to further investigate using Euclidean dimension estimation (as opposed to fractal dimensions) for image segmentation, as we feel this is a very interesting application which has not been thoroughly researched. Specifically, we are interested in combining Euclidean dimension with other measures of textures in order to optimally segment a natural image.

3-A Appendix: Non-linear Least Squares Solution of Dimension Estimation

Here we detail how to solve the non-linear least squares problem for the k-NN algorithm for dimension estimation. Given the vector of length functionals $\boldsymbol{L}_n = \left\{L_{\gamma,k}(\boldsymbol{X}_p^1), \ldots, L_{\gamma,k}(\boldsymbol{X}_p^N)\right\}$, for a specific number of samples n, we solve

$$\hat{m} = \arg\min_{m\in\mathbb{Z}} \left\{ \min_c \sum_{i=1}^{Q} \left\| \boldsymbol{L}_{n_i} - n_i^{\alpha(m)} c\mathbf{1} \right\|^2 \right\}, \tag{3.12}$$

where $\mathbf{1}$ is the vector of length n_i whose elements are all 1, in the following manner:

`for` $m = 2$ `to` d `do`

1. Calculate $\hat{c}(m)$ from the expansion of (3.4):

   a) $\hat{c} = \min_c \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i}\|^2 - 2c\sum_{i=1}^{Q} n_i^{\alpha} \boldsymbol{L}_{n_i}^T \mathbf{1} + c^2 \sum_{i=1}^{Q} (n_i^{\alpha})^2 \mathbf{1}^T\mathbf{1},$

   $$\Rightarrow \hat{c} = \sum_{i=1}^{Q} n_i^{\alpha} \boldsymbol{L}_{n_i}^T \mathbf{1} / \sum_{i=1}^{Q} (n_i^{\alpha})^2 \mathbf{1}^T\mathbf{1}$$

2. Calculate the error, $\boldsymbol{\epsilon}(m)$ with m and $\hat{c}$ from step 1

$$\boldsymbol{\epsilon}(m) = \sum_{i=1}^{Q} \|\boldsymbol{L}_{n_i} - \hat{c} n_i^{\alpha(m)} \mathbf{1}\|^2$$

`end`.

$\hat{m} = \arg\min_i \boldsymbol{\epsilon}(i)$

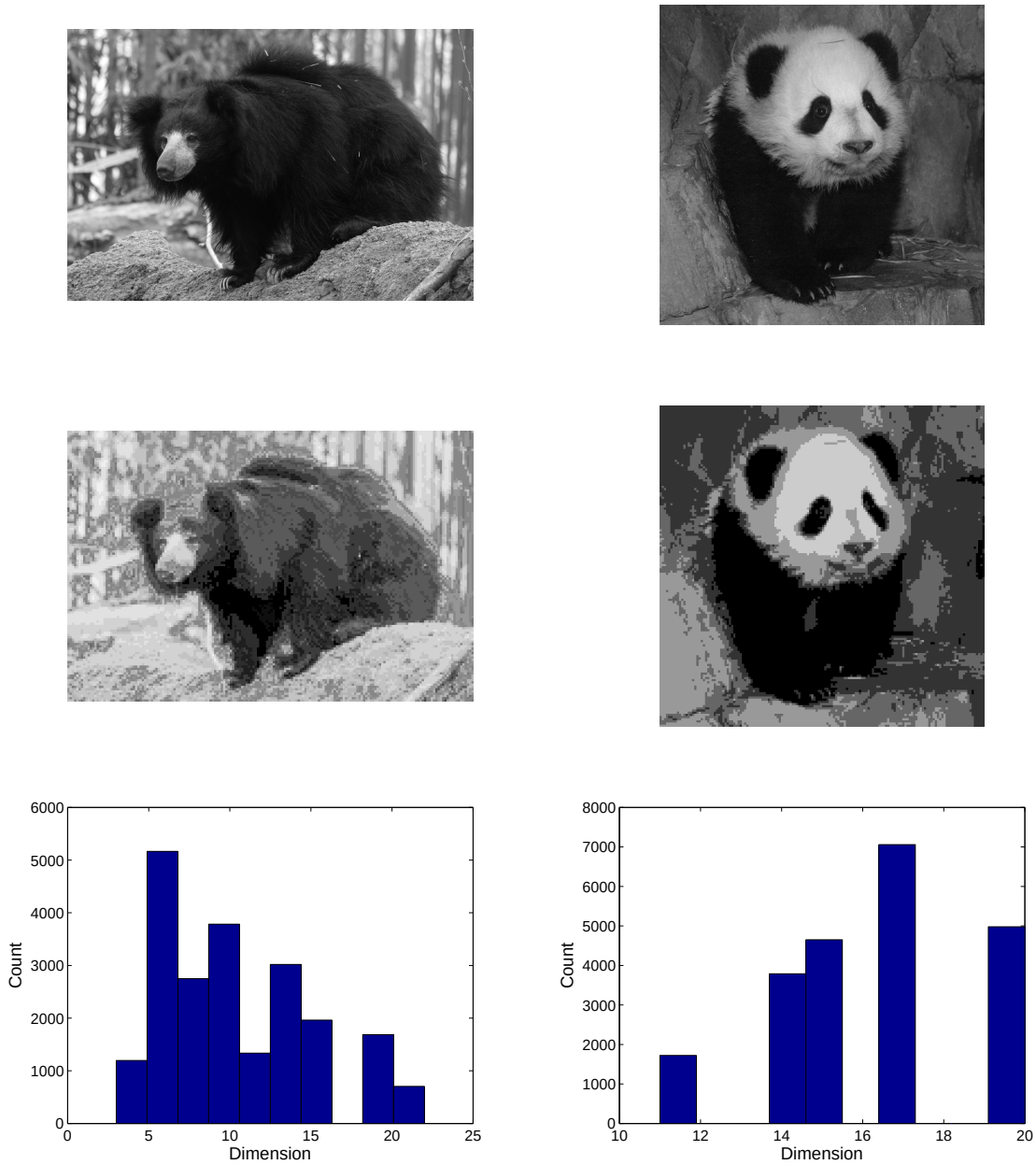


Figure 3.17: Segmentation of multi-texture images using local dimension estimation and neighborhood smoothing. The first row contains the original images, the second row contains the images of local dimension estimates (scaled to $[0, 255]$), while the third row is the histogram of local dimension estimates.

CHAPTER IV

Information-Geometric Embeddings

4.1 Introduction

Recently presented methods of manifold learning and dimensionality reduction [5, 71, 74] focus on finding a low-dimensional representation of the data which is restricted to lie on some Riemannian submanifold of Euclidean space. These derivations of multidimensional scaling (MDS) [27] utilize pairwise Euclidean distances to reconstruct manifolds and embed points into a low-dimensional space. However, it has been well documented that these methods use Euclidean distance as a measure of dissimilarity between elements, and other measures of dissimilarity may be substituted. Isomap [74], for example, approximates geodesic distances between data samples. Laplacian Eigenmaps [5] simply uses Euclidean distance as a means to calculate a weight function. Hence, if an appropriate distance between PDFs is utilized, these well-respected algorithms could be used for an entirely new class of problems on statistical manifolds.

Consider the collection of PDFs $\mathcal{P} = \{p_1, \ldots, p_N\}$ lying on some statistical manifold $\mathcal{M}$. Our goal is to reconstruct $\mathcal{M}$ using only the information available in $\mathcal{P}$. This is a similar setting to traditional manifold learning algorithms which aim to reconstruct Riemannian manifolds based on a finite sampling. In this chapter, we

extend these principles to statistical manifolds. Specifically, we focus on the case where the data is high-dimensional and cannot be represented in a straightforward and meaningful manner in Euclidean space. In many of these cases, a lower dimensional statistical manifold can be used to assess the data for various learning tasks.

Many applications of statistical manifolds have proved promising, such as document classification [19,53,55], flow cytometry analysis [15,16,34], face recognition [4], texture segmentation [56], image analysis [73], clustering [72], and shape analysis [52]. While all have proposed alternatives to using Euclidean geometry for data modeling, most methods (outside of our own work) focus on clustering and classification, and do not explicitly address the problems of dimensionality reduction and visualization. Additionally, most presented work has been in the parametric setting, in which parameter estimation is a necessity for the various methods. This becomes ad-hoc and potentially troublesome if a good model is unavailable.

We provide a start-to-finish framework for determining an information-geometric embedding of PDFs $A : p(x) \to y$, where $y \in \mathbb{R}^m$; expressed through two separate algorithms approaching the problem of statistical manifold reconstruction. These methods include a characterization of data sets in terms of a nonparametric statistical model, a geodesic approximation of the Fisher information distance as a metric for evaluating similarities between data sets, and a dimensionality reduction procedure to obtain a low-dimensional Euclidean embedding of the original high-dimensional data set for the purposes of both classification and visualization. The first presented algorithm – termed *Fisher information nonparametric embedding* (FINE) – embeds PDFs into an open low-dimensional Euclidean space. This is useful when there is no *a priori* knowledge of the manifold structure. If the manifold geometry is known to

be that of a low-dimensional hyper-sphere (e.g. a transformation of multinomial distributions), we offer *Spherical laplacian information maps* (SLIM), which constrains the embedding to the surface of a sphere in Euclidean space. This is useful as the measure of distance in the embedding space (i.e. great-circle distance) is an accurate representation of the true Fisher information distance in the probability space. Both FINE and SLIM are non-linear embedding methods, driven by information, not Euclidean, geometry. In conjunction, our methods require no explicit model assumptions; only that the given data is a realization from an unknown model with some natural parameterization.

Recent work by Lee *et al.* [57] similar to our own [18, 19] has demonstrated the use of statistical manifolds for dimensionality reduction. Specifically, we consider the work presented by Lee *et al.* to be a specialized case of our more general framework. They focus on the specific case of image segmentation, which consists of multinomial distributions as points which lie on an n-simplex (or projected onto an $(n+1)$-dimensional sphere). By framing their problem as such, they are able to exploit the properties of such a manifold: using the cosine distance as an exact computation of the Fisher information distance, and using linear methods (PCA) of dimensionality reduction. They have shown very promising results for the problem of image segmentation, and briefly mention the possibility of using non-linear methods of dimensionality reduction, which they consider unnecessary for their problem. The work we present differs in that we make no assumptions on the type of distributions making up the statistical manifold. Hence, our geodesic approximation for the Fisher information accounts for submanifolds of interest. This is illustrated later in Fig. 4.1, where the submanifold lies on the $(n+1)$-dimensional sphere, but does not fill the entire space. As such, there is no exact measure of the Fisher information between

points, and we must approximate with a geodesic along the manifold. Additionally, we utilize non-linear methods of dimensionality reduction, which we consider to be more relevant for many non-linear types of applications. Finally, by considering all statistical manifolds rather than focusing on those of consisting solely of multinomial distributions, we are able to apply our methods to many problems of practical interest.

The remainder of this chapter is organized as follows: Section 4.2 illustrates our estimation of the Fisher information distance by approximating the geodesic on the statistical manifold. We review several manifold learning techniques for dimensionality reduction in Section 4.3, and proceed with the formulation of the FINE algorithm in Section 4.4 and SLIM in Section 4.5. We illustrate the results of using FINE and SLIM on real and synthetic data sets in Section 4.6. Finally, we draw conclusions and discuss the possibilities for future work in Section 4.7.

4.2 Approximation of Distance on Statistical Manifolds

Let us consider the approximation function $\hat{D}_F(p_1, p_2)$ of the Fisher information distance between p_1 and p_2, which may can be calculated using a variety of metrics as $p_1 \to p_2$ (see Section 2.3). If p_1 and p_2 do not lie closely together on the manifold, these approximations become weak, as the convergence properties no longer hold. It has previously been suggested [57] to use the cosine distance as a strict approximation of the Fisher information distance. This is due to the fact that the cosine distance measures a portion of a great circle on a hyper-sphere, and in the discrete case all PDFs can be considered as multinomial distributions which may be projected onto a hyper-sphere manifold. This usage of the cosine distance is true only in the assumption that the manifold of interest fills the entire space of the hyper-sphere.

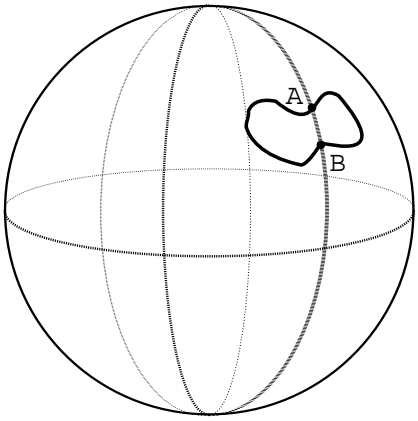


Figure 4.1: Given a 1-dimensional submanifold (the curvy dark line) of interest lying on a 2-dimensional sphere manifold, the Fisher information distance is the shortest path connecting the points A and B along the 1-D submanifold, rather than the length of a portion of the great circle connecting the points on the sphere.

In many cases the PDFs are constrained to form a submanifold of interest, and the geodesic is no longer accurately described as a portion of a great circle on the hyper-sphere. This is illustrated in Fig. 4.1 in which we represent a $(d-1)$-dimensional submanifold which occupies a subspace of the d-dimensional hyper-sphere ($d = 2$ for illustration). The Fisher information distance is equal to the shortest path along the submanifold (curvy line), and in this case that is not equal to the portion of a great circle on a hyper-sphere connecting the two points. Hence, there are situations in which standard approximations of the information distance do not converge to the true distance, and it is necessary to approximate the geodesic along the manifold.

A good approximation can still be achieved if the manifold is densely sampled between the two end points. By defining the path between p_1 and p_2 as a series of connected segments and summing the length of those segments, we may approximate the length of the geodesic with graphical methods. Specifically, given the set of N PDFs parameterized by $\mathcal{P}_\theta = \{\theta_1, \ldots, \theta_N\}$, the Fisher information distance between

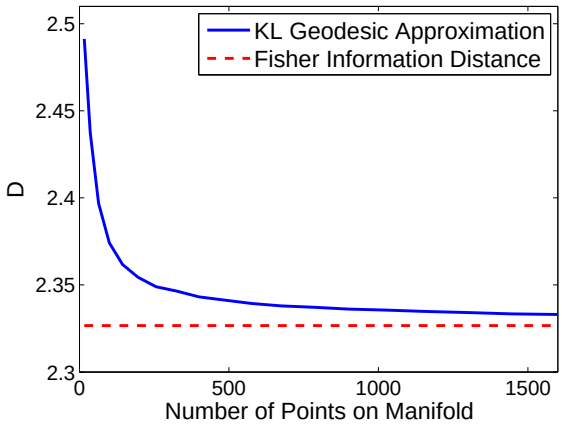


Figure 4.2: Convergence of the graph approximation of the Fisher information distance using the Kullback-Leibler divergence. As the manifold is more densely sampled, the approximation approaches the true value.

p_1 and p_2 can be estimated as:

$$D_F(p_1, p_2) \approx \min_{m, \{\theta_{(1)}, \ldots, \theta_{(M)}\}} \sum_{i=1}^{M-1} D_F(p(\theta_{(i)}), p(\theta_{(i+1)})), \quad p(\theta_{(i)}) \to p(\theta_{(i+1)}) \; \forall \, i$$

where $p(\theta_{(1)}) = p_1$, $p(\theta_{(M)}) = p_2$, $\{\theta_{(1)}, \ldots, \theta_{(M)}\} \in \mathcal{P}_\theta$, and $M \leqslant N$.

Using an approximation of the Fisher information distance as $p_1 \to p_2$, we can now define an approximation function G for all pairs of PDFs:

$$G(p_1, p_2; \mathcal{P}) = \min_{M, \mathcal{P}} \sum_{i=1}^{M-1} \hat{D}_F(p_{(i)}, p_{(i+1)}), \quad p_{(i)} \to p_{(i+1)} \; \forall \, i \tag{4.1}$$

where $\mathcal{P} = \{p_1, \ldots, p_N\}$ is the available collection of PDFs on the manifold. Intuitively, this estimate calculates the length of the shortest path between points in a connected graph on the well sampled manifold, and as such $G(p_1, p_2; \mathcal{P}) \to D_F(p_1, p_2)$ as $N \to \infty$. This is similar to the manner in which Isomap [74] approximates distances on Euclidean manifolds. Figure 4.2 illustrates this approximation by comparing the KL graph approximation to the actual Fisher information distance for the univariate Gaussian case. As the manifold is more densely sampled (uniformly in mean and variance parameters for this simulation), the approximation converges to the true Fisher information distance, as calculated in (2.7).

4.3 Dimensionality Reduction

Given a matrix of dissimilarities between entities, many algorithms have been developed to find a low-dimensional embedding of the original data $\psi : \mathcal{M} \rightarrow \mathbb{R}^d$. These techniques have been classified as a group of methods called multidimensional scaling (MDS) [27]. There are supervised methods, which are generally used for classification purposes, and unsupervised methods, which are often used for clustering and visualization. Using these MDS methods allows us to find a single low-dimensional coordinate representation of each high-dimensional, large sample, data set.

4.3.1 Classical Multi-Dimensional Scaling

Classical MDS (cMDS) takes a matrix of dissimilarities and embeds each point into a Euclidean space. This unsupervised method permits the calculation of the low-dimensional embedding coordinates which reveal any natural separation or clustering of the data, preserving the data geometry.

Define D as a dissimilarity matrix which contains (or approximates) Euclidean distances between N element pairs (e.g. $D \in \mathbb{R}^{N \times N}$). Let B be the "double centered" matrix which is calculated by taking the matrix of *squared* dissimilarities (denoted $D^{(2)}$), subtracting its row and column means, then adding back the grand mean and multiplying by $-\frac{1}{2}$. Mathematically, this process is solved by

$$B = -\frac{1}{2} H D^{(2)} H,$$

where $H = I - (1/N)11^T$, I is the N-dimensional identity matrix, and 1 is an N-element vector of ones.

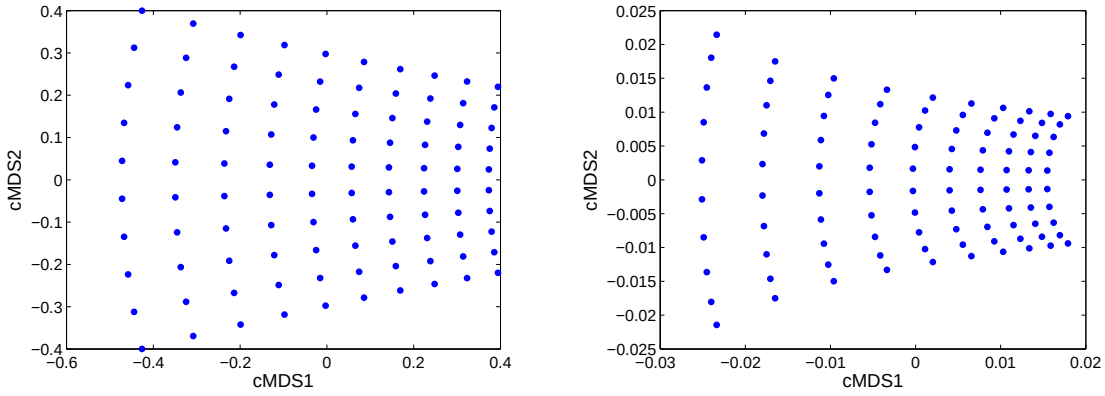
The embedding coordinates, $\boldsymbol{Y} \in \mathbb{R}^{d \times N}$, can then be determined by taking the eigenvalue decomposition of B,

$$B = [V_1 \; V_2] \mathrm{diag}\left(\lambda_1, ..., \lambda_N\right) [V_1 \; V_2]^T,$$

where $[V_1\ V_2]$ is a partitioned matrix, and calculating

$$\boldsymbol{Y} = \text{diag}\left(\lambda_1^{1/2}, ..., \lambda_d^{1/2}\right) V_1^T.$$

The matrix V_1 consists of the eigenvectors corresponding to the d largest eigenvalues $\lambda_1, \ldots, \lambda_d$ while the remaining $N - d$ eigenvectors are represented as V_2. The term 'diag$(\lambda_1, \ldots, \lambda_N)$' refers to an $N \times N$ diagonal matrix with λ_i as its i^{th} diagonal element.



(a) Exact Fisher Information

(b) Kullback-Leibler Approximation

Figure 4.3: Classical MDS to the matrix of a) Fisher information distances and b) Kullback-Leibler geodesic approximations of the Fisher information distance, on a grid of univariate normal densities, parameterized by (μ, σ)

To continue our Gaussian illustration from Section 2.2.2, in which $\mathcal{P} = \{p_1, \ldots, p_N\}$ is a family of univariate Gaussian distributions, let D be the matrix of exact Fisher information distances defined in (2.7), where $D(i, j) = D_F(p_i, p_j)$. Figure 4.3(a) displays the results of applying cMDS to D. We demonstrate the embedding with the geodesic approximation of the Fisher information distance, $D(i, j) = G(p_i, p_j; \mathcal{P})$, in Fig. 4.3(b), which is very similar to the embedding created with the exact values. It is clear that while the densities defining the set $\mathcal{P}$ are parameterized on a rectangular grid, the manifold on which $\mathcal{P}$ lives is not rectangular itself, which is due to the differing effects that changes in mean and variance have on the Gaussian PDF.

4.3.2 Laplacian Eigenmaps

Laplacian Eigenmaps (LEM) is an unsupervised technique developed by Belkin and Niyogi and first presented in [5]. This performs non-linear dimensionality reduction by performing an eigenvalue decomposition on the graph Laplacian formed by the data. As such, this algorithm is able to discern low-dimensional structure in high-dimensional spaces that were previously indiscernible with methods such as principal components analysis and classical MDS. The algorithm contains three steps and works as follows:

1. Construct adjacency graph

   Given dissimilarity matrix D_X between data points in the set $\boldsymbol{X}$, define the graph G over all data points by adding an edge between points i and j if $\boldsymbol{X}_i$ is one of the k-nearest neighbors of $\boldsymbol{X}_j$ (k is defined by the user).

2. Compute weight matrix W

   If points i and j are connected, assign $W_{ij} = e^{-\frac{D_X(i,j)^2}{t}}$, otherwise $W_{ij} = 0$.

3. Construct low-dimensional embedding

   Solve the generalized eigenvalue problem

   $$L\mathbf{v} = \lambda D\mathbf{v},$$

   where D is the diagonal weight matrix in which $D_{ii} = \sum_j W_{ji}$, and $L = D - W$ is the Laplacian matrix. If $[\mathbf{v}_1, \ldots, \mathbf{v}_d]$ is the collection of eigenvectors associated with d smallest generalized eigenvalues which solve the above, the d-dimensional embedding is defined by $\mathbf{y}_i = (v_{i1}, \ldots, v_{id})^T, 1 \leqslant i \leqslant N$.

4.3.3 Additional MDS Methods

While we choose to only detail the cMDS and LEM algorithms, there are many other methods for performing dimensionality reduction in a linear fashion (PCA) and

non-linearly (Local Linear Embedding [71]) for unsupervised learning. For supervised learning there are also linear (Linear Discriminant Analysis [35,42,63]) and non-linear (Classification Constrained Dimensionality Reduction [67], Neighbourhood Component Analysis [38]) methods, all of which can be applied to our framework. We do not highlight the heavily utilized Isomap [74] algorithm since it is identical to using cMDS on the approximation of the geodesic distances.

4.4 FINE Algorithm

We have presented a series of methods for manifold learning developed in the field of information geometry. By performing dimensionality reduction on a family of data sets, we are able to both better visualize and classify the data. In order to obtain a lower dimensional embedding, we calculate a dissimilarity metric between data sets within the family by approximating the Fisher information distance between their corresponding PDFs. This has been illustrated with the family of univariate normal probability distributions.

In problems of practical interest, however, the parameterization of the probability densities is usually unknown. We instead are given a family of data sets $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, in which we may assume that each data set $\boldsymbol{X}_i$ is a realization of some underlying probability distribution to which we do not have knowledge of the parameters. As such, we rely on nonparametric techniques to estimate both the probability density and the approximation of the Fisher information distance. In the work presented in this thesis, we implement kernel density estimation methods (see Appendix A), although k-NN methods are also applicable. Following these approximations, we are able to perform the same multidimensional scaling operations as previously described.

Fisher Information Nonparametric Embedding (FINE) is presented in Algorithm 4.1 and combines all of the presented methods in order to find a low-dimensional embedding of a collection of data sets. If we assume each data set is a realization of an underlying PDF, and each of those distributions lie on a manifold with some natural parameterization, then this embedding can be viewed as an embedding of the actual manifold into Euclidean space. Note that in line 5, 'mds(G, d)' refers to using any multidimensional scaling method to embed the dissimilarity matrix G into a Euclidean space with dimension d.

Algorithm 4.1. Fisher Information Nonparametric Embedding

Input: Collection of data sets $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$; the desired embedding dimension d

1: **for** $i = 1$ to N **do**

2: Calculate $\hat{p}_i(\boldsymbol{x})$, the density estimate of $\boldsymbol{X}_i$

3: **end for**

4: Calculate G, where $G(i, j)$ is the geodesic approximation of the Fisher information distance between p_i and p_j (4.1)

5: $\boldsymbol{Y} = \text{mds}(G, d)$

Output: d-dimensional embedding of $\mathcal{X}$, into Euclidean space $\boldsymbol{Y} \in \mathbb{R}^{d \times N}$

At this point it is worth stressing the benefits of this framework. Through information geometry, FINE enables the joint embedding of multiple data sets $\boldsymbol{X}_i$ into a single low-dimensional Euclidean space. By viewing each $\boldsymbol{X}_i \in \mathcal{X}$ as a realization of $p_i \in \mathcal{P}$, we reduce the numerous samples in $\boldsymbol{X}_i$ to a single point. The dimensionality of the statistical manifold may be significantly less than that of the Euclidean realizations. For example, a Gaussian distribution is entirely defined by its mean μ and covariance Σ, leading to a 2-dimensional statistical manifold, while the dimen-

sionality of the realization $\boldsymbol{X} \sim \mathcal{N}(\mu, \Sigma)$ may be significantly larger (i.e. $\mu \in \mathbb{R}^d$, $d \gg 2$). MDS methods reduce the dimensionality of p_i from the Euclidean dimension to the dimension of the statistical manifold on which it lies. This results in a single low-dimensional representation of each original data set $\boldsymbol{X}_i \in \mathcal{X}$.

4.5 Spherical Embedding Constraints

With FINE, we find an embedding into an open Euclidean space in $\mathbb{R}^d$, for which the L_2-norm is an appropriate and accurate distance metric, directly related to the Fisher information distance on the original statistical manifold. This is useful when the manifold structure is unknown, as the embedding space is relatively unconstrained. Suppose, however, that there exists *a priori* knowledge that the statistical manifold is a portion of a hyper-sphere. This can be realized with multinomial distributions by applying a monotonic transformation $p'(x) = \sqrt{p(x)}$, converting the original d-dimensional simplex to a $(d+1)$-dimensional hyper-sphere with unit radius. In such situations, it may be beneficial to constrain the low-dimensional embedding to the surface of a sphere, which will enable the usage of the great-circle distance, the natural measure of a geodesic on a sphere. Specifically, given points on the unit sphere parameterized with spherical coordinates $\theta = [\phi, \psi]^T$, $-\frac{\pi}{2} \leqslant \phi \leqslant \frac{\pi}{2}$ and $0 \leqslant \psi \leqslant 2\pi$, the distance between θ_i and θ_j is defined as

$$D_{S^2}(\theta_i, \theta_j) = \arccos\left(\cos(\phi_i)\cos(\phi_j)\cos(\psi_i - \psi_j) + \sin(\phi_i)\sin(\phi_j)\right). \tag{4.2}$$

Let us briefly return to Laplacian Eigenmaps [5], which looked to solve the following optimization problem:

$$\boldsymbol{Y} = \arg\min_{\{y_i\}} \sum_i \sum_j W_{ij} \|y_i - y_j\|^2$$

under appropriate constraints, where the weights W_{ij} are chosen to incur heavy penalties if neighboring points are mapped far apart. Note that this cost is optimizing

the total length of the embedding map, with edge lengths between nodes equal to W_{ij} times some measure of distance (Euclidean in this case).

We may utilize this framework towards an information-geometric embedding by modifying the choice of distance measure to that of the great-circle distance. Specifically, we can solve the optimization:

$$\Theta = \arg\min_{\{\theta_i\}} \sum_i \sum_j W_{ij} D_{S^2}(\theta_i, \theta_j), \tag{4.3}$$

under similarly appropriate constraints and weightings, where $\Theta = [\theta_1, \ldots, \theta_N]$. While using spherical MDS [27] may be also be appropriate, optimizing (4.3) adds a sense of locality that better preserves the local neighborhood structure of the manifold.

Notice that under no additional constraints, the trivial solution to (4.3) is to collapse all samples to the same embedded point. To prevent this, we add a constraint designed to regulate the spread of the embedded points on the sphere. Specifically, let us solve (4.3) such that we maximize

$$\sum_i \sum_j D_{S^2}(\theta_i, \theta_j)^\gamma, \tag{4.4}$$

where $0 < \gamma < 2$ is a power-weighting constant which regulates the spread on the sphere. One may view this constraint as maximizing the length of the graph formed when each embedded point represents a node and the length of the edge between nodes is the great-circle distance between points, raised to the power γ. By using (4.3) in conjunction with maximizing (4.4), we obtain the final objective function

$$\Theta = \arg\max_{\{\theta_i\}} \sum_i \sum_j D_{S^2}(\theta_i, \theta_j)^\gamma - W_{ij} D_{S^2}(\theta_i, \theta_j), \tag{4.5}$$

which ensures that close PDFs will be represented by close points in the embedding space, but the trivial solution is avoided.

One may also view our spread constraint (4.4) as having a relationship to controlling the entropy of the data. As detailed by Costa and Hero in [25], the data entropy may be estimated as a function of the length the minimal spanning tree (MST)

$$\hat{L}_\gamma(\boldsymbol{X}) = \min_{T \in \mathcal{T}} \sum_{e \in T} D(e)^\gamma, \tag{4.6}$$

where $\mathcal{T}$ is the set of spanning trees over $\boldsymbol{X}$, e is an edge between sample points, and $D(e)$ is the length of that edge (i.e. the distance between sample points). Larger values of $\hat{L}_\gamma(\boldsymbol{X})$ are related to larger entropy values of the data $\boldsymbol{X}$. We essentially measure the length of the maximal spanning tree (i.e. all nodes connected by an edge), which is indeed an element in $\mathcal{T}$. Hence, while our cost was designed to regulate the spread of embedded points to prevent trivial solutions, there is also a direct relationship to the entropy of the data. This result is also intuitive, as entropy is minimized with the trivial point solution, while maximized over a uniform distribution.

Unlike LEM, there is no closed form eigenvalue solution to this optimization, as the distance measure is highly non-linear. Hence, we solve the optimization with gradient ascent methods (see Appendix 4-A). Let our objective function be measured as

$$J = \sum_i \sum_j D_{S^2}(\theta_i, \theta_j)^\gamma - W_{ij} D_{S^2}(\theta_i, \theta_j),$$

we may iteratively determine the optimal embedding Θ through the process

$$\Theta_{l+1} = \Theta_l + \mu \frac{\partial}{\partial \Theta_l} J,$$

where μ is the step size and $\frac{\partial}{\partial \Theta} J$ is the direction of the gradient of the objective. The complete derivation of this gradient is available in Appendix 4-B.

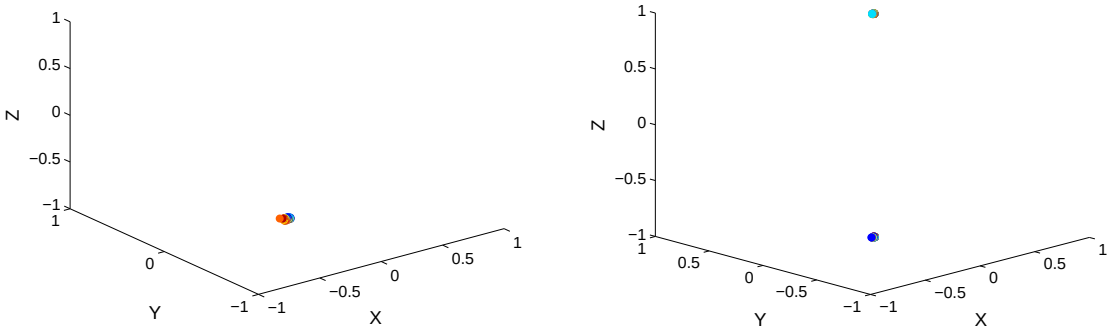
We refer to this framework as *Spherical Laplacian Information Maps* (SLIM), as we find an information-geometric embedding of a statistical manifold, constrained to

the surface of an intrinsically 2-dimensional sphere. The weights W_{ij} are calculated in a similar way to LEM, using the Fisher information distance rather than Euclidean distance,

$$W_{ij} = \exp(-D_F(p_i, p_j)/t),$$

if nodes i and j are connected, with t being some constant.

4.5.1 Spread Constraint



(a) $\gamma = .01$ (b) $\gamma = 2$

Figure 4.4: When using $\gamma \to 0$ or $\gamma \to \infty$, a trivial solution for SLIM is found in which PDFs collapse to either 1 or 2 points, respectively, at the poles of the sphere. The illustrated data was 74 3-dimensional normal distributions with means equal to the location on the unit sphere.

The choice of γ is of particular importance, as this value controls the actual spread between points. For example, as $\gamma \to 0$, the constraint (4.4) approaches the constant N, the number of PDFs, as each power-weighted distance approaches 1. This will result in an embedding for which trivial point solution is optimal. As $\gamma \to \infty$, the solution is optimal when all samples collapse to 2 points at the poles of the sphere, which is the largest distance possible. Both of these cases are illustrated in Fig. 4.4, where we demonstrate on a collection of 3-dimensional normal PDFs with mean values equal to their location on the unit sphere. The full description of this data is discussed shortly in Section 4.6.1, where we also demonstrate the embedding with

$\gamma = 0.5$, which obtains the desired results.

4.5.2 SLIM Algorithm

We now present the full algorithm for SLIM, which embeds PDFs onto a 2-dimensional spherical subspace. The resultant embedding is parameterized through spherical coordinates $\theta = [\phi, \psi]^T$, which maps to a 3-dimensional Euclidean subspace, constrained to lie on the surface of a sphere. The user-defined constant γ determines how large a portion of the sphere the embedding should occupy.

Algorithm 4.2. Spherical Laplacian Information Maps

Input: Collection of data sets $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$; power-weighting constant γ; step size μ

1: **for** $i = 1$ to N **do**
2: Calculate $\hat{p}_i(\boldsymbol{x})$, the density estimate of $\boldsymbol{X}_i$
3: **end for**
4: Calculate the pairwise weight matrix $W_{ij} = \exp(-D_F(p_i, p_j)/t)$ if nodes i and j are connected
5: $l = 1$
6: **while** $|J_l - J_{l-1}| > \epsilon$ **do**
7: Calculate $\frac{\partial}{\partial \Theta_l} J$
8: $\Theta_{l+1} = \Theta_l + \mu \frac{\partial}{\partial \Theta_l} J$
9: $J = \sum_i \sum_j D_{S^2}(\theta_i, \theta_j)^\gamma - W_{ij} D_{S^2}(\theta_i, \theta_j)$
10: $l = l + 1$
11: **end while**

Output: Embedding of $\mathcal{X}$, constrained to the sphere $\Theta = [\theta_1, \ldots, \theta_N]$

The full description of the SLIM algorithm is available in Algorithm 4.2. Empirical

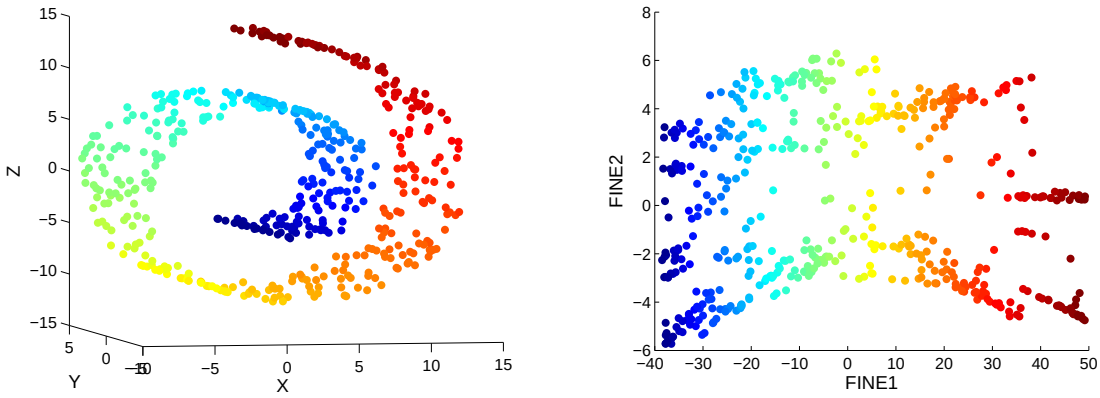
testing suggests that a value of $0.1 < \gamma < 1$ yields desirable results, although we would suggest users empirically determine an appropriate γ for the data of interest. We note that although we restrict our SLIM embedding to the 2-dimensional sphere, it may be formulated for embedding onto an arbitrary d-dimensional hyper-sphere, although the implementation details are more difficult.

4.6 Simulations

We have illustrated the uses of the presented framework in the previous sections with a manifold consisting of the set of univariate normal densities, $\mathcal{P}$. We now present several synthetic and practical applications for the framework, all of which are based around visualization and classification. In each application, the densities are unknown, but we assume they lie on a manifold with some natural parameterization. Unless otherwise noted, all densities are estimated with a Gaussian kernel KDE.

4.6.1 Synthetic Data

Manifold Reconstruction



(a) Swiss Roll

(b) FINE reconstruction

Figure 4.5: Given a collection of data sets with a Gaussian distribution having means equal to the location of points a sampled 'swiss roll' manifold, our methods are able to reconstruct the original 'unrolled' statistical manifold from which each data set is derived.

To demonstrate the ability of our methods to reconstruct the statistical manifold,

we create a known manifold of densities. Let $\boldsymbol{Y} = [y_1, \ldots, y_N]$, where each y_i is uniformly sampled on the 'swiss roll' manifold (see Fig. 4.5(a)). Let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, where each $\boldsymbol{X}_i$ is generated from a normal distribution $\mathcal{N}(y_i, I)$, where I is the identity matrix. As such, we have developed a statistical manifold of known parameterization, which is sampled by known PDFs. Utilizing FINE in an unsupervised manner, with the geodesic symmetric KL-divergence as our measure of dissimilarity, we are able to recreate the original manifold $\boldsymbol{Y}$ strictly from the collection of data sets $\mathcal{X}$. This is shown in Fig. 4.5(b) where each set is embedded into 2 cMDS dimensions, and the swiss roll is reconstructed in an 'unrolled' manner. While this embedding could be constructed by using Isomap on $\boldsymbol{Y}$ – or the mean of each set $\boldsymbol{X}_i$ – that requires a parametric model of the data. Our nonparametric methods illustrate that FINE can be used for visualizing the statistical manifold as well, without a priori knowledge of the data.

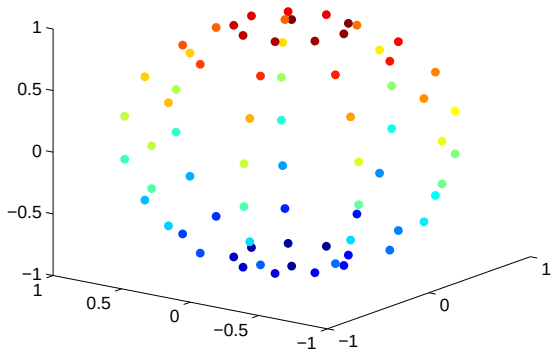
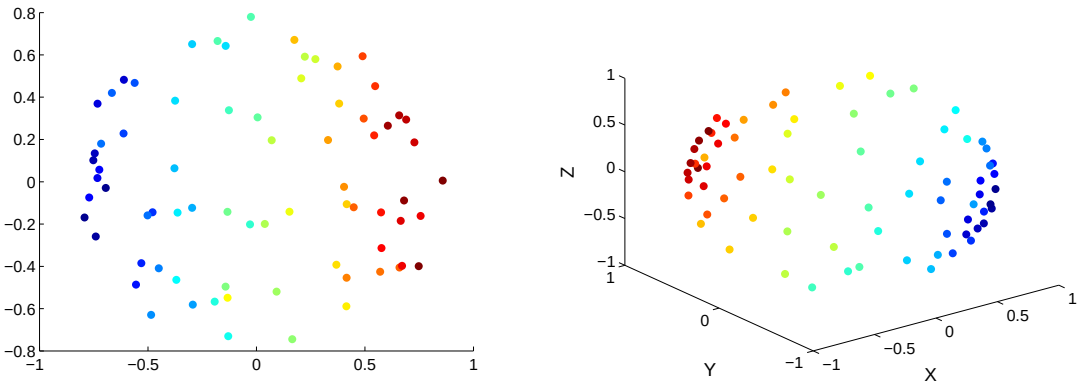


Figure 4.6: An example statistical manifold in which Gaussian distributions have mean values equal to the location on the sphere.

We continue by reconstructing a statistical manifold parameterized by the intrinsically 2-dimensional sphere. Given that there is no way to appropriately embed the sphere into a 2-dimensional Euclidean space, this is a good demonstration of the



(a) FINE Reconstruction

(b) SLIM Reconstruction

Figure 4.7: The unconstrained space of FINE is unable to embed the statistical manifold parameterized by the sphere into 2-dimensional Euclidean space. By constraining the embedding to the surface of a sphere, SLIM gives a more accurate reconstruction.

power of SLIM. We define our data in a similar manner to the swiss roll example; let $\boldsymbol{Y} = [y_1, \ldots, y_N]$, where each y_i lies on the surface of the sphere. Note that these $\{y_i\}$ are generated with uniform spacing in both azimuth and elevation angles (see Fig. 4.6). Let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, where each $\boldsymbol{X}_i$ is generated from a normal distribution $\mathcal{N}(y_i, I)$. Figure 4.7(a) illustrates the clear inability to appropriately embed a spherical statistical manifold into a 2-dimensional Euclidean space. Notice that points which are the maximum distance apart on the manifold are 'flattened' to be nearest neighbors in the embedding space. When utilizing SLIM however, one can easily reconstruct the spherical manifold into a 2-dimensional space parameterized by spherical coordinates $\theta = [\phi, \psi]^T$. In Fig. 4.7(b) we illustrate this embedding in a 3-dimensional Euclidean space (by converting from spherical to cartesian coordinates) simply for the purposes of visualization. Although this visualization is difficult due to the 2-dimensional constraints of this presentation medium, one can easily use intuition to mentally envision the embedding on the sphere. Note that in this SLIM simulation we set $\gamma = 0.5$.

One may argue that an embedding with FINE into a 3-dimensional Euclidean

space will alleviate the discussed issues and still form a sphere in an unconstrained space. While this is indeed true, the embedding is still not the optimal representation. This arises from the fact that embedding into a Euclidean space forces the usage of the L_2-norm of distance, while embedding into a spherical space utilizes the great-circle distance. Given that any unsupervised embedding should preserve relative pairwise distances, let us measure the error between embedding methods. Specifically, we define $E_\star = \|\tilde{D}_F(\mathcal{X}) - \tilde{D}_\star(\mathcal{X})\|_F^2$, where $\tilde{D}_F$ is the pairwise Fisher distance matrix approximated with the geodesic Hellinger distance, scaled such that the maximum distance equals 1. $\tilde{D}_\star$ is the similarly scaled pairwise distance matrix in the embedding space, calculated with the L_2-norm when using FINE and the great-circle distance when using SLIM. Results show that FINE yields an error of $E_{FINE} = 53.4$, while the SLIM embedding was much more accurate at $E_{SLIM} = 37.9$.

Dimensionality Reduction

While the previous simulations focused on the task of manifold reconstruction, we now illustrate the usage of SLIM for dimensionality reduction in the following manner. Let $\alpha^{(i)} = [\alpha_1^{(i)}, \ldots, \alpha_5^{(i)}]^T$ be uniformly distributed as a 5-dimensional vector satisfying the properties of a multinomial distribution: $\alpha_j^{(i)} \geqslant 0$ and $\sum_j \alpha_j^{(i)} = 1$. For each $\alpha^{(i)}$, we draw an i.i.d. realization $\boldsymbol{X}_i$ from a Dirichlet distribution

$$f\left(x_1, \ldots, x_4; \alpha_1^{(i)}, \ldots, \alpha_5^{(i)}\right) = \frac{1}{B(\alpha^{(i)})} \prod_{j=1}^{5} x_j^{\alpha_j^{(i)}-1},$$

where $x_5 = 1 - \sum_j^4 x_j$ and

$$B(\alpha^{(i)}) = \frac{\prod_j \Gamma(\alpha_j^{(i)})}{\Gamma\left(\sum_j \alpha_j^{(i)}\right)}$$

is the multinomial beta function, expressed in terms of the gamma function. Hence, we create a collection of data sets $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ from a statistical manifold

paramaterized by the simplex. Given that the simplex can be mapped to a portion of the sphere by the square root, this may be a good scenario for SLIM.

Let us further add a classification aspect to the problem, by defining class labels such that those data sets generated with parameters $\alpha_1^{(i)} + \alpha_2^{(i)} > 0.4$ belong to class 1, while all other sets belong to class 2. Essentially, this measures whether or not more than 40% of the probability mass was covered in the first 40% (2 out of 5) of the variates of the parameterization.

Using $N = 100$ data sets, we perform leave-one-out cross validation over 20 classification trials, i.i.d. in $\{\alpha^{(i)}\}$. We compare classification performance (with a k-NN classifier) of SLIM to that of FINE with LEM, embedded in both 2 and 3 dimensions, and illustrate the best performance results in Table 4.1 (over $k \in [1, 15]$). We believe that SLIM shows superior performance to FINE in 2-D, and comparable to FINE in 3-D, due to the fact that the original PDFs could be easily parameterized by the non-negative portion of the hyper-sphere. When using SLIM for dimensionality reduction, we maintain the spherical constraint while the mapping allows for negativity, essentially yielding an additional degree of freedom. This explains the similar results to the 3-dimensional embedding with FINE.

Method	Classification Rate (%)	
	Mean	STD
SLIM 2-D	80.3	6.3
FINE 2-D	76.9	6.9
FINE 3-D	80.4	5.5

Table 4.1: Classification rates for performing dimensionality reduction on the set of Dirichlet distributions parameterized by multinomials. The 2-dimensional embedding found by SLIM outperforms that of FINE using LEM in 2-dimensions and performs comparably to the 3-D embedding.

Note that we are not implying that SLIM is in general a superior algorithm to FINE. In fact the spherical constraint forces significant limitations on SLIMs usage. However, when *a priori* knowledge states that the manifold is indeed a sphere, or

portion thereof, the constraint is appropriate and yields potentially significant gains for the final embedding.

4.6.2 Document Classification

Recent work has shown interest in using dimensionality reduction for the purposes of document classification [51] and visualization [45]. Typically documents are represented as very high-dimensional PDFs, and learning algorithms suffer from the *curse of dimensionality*. Dimensionality reduction not only alleviates these concerns, but it also reduces the computational complexity of learning algorithms due to the resultant low-dimensional space. As such, the problem of document classification is an interesting application for FINE.

Given a collection of documents of known class, we wish to best classify a document of unknown class. A document can be viewed as a realization of some overriding probability distribution, in which different distributions will create different documents. For example, in a newsgroup about computers you could expect to see multiple instances of the term "laptop", while a group discussing recreation may see many occurrences of "sports". The counts of "laptop" in the recreation group, or "sports" in the computer group would predictably be low. As such, the distributions between articles in computers and recreation should be distinct. In this setting, we defined the PDFs as the *term frequency* representation of each document. Specifically, let x_i be the number of times term i appears in a specific document. The PDF of that document can then be characterized as the multinomial distribution of normalized word counts, with the maximum likelihood estimate provided as

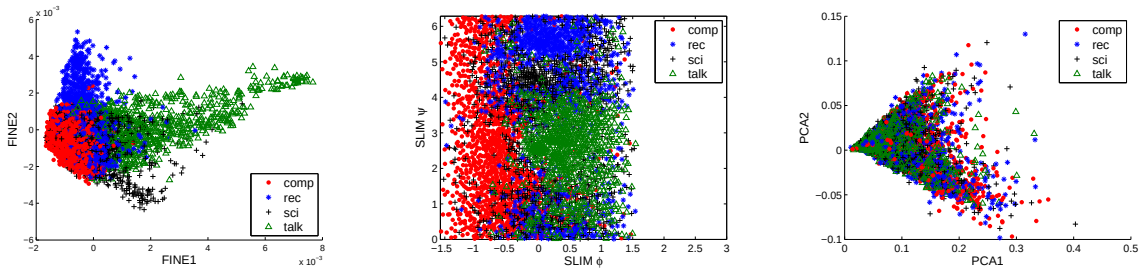
$$\hat{p}(x) = \left(\frac{x_1}{\sum_i x_i}, \ldots, \frac{x_N}{\sum_i x_i} \right). \tag{4.7}$$

Note that this term frequency representation of a multinomial distribution is

nonparametric, so although we do not utilize kernel methods for this problem, we still require no explicit parameter estimation. Since this representation is multinomial and highly sparse, we choose the Hellinger distance as our approximation function, recalling that D_H has a monotonic transformation onto the cosine distance D_C, which is a natural metric on a sphere defined by multinomial PDFs.

For illustration, we will utilize the well known 20 Newsgroups data set[1], which is commonly used for testing document classification methods. This set contains word counts for postings on 20 separate newsgroups. We choose to restrict our simulation to the 4 domains with the largest number of sub-domains (comp.*, rec.*, sci.*, and talk.*), and wish to classify each posting by its highest level domain. Specifically we are given $\mathcal{P} = \{p_1, \ldots, p_N\}$ where each p_i corresponds to a single newsgroup posting and is estimated with (4.7). We note that the data was preprocessed to remove all words that occur in 5 or less documents[2].

Unsupervised FINE



(a) FINE (b) SLIM (c) PCA

Figure 4.8: 2-dimensional embeddings of 20 Newsgroups data. The data displays some natural clustering in the information based embeddings, while the PCA embedding does not distinguish between classes.

First, we utilize unsupervised methods to see if a natural geometry exists between domains. Using FINE (with LEM) and SLIM, we find 2-dimensional embeddings of

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/
[2] http://www.cs.uiuc.edu/homes/dengcai2/Data/TextData.html

$\mathcal{P}$. Figure 4.8 shows the natural geometric separation between the different document classes, although there is some expected overlap. Note that in the SLIM embedding 4.8(b), we plot in spherical coordinates. Contrarily, a PCA embedding (Fig. 4.8(c)) does not demonstrate the same natural clustering. PCA is often used as a means to lower the dimension of data for learning problems due to its optimality for Euclidean data. However, the PCA embedding of the 20 Newsgroups set does not exhibit any natural class separation due to the non-Euclidean nature of the data.

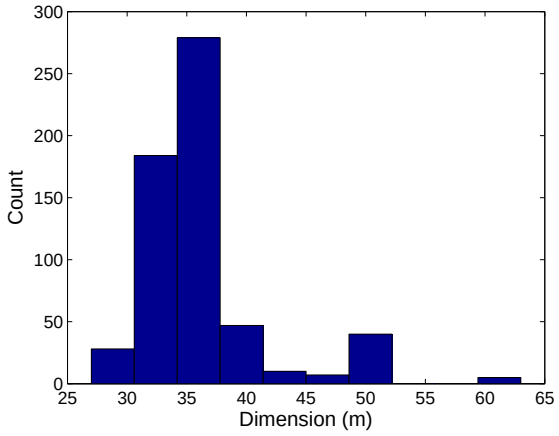


Figure 4.9: Local dimension estimates for each document from a random subset of 600 documents in the 20 Newsgroups data set.

We now compare the classification performance of FINE to that of PCA. In the case of document classification, dimensionality reduction is important as the natural dimension (i.e. number of words) for the 20 Newsgroups data set is $26,214$. Using the k-NN algorithm for local intrinsic dimension estimation (with no smoothing), Fig. 4.9 shows the histogram of the true dimensionality of the sample documents; we test performance for low-dimensional embeddings $\mathcal{P} \to \mathbb{R}^d$ for $d \in [5, 50]$. Following each embedding, we apply an SVM with a linear kernel to classify the data in an 'all-vs-all' setting (i.e. classify each test sample as one of 4 different potential classes in a single event). The training and test sets were separated according to the recommended

indices, and each set was randomly sub-sampled for computational purposes, keeping the ratio of training to test samples constant (400 training samples, 200 test samples). Both the FINE and PCA settings jointly embed the training and test sets.

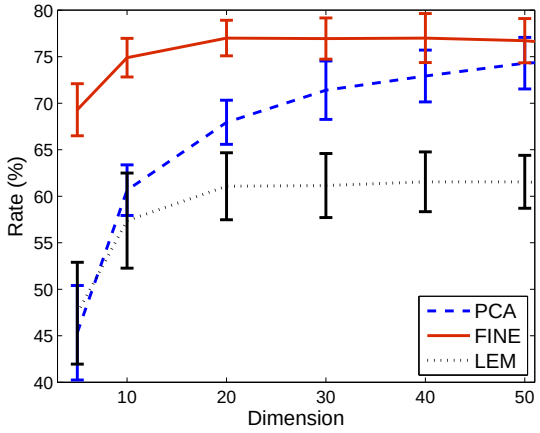


Figure 4.10: Classification rates for low-dimensional embedding using different methods for dimensionality reduction. 1-standard deviation confidence intervals shown over 20-fold cross validation.

Figure 4.10 illustrates that the embedding calculated with FINE outperforms using PCA as a means of dimensionality reduction. The classification rates are shown with a 1-standard deviation confidence interval, and FINE with a dimension as low as $d = 10$ generates results comparable to those of a PCA embedding with $d = 50$. To ease any concerns that Laplacian Eigenmaps (LEM) is simply a better method for embedding these multinomial PDFs, we calculated an embedding with LEM in which each PDF was viewed as a Euclidean vector with the L_2-distance used as a dissimilarity metric. This form of embedding performed much worse than the information based embedding using the same form of dimensionality reduction and the same linear kernel SVM, while comparable to the PCA embedding in very low dimensions.

Supervised FINE

If we allow FINE to use supervised methods for embedding, we can improve classification performance. By embedding with Classification Constrained Dimensionality Reduction (CCDR) [67], which is essentially LEM with an additional tuning parameter defining the emphasis on class labels in the embedding, we hope to improve classification performance and compete with leading methods. We now compare FINE to the diffusion kernels developed by Lafferty and Lebanon [53] for the purpose of document classification. The diffusion kernels method uses the full term-frequency representation of the data and does not utilize any dimensionality reduction. We stress this difference to determine whether or not using FINE for dimensionality reduction can generate comparable results.

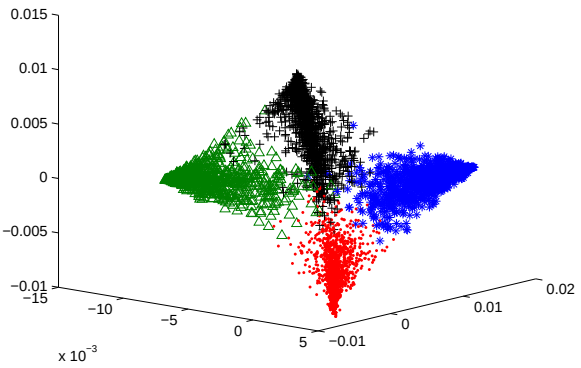


Figure 4.11: 3-dimensional embedding of 20 Newsgroups corpus using FINE in a supervised manner.

We first illustrate the classification performance in a 'one vs. all' setting, in which all samples from a single class were given a positive label (i.e. 1) and all remaining samples were labeled negatively (i.e. -1). In the FINE setting, we first subsampled from the training and test sets, using a test set size of 200, then used CCDR to embed the entire data set into $\mathbb{R}^d$, with $d \in [5, 95]$ chosen to maximize classification

		FINE		Diffusion Kernels	
Task	L	Mean	STD	Mean	STD
comp.*	40	**82.3750**	4.1003	75.5750	3.9413
	80	**85.8250**	2.8713	83.0250	3.4469
	120	**87.6000**	2.0876	85.5750	3.2129
	200	**87.9750**	2.3978	87.8500	2.2775
	400	**89.8000**	2.0926	89.6250	1.9992
	600	90.6500	2.0970	**91.3000**	2.4677
	1000	91.3000	2.3864	**91.9000**	2.2572
rec.*	40	**82.3500**	3.2610	76.2000	3.1514
	80	**86.3500**	2.0462	82.0000	3.8251
	120	**87.1500**	2.3345	83.1250	3.9599
	200	**89.5500**	1.4133	86.8750	2.1143
	400	**91.4750**	2.2152	90.7000	2.0545
	600	92.7500	1.2722	**93.1000**	2.0494
	1000	93.2000	1.3318	**94.6250**	1.4223
sci.*	40	**78.6500**	2.8102	76.3250	3.2898
	80	**80.3750**	3.3280	77.4750	4.2286
	120	**81.5250**	2.8722	78.2250	3.1518
	200	**83.4000**	2.9585	82.2000	3.0236
	400	86.1750	2.2021	**86.2000**	2.2325
	600	**87.1750**	2.9212	87.0500	2.9731
	1000	89.3000	2.3022	**89.8000**	2.2384
talk.*	40	**89.1250**	3.1241	82.2750	2.9131
	80	**90.4250**	2.8895	85.9250	3.6859
	120	**91.1250**	2.5745	86.5500	4.0161
	200	**92.6500**	1.8503	89.7750	3.1518
	400	**93.1000**	1.9775	92.4750	2.1672
	600	**94.7500**	1.3908	94.3750	1.5634
	1000	**94.8500**	1.5483	**94.8500**	1.4244

Table 4.2: Experimental results on 20 Newsgroups corpus, comparing FINE using CCDR and a linear SVM to a multinomial diffusion kernel based SVM. The performance (classification rate in %) is reported as mean and standard deviation for different training set sizes L, over a 20-fold cross validation.

performance. The classification task was performed using a simple linear kernel SVM,

$$K(X, Y) = X \cdot Y.$$

For the diffusion kernels setting,

$$K(X, Y) = (4\pi t)^{-\frac{n}{2}} \exp\left(-\frac{1}{t} \arccos^2\left(\sqrt{X} \cdot \sqrt{Y}\right)\right),$$

we chose parameter value t which optimized the classification performance at each iteration. The experimental results of performance versus training set size, with

20-fold cross validation, are shown in Table 4.2, where the highest performance at each range is emphasized. FINE shows a significant performance increase over the diffusion kernels method for sets with low sample size. As the sample size increases, however, the gap in performance between the diffusion kernels method and FINE decreases, with diffusion kernels eventually surpassing FINE.

We now modify the classification task from a 'one vs. all' to an 'all vs. all' setting, in which each class is given a different label and the task is to assign each test sample to a specific class. Classification rates are defined as the number of correctly classified test samples divided by the total number of test samples (kept constant at 200). The structure of the experiment is otherwise identical to the 'one vs. all' setting. We once again notice in Fig. 4.12 that FINE outperforms the diffusion kernels method for low sample sizes. The point at which the diffusion kernels method surpasses FINE has decreased (i.e. $L \approx 200$ for 'all vs. all' compared to $L \approx 600$ for 'one vs. all'), yet FINE is still competitive as the sample size increases.

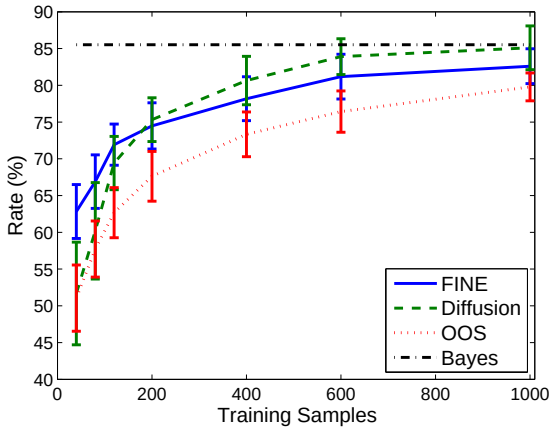


Figure 4.12: Classification rates for low-dimensional embedding with FINE using CCDR vs Diffusion kernels. The classification task was all vs. all. Rates are plotted versus number of training samples. Confidence intervals are shown at one standard deviation. For comparison to the joint embedding (FINE), we also plot the performance of FINE using out of sample extension (OOS). The optimal Bayes classification rate is also displayed.

Note that we also plot the optimal Bayes classification rate, which is calculated

through an upper bound with the min-max Chernoff criteria [43]. This defines probability of classification error as

$$P_e^* \leqslant \max_{s,i\neq j} \pi_i^s \pi_j^{1-s} \exp\left(-D_{CH}(p_i, p_j; s)\right),$$

where $\{p_i\}$ are class PDFs, π_i are the class probabilities, and

$$D_{CH}(p_i, p_j; s) = -\log \int p_i^s(x) p_j^{1-s}(x)\, dx$$

is the Chernoff distance between PDFs. We see that both FINE and the diffusion kernels method approach the performance of the optimal Bayes classifier as the training set increases in size. We utilized the entire training set – 2407 samples – to estimate class PDFs when calculating the optimal Bayes performance.

While our focus when using FINE has been on jointly embedding both the training and test samples (while keeping the test samples unlabeled), Fig. 4.12 also illustrates the use of out of sample extension (OOS) [66] with FINE. In this scenario, the training samples are embedded as normal with CCDR, while the test samples are embedded into the low-dimensional space using interpolation. This setting allows for a significant decrease in computational complexity given the fact that the FINE embedding has already been determined for the training samples (e.g. new test samples are received). A decrease in performance exists when compared to the jointly embedded FINE, which is reduced as the number of training samples increases.

Analysis of the results in both the 'one vs. all' and 'all vs. all' cases shows that FINE can improve upon the deficiencies of the diffusion kernels method in the low sample size region. By viewing each document as a coarse approximation of the overriding class PDF, it is easy to see that, for low sample sizes, the estimate of the within class PDF generated by the diffusion kernels will be highly variable, which leads to poor performance. By reducing the dimension with FINE, the variance is

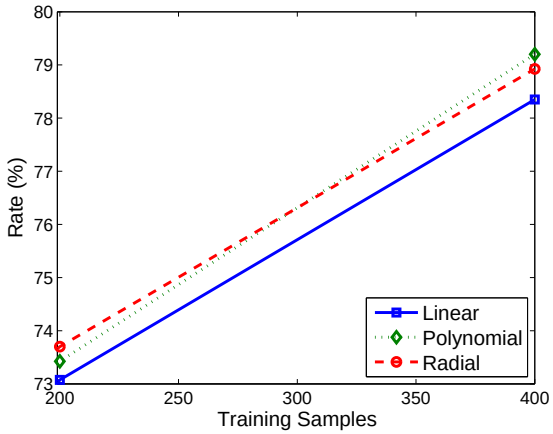


Figure 4.13: Comparison of classification performance on the 20 Newsgroups data set with FINE using different SVM kernels; one linear and two non-linear (2^{nd} polynomial and radial basis function).

limited to significantly fewer dimensions, enabling documents within each class to be drawn nearer to one another. While this could also bring the classes closer to each other, the utilization of CCDR ensures class separation. This results in better classification performance than using the entire multinomial distribution. As the number of training samples increases, the effect of dimensionality is reduced, which allows the diffusion kernels to better approximate the multinomial PDF representative of each class. This reduction in variance across all dimensions ensures that a few anomalous documents will not have the same drastic effect as they would in the low sample size region, resulting in over-fitting. Hence, the performance gain surpasses that of FINE, due to the fact that the *curse of dimensionality* was alleviated elsewhere (i.e. increase in sample size). We note that while FINE performs slightly worse than diffusion kernels in the large sample size region, it still performs competitively with a leading classification method which utilizes the full dimensional data.

An additional reason for the diffusion kernels improved performance over FINE in the large sample size region is that we have restricted FINE to using a linear kernel for this experiment, while the diffusion kernels method is very non-linear. We do this to

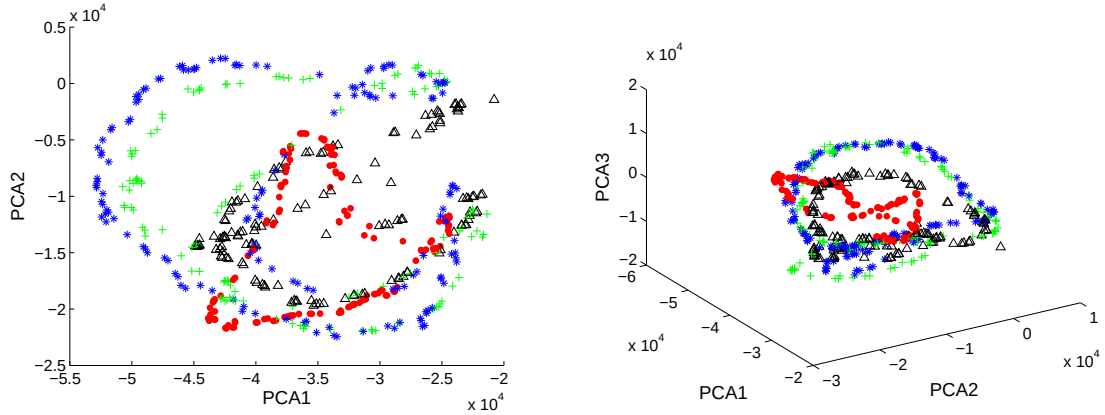
show that even a simple linear classifier can perform admirably in the FINE reduced space. Using a non-linear kernel would show increased performance with FINE. This is illustrated in Fig. 4.13, where we compare the performance of FINE using an SVM classifier with a linear kernel ($K(X,Y) = X^TY$), 2$^{\text{nd}}$ degree polynomial kernel ($K(X,Y) = (\gamma X^TY)^2$), and a radial basis function kernel ($K(X,Y) = \exp(-\gamma|X - Y|^2)$), where γ is a weighting constant. For visualization purposes, we show the results for only a subset of the training sample range (i.e. $L = [200, 400]$), but it is clear that the use of non-linear kernels improves the performance of FINE. The problem of which of the many possible non-linear kernels is optimal remains open and is a subject for future work.

4.6.3 Object Recognition

The problem of object recognition from image sets is similar to the standard classification task. One is given a collection of training data $\mathcal{I} = \{(\boldsymbol{I}^1, y_1), \ldots, (\boldsymbol{I}^N, y_N)\}$, where $\boldsymbol{I}^i = [I_1, \ldots, I_{n_i}]$ is a set of n_i images $\{I_j\}$ of the same object. These images may be captured at different vantage points, showcasing different attributes of the object. We wish to classify an unknown set of images $\boldsymbol{I}$ with some function $f(\boldsymbol{I}) : \boldsymbol{I} \rightarrow y$ [13].

Let us first illustrate the potential difficulties with this problem. Let $\mathcal{I}$ be a collection of $\sim$ 150 image captures each of $N = 4$ unique objects. Each image is taken at a different angle, holding pitch constant while rotating the yaw (full details of image requisition will be described shortly). We use principal component analysis (PCA) on the entire collection of rasterized images (i.e. $\boldsymbol{X} = [\boldsymbol{I}^1, \ldots, \boldsymbol{I}^N]$) to project each image onto the first 2 and 3 principal components of $\boldsymbol{X}$; Fig. 4.14 shows these results. One can naturally see a path formed which demonstrates the natural transition from one image taken at one yaw to the next taken with a slight change

in yaw. It is also clear that the paths which different objects take are very similar, which would make it difficult to distinguish one from the other in most cases. Add that in practice, there may be $\ll 150$ available images per object, and the problem of differentiating image sets (i.e. recognition) becomes very difficult.



(a) First 2 PCs (b) First 3 PCs

Figure 4.14: Projected each image onto the first principal components (PCs). It is clear that there is some trajectory which is followed by each object, corresponding to the change in yaw in each image.

Looking at the trajectories, however, it becomes apparent that there is some generative model which governs the path. While any given point in an object trajectory may be difficult to distinguish from the path of a different object, the entire path maybe more easily discerned. We take a statistical approach by modeling each trajectory as a probability density function, which allows for an information-geometric framework to the problem. Specifically, given $\mathcal{I}$ as training data, and we may estimate the PDFs of each $\boldsymbol{I}^i$ as $p_i(\boldsymbol{I})$, for $i \in [1, N]$. This is performed using a KDE on the rasterized image sets. Once the object class PDFs are estimated with the training data, test sets are classified by minimizing the information divergence between test and training sets. Let $\boldsymbol{I}$ be a test image set with estimated PDF $p(\boldsymbol{I})$, our classifier

$y = f(\boldsymbol{I})$ is

$$f(\boldsymbol{I}) = \arg\min_i G(p(\boldsymbol{I}), p_i(\boldsymbol{I}); \mathcal{P}), \tag{4.8}$$

where $\mathcal{P} = \{p_i\}$. This may be essentially viewed as a 1-nearest neighbor classifier, using the information divergence as an appropriate metric. For the purposes of this simulation, we utilize the Hellinger distance as our metric, due to the stability it provides by being a bounded measure.

With the denoted setup (4.8), we are operating in a semi-supervised framework, as test samples are utilized to approximate the geodesic $G(p(\boldsymbol{I}), p_i(\boldsymbol{I}); \mathcal{P})$. If this is undesirable, as in many instances there is not an available collection of test samples, the classification may be performed in the same manner using the strict Hellinger distance $D_H(p(\boldsymbol{I}), p_i(\boldsymbol{I}))$ as opposed to the geodesic approximation.

4.6.4 Data Setup

The data we will analyze was collected at Tech-edge building, in the Air Force Research Laboratory[3]. The experiment was performed with 4 unique objects – 3 different model laptops and an LCD monitor. Each object was positioned on a swiveling desk, with a stationary camera (Canon VB-50iR) located above and to the left side of the object. The desk was then spun by a rope (so that no person is in the scene) and the camera captured still frames of the object at 15 fps with a 640×480 resolution, for roughly 10 seconds. An illustration of these retrieved data sets may be found in Fig. 4.15. Note that for each trial, the object was placed at the same location on the desk, and the desk was spun at an (attempted) equal speed.

Given the lack of unique objects, but the well sampled trajectories of the objects with changes in yaw, we may artificially manufacture "new" realizations of unique

[3]This data collection was partially supported by the AFRL ATR Center through a summer internship of Christine Kyung-min Kim and a Signal Innovations subcontract to the University of Michigan.

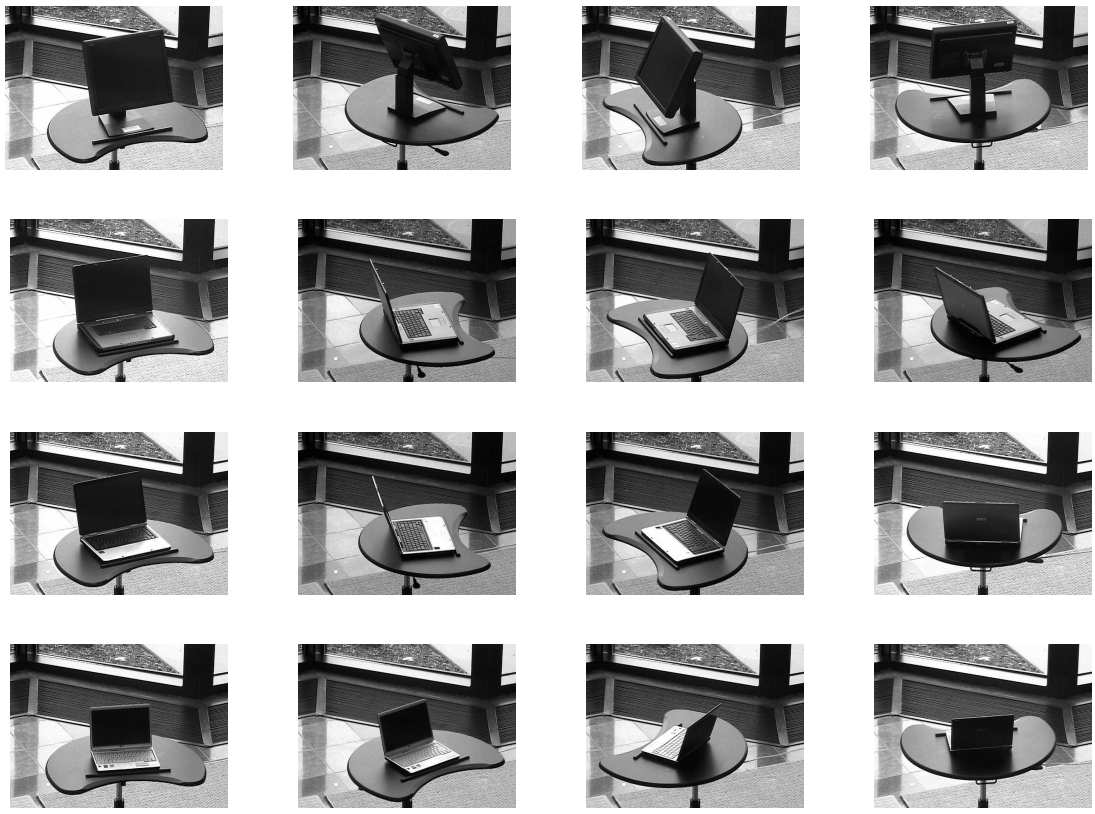


Figure 4.15: Sample images from the image sets. The objects rotate on the table, giving the camera different capture angles. Pitch remained constant while yaw changed with the rotation.

objects by subsampling along the trajectory. Specifically, let $\boldsymbol{I} = [I_1, \ldots, I_n]$, ordered according to change in object yaw, and let l be the sample spacing. Rather than having only 1 image set for the object, we can create n/l image sets by subsampling in the following manner:

$$\boldsymbol{I}^j = [I_j, I_{j+l}, I_{j+2l}, \ldots], \tag{4.9}$$

which generate uniformly spaced, i.i.d. realizations along the yaw trajectory. Although artificially generated, this is statistically equivalent to capturing a sequence of images from identical items which have been positioned differently (with respect to yaw). Note that each manufactured set has entirely unique images, so no two estimated PDFs will be identical. This is key as it simulates the setting for this

object recognition task.

4.6.5 Results

We first wish to study the effect of test sample size on recognition capability. We begin by partitioning our training set to ~ 10 sample images for each of the 4 objects, obtained with subsampling using (4.9). Next, we partition our test set using $\sim N_t$ samples per test object, with $N_t \in [2, 10]$. Given the small sample sizes, we preprocess the data by projecting each image onto the first 10 principal components of the training set. To test recognition capabilities, we use the 1-NN classifier (4.8) and plot the classification error, over a 10-fold cross validation, in Fig. 4.16. We also compare to the method presented in [4], which classifies image sets by maximizing the KL-divergence between test set and training set. Note that we have modified the method to use a KDE rather than GMM for density estimation. While this may cause a minor change in performance, we aim to keep as many factors constant as possible for a fair comparison. Additionally, given the low number of samples we are considering, a GMM offers very little difference to a KDE.

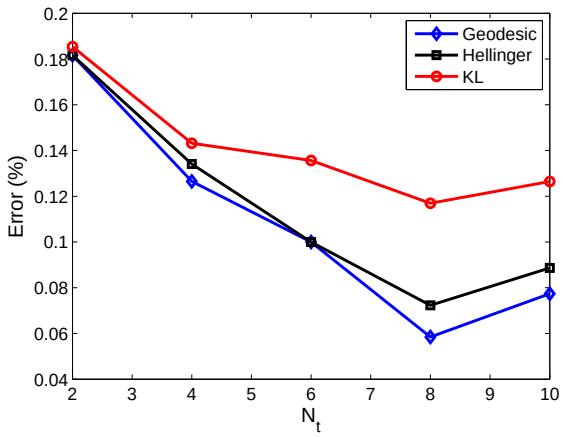
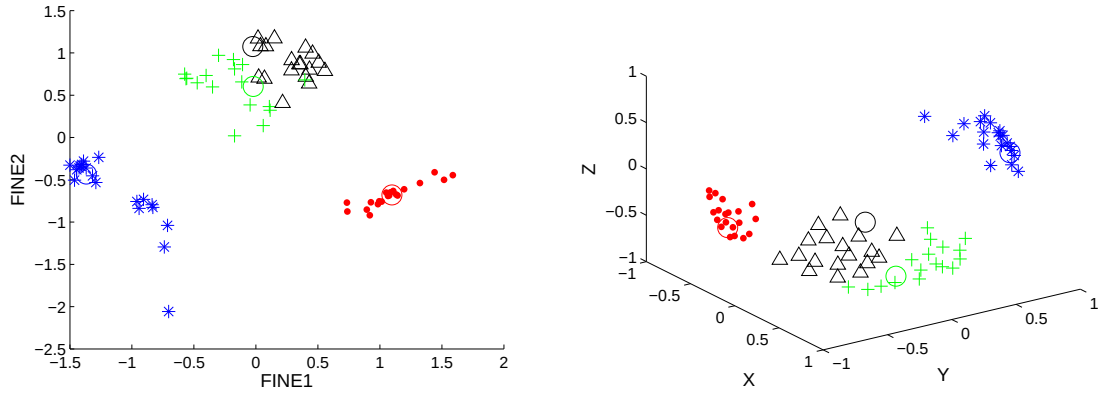


Figure 4.16: Classification error rates for object recognition using different information divergences. The stability of the Hellinger distance for low training set sample sizes shows superior performance, garnering even better rates when using the geodesic approximation.



(a) FINE (b) SLIM

Figure 4.17: Embedding of the image sets with FINE and SLIM. We can see that two of the laptops ($\triangle$ and $+$) are very similar, while the third laptop ($\star$) and LCD monitor ($\cdot$) are clearly separable.

It is clear that the proposed method using the geodesic distance outperforms the KL method. To ease concerns that the performance gain is strictly due to the geodesic distance approximation $G(p, p_i; \mathcal{P})$, which may not be practically available in all cases, we also illustrate classification performance using the strict Hellinger distance $D_H(p, p_i)$. There is a slight decrease in performance, which shows that there is indeed some gain from the geodesic approach, but performance is still far superior to that of the KL-divergence. We believe this is due to the instability of the KL measure, which is highlighted when dealing with low sample size. As the sample size of the training set increases, and the PDFs are better estimated, we believe both methods would perform comparably.

Finally, we illustrate the embeddings of the data obtained with FINE and SLIM. For this case we used $l = 7$, such that each test image set had roughly 70% the number of sample images as the training sets. The embedding results are shown in Fig. 4.17, and the natural clustering is visually identified. Each point represents a unique image set $\boldsymbol{I}$, and the points corresponding to training sets are denoted with the circle. Note that this embedding was entirely unsupervised. This visualization, which is entirely

based on the natural information-geometry between the image sets, is useful for comparing objects. One may notice that two of the laptop image sets are similarly embedded, while the other two are clearly separated. It is logical that the points corresponding to the LCD monitor lie furthest away from the points representing laptop image sets, as they are the most dissimilar. We can not visually decipher the reason 2 laptops seem so close, but note that they are still distinguishable even in 2 dimensions, even more so in 3 dimensions.

Orientation Angle Recognition

Let us now briefly present an additional application for SLIM on the available data. Specifically, looking at Fig. 4.14 we noted that there was an apparent trajectory for which the images progressed. This corresponded to the change in yaw angle when capturing the sequence. Given that this angle changes as a function of a rotation, the angle will eventually 'reset' as the rotation continues, as it exists in $[0, 2\pi)$. Hence, embedding each individual image onto the unit circle, or surface of the unit sphere, should properly illustrate this trajectory.

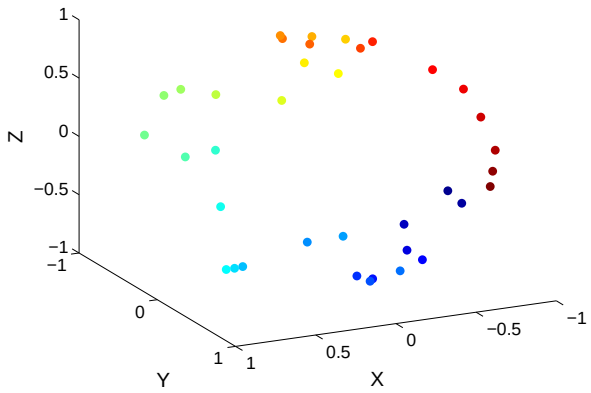


Figure 4.18: By embedding each image on the sphere with SLIM, we can see the clear rotational trajectory (denoted by change in color) that is taken by the image capturing system.

We proceed by sampling a portion of the image trajectory for a single object, cor-

responding to one complete rotation in yaw (37 images total). We then characterize each rasterized image I_i as a multinomial distribution over the entire pixel space, such that

$$p_i(I) = \left[\frac{I_i(1)}{\sum_j I_i(j)}, \ldots, \frac{I_i(m)}{\sum_j I_i(m)} \right]^T,$$

where m is the length of I_i ($m = 307200$ in this case). Given these multinomial PDFs, we calculate the pairwise Hellinger distances and implement SLIM. Results are illustrated in Fig. 4.18, where we see the clear trajectory which governs the images. Colors are applied sequentially to the points so one can view the order for which the path takes (starting at blue and ending at red).

While this work is currently in its infancy, we present it here to illustrate the potential power of this approach. Specifically, given a constant rotation radius, any image that is captured of an object may be mapped to the surface of a sphere. Given a database of potential objects taken at different vantage points (in both pitch and yaw), we believe SLIM may be used to determine the exact orientation angle of the capturing device. This is parallel to determining the orientation of the object given a known camera location, and could prove very helpful in task of recognizing an object from a single image.

4.7 Conclusions and Future Work

The assumption that high-dimensional data lies on a Riemannian sub-manifold of Euclidean space is often based on the ease of implementation due to the wealth of knowledge and methods based on Euclidean space. This assumption is not viable in many problems of practical interest, as there is often no straightforward and meaningful Euclidean representation of the data. In these situations it may be more appropriate to assume the data is a realization of some PDF lying on a statistical

manifold. Using information geometry, we have shown the ability to find a low-dimensional embedding of the manifold, which allows us to not only find the natural separation of the data, but to also reconstruct the original manifold and visualize it in a low-dimensional Euclidean space. This allows the use of many well known learning techniques which work based on the assumption of Euclidean data.

We have illustrated the ability of both FINE and SLIM to be used in synthetic and practical applications; for visualization, clustering, and classification. We will present additional applications in proceeding chapters. In future work, we plan to utilize different classification methods (such as k-NN and using different SVM kernels) to maximize our document classification performance. We also plan to continue studies on the effect of using out of sample extension on our performance. Additionally, we aim to continue with the idea of determining object orientation angle with SLIM. Lastly, we will continue to find applications which fit the setting for FINE and SLIM, such as internet anomaly detection and face recognition, and determine whether or not these problems would benefit from our framework.

4-A Appendix: Gradient Descent

Gradient descent (or the method of *steepest* descent) allows for the solution of convex optimization problems by traversing a surface or curve in the direction of greatest change, iterating until the minimum is reached (gradient *ascent* searches for the maximum). Specifically, let $J(x)$ be a real-valued objective function which is differentiable about some point x_i. The direction in which $J(x)$ decreases the fastest, from the point x_i, is that of the negative gradient of J at x_i, $-\frac{\partial}{\partial x}J(x_i)$. By calculating the location of the next iteration point as

$$x_{i+1} = x_i - \mu \frac{\partial}{\partial x} J(x_i),$$

where μ is a small number regulating the step size, we ensure that $J(x_i) \geqslant J(x_{i+1})$. Continued iterations will result in $J(x)$ converging to a local minimum. Gradient descent does not guarantee that the process will converge to an absolute minimum, so typically it is important to initialize x_0 near the estimated minimum. Note that if gradient ascent is desired, the optimization is solved as

$$x_{i+1} = x_i + \mu \frac{\partial}{\partial x} J(x_i).$$

4-B Appendix: SLIM Gradient Calculation

We now derive the gradient for the SLIM algorithm. Recall the objective function in which we maximize

$$J = \sum_i \sum_j D_{S^2}(\theta_i, \theta_j)^\gamma - W_{ij} D_{S^2}(\theta_i, \theta_j), \tag{4.10}$$

which may be further described as

$$J = J_1 - J_2,$$

where

$$J_1 = \sum_i \sum_j D_{S^2}(\theta_i, \theta_j)^\gamma$$

is the power-weighted graph length and

$$J_2 = \sum_i \sum_j W_{ij} D_{S^2}(\theta_i, \theta_j)$$

is the multiplicative-weighted graph length. As the gradient operator is linear, we may solve

$$\frac{\partial}{\partial \Theta} J = \frac{\partial}{\partial \Theta} J_1 - \frac{\partial}{\partial \Theta} J_2. \tag{4.11}$$

For ease of notation, define

$$D_{S^2}(\theta_i, \theta_j) = \arccos(f(\theta_i, \theta_j)),$$

where

$$f(\theta_i, \theta_j) = \cos(\phi_i)\cos(\phi_j)\cos(\psi_i - \psi_j) + \sin(\phi_i)\sin(\phi_j).$$

Let us now derive the gradient quantities individually:

$$\begin{aligned}
\frac{\partial}{\partial \phi_i} J_1 &= \sum_j \arccos(f(\theta_i, \theta_j))^{\gamma-1} \frac{-2}{\sqrt{1 - f(\theta_i, \theta_j)^2}} \times \\
&\quad (-\sin\phi_i \cos\phi_j \cos(\psi_i - \psi_j) + \cos\phi_i \sin\phi_j) \\
\frac{\partial}{\partial \psi_i} J_1 &= \sum_j \arccos(f(\theta_i, \theta_j))^{\gamma-1} \frac{-2}{\sqrt{1 - f(\theta_i, \theta_j)^2}} \times \\
&\quad (\cos\phi_i \cos\phi_j (-\sin\psi_i \cos\psi_j + \cos\psi_i \sin\psi_j) + \cos\phi_i \sin\phi_j) \\
\frac{\partial}{\partial \phi_i} J_2 &= \sum_j W_{ij} \frac{-2}{\sqrt{1 - f(\theta_i, \theta_j)^2}} (-\sin\phi_i \cos\phi_j \cos(\psi_i - \psi_j) + \cos\phi_i \sin\phi_j) \\
\frac{\partial}{\partial \psi_i} J_2 &= \sum_j W_{ij} \frac{-2}{\sqrt{1 - f(\theta_i, \theta_j)^2}} (\cos\phi_i \cos\phi_j (-\sin\psi_i \cos\psi_j + \cos\psi_i \sin\psi_j) + \\
&\quad \cos\phi_i \sin\phi_j)
\end{aligned} \tag{4.12}$$

Hence, we may define the gradient of the constrained objective function by substituting the equations in (4.12) into (4.11) for each element of the matrix $\Theta = [\theta_1, \ldots, \theta_N]$, where $\theta_i = [\phi_i, \psi_i]^T$.

CHAPTER V

Information Preserving Component Analysis

5.1 Introduction

Consider a signal $\boldsymbol{X} = [x_1, \ldots, x_n]$ in which each $x_i \in \mathbb{R}^d$. For many learning methods, it is often desirable to reduce the dimensionality of $\boldsymbol{X}$, finding a transformation $A : \boldsymbol{X} \to \boldsymbol{Y}$ where $\boldsymbol{Y} = [y_1, \ldots, y_n]$ and each $y_i \in \mathbb{R}^m$, $m < d$. This is standard fare for manifold learning yielding the non-linear embedding methods thoroughly discussed in previous chapters [5, 71, 74], and linear projection methods such as principal component analysis [37] and independent component analysis (ICA) [46].

Now consider a collection of signals $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ in which dimensionality reduction is still desired. Typically, each set would be reduced in an individual manner. If there is deemed a relationship between the sets, it has generally been approached as a classification problem in which each signal $\boldsymbol{X}_i$ is considered a set of points belonging to class i. In this setting, Fisher's linear discriminant analysis (LDA) [35,42,63] methods are typically used for supervised dimensionality reduction.

What if we view this problem in a different light; rather than considering each $\boldsymbol{X}_i$ to be a collection of points in a specific class, let us generalize the relationship between sets $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$. Specifically, consider the case for which each $\boldsymbol{X}_i$ is a realization of some unknown generating function p_i, in which p_i and p_j may or may

not be equivalent. This agrees with the standard classification problem, in which each p_i represents a class PDF, but it also allows for different relationships between PDFs. Specifically, rather than having a number classes equal to the number of data sets N, there may be significantly fewer classes $M \ll N$, in which M is unknown and no labels are available. Dimensionality reduction is still desirable, however, for the purposes of feature extraction and visualization rather than classification.

Let us illustrate with a simple example. Suppose a census is performed in each state generating a collection of data about each of its residents such as height, weight, income, ethnicity, education level, etc.. Standard methods of feature extraction will find the features which best describe each state on an individual level. We are interested in determining the most important features when comparing all states at the same time. While ethnicity may not be a distinguishing characteristic within the state of Wyoming, and may not be recognized as such when solely extracting features from that individual state, it would be quite informative when comparing all 50 states. Hence, we desire to find a method of dimensionality reduction which best relates a collection of signals, one to another.

In this chapter we propose a linear method of dimensionality reduction – which we refer to as *Information Preserving Component Analysis (IPCA)* – that preserves probabilistic similarities between multiple related data sets. Rather than making Euclidean assumptions on the data, we characterize the data as a realization of some generative model, or probability distribution, and use the Fisher information distance as a similarity measure between sets, which we aim to preserve in the low dimension. Whereas standard methods of dimensionality reduction aim to find some optimal transformation of data points $A : x \to y$, where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$ $(m < d)$, IPCA aims to find the optimal transformation of PDFs $A : p(x) \to p(y)$. By preserving

the Fisher information distance between the estimated PDFs generating the data sets, IPCA ensures that the low-dimensional representation maintains the similarities between data sets which are contained in the full-dimensional data, minimizing the loss of information.

We will show that IPCA can be used both in the unsupervised and supervised frameworks of dimensionality reduction. In the unsupervised case, the projection onto the same low-dimensional subspace enables a visual comparison between various related signals in a manner that could not be done by projecting each individually, which would create a unique subspace for each signal. When using dimensionality reduction for the classification task, IPCA offers a subspace which minimizes the upper bound on probability of classification error. In both cases, analysis of the loading vectors within the IPCA projection matrix offers a form of feature extraction, identifying which variables are most important towards information preservation, which has the significant benefit of allowing for exploratory data analysis.

The remainder of this chapter proceeds as follows: We present our methods for finding the unsupervised IPCA projection in Section 5.2, followed by an adaptation to the supervised case in Section 5.3. Simulation results for synthetic data, spam (i.e. unsolicited email) analysis, and soil imagery classification are shown in Section 5.4, followed by a discussion and areas for future work in Section 5.5.

5.2 Unsupervised IPCA

Rather than focusing on the relationships between elements in a single data set $\boldsymbol{X}$, it is often desirable to compare each set in a collection $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ in which $\boldsymbol{X}_i$ has n_i elements $x \in \mathbb{R}^d$. We can define a similarity between data sets $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ with the Fisher information distance as $D_F(p_i, p_j)$, in which p_i is the estimated

PDF of set $\boldsymbol{X}_i$. With an abuse of notation, we will further refer to $D_F(p_i, p_j)$ as $D_F(\boldsymbol{X}_i, \boldsymbol{X}_j)$ with the knowledge that the Fisher information distance is calculated with respect to PDFs, not realizations.

We define the *Information Preserving Component Analysis (IPCA)* projection matrix $A \in \mathbb{R}^{m \times d}$, in which A reduces the dimension of $\boldsymbol{X}$ from d to m ($m \leqslant d$), such that

$$D_F(A\boldsymbol{X}_i, A\boldsymbol{X}_j) = D_F(\boldsymbol{X}_i, \boldsymbol{X}_j), \; \forall \, i, j. \tag{5.1}$$

Formatting as an optimization problem, we would like to solve:

$$A = \arg \min_{A: AA^T = I} J(A), \tag{5.2}$$

where I is the identity matrix and $J(A)$ is some cost function designed to implement (5.1). Note that we include the optimization constraint $AA^T = I$ to ensure our projection is orthonormal, which keeps the data from scaling or skewing as that would undesirably distort the data. Let $D(\mathcal{X})$ be a dissimilarity matrix such that $D_{ij}(\mathcal{X}) = D_F(\boldsymbol{X}_i, \boldsymbol{X}_j)$, and $D(\mathcal{X}; A)$ is a similar matrix where the elements are perturbed by A, i.e. $D_{ij}(\mathcal{X}; A) = D_F(A\boldsymbol{X}_i, A\boldsymbol{X}_j)$. We formulate the following cost function:

$$J(A) = \sum_i \sum_j W_{ij} \left(D_{ij}(\mathcal{X}) - D_{ij}(\mathcal{X}; A) \right)^2, \tag{5.3}$$

where W_{ij} is some weighting factor.

It should be clear that setting $W_{ij} = 1, \forall i, j$ is a direct implementation of our stated objective. We may modify the weights, however, to apply a sense of locality to our objective, which is useful give that the approximations to the Fisher information distance are valid only as $p \to q$. In problems in which PDFs may significantly differ, this will prevent the algorithm from being unnecessarily biased by PDFs which are

very far away. Specifically, we may define weights using an exponential heat kernel

$$W_{ij} = \exp\left(-D_{ij}(\mathcal{X})/c\right),$$

where c is some constant. This will ensure that the PDFs which are "close" are given more weight than those for which the Fisher information distance approximation is weak. We may also define our weights as $W_{ij} = 1$ if $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are neighbors – either defined through some ϵ-ball or k-NN graph. This weighting again adds a sense of locality, but makes no attempt to preserve "far" distances, rather than diminishing their importance as with the heat kernel weighting.

While the choice of cost weighting function is dependent on the problem, the overall projection method ensures that the similarity between data sets is maximally preserved in the desired low-dimensional space, allowing for comparative learning between sets.

5.2.1 Optimization

Using gradient descent, we are able to solve (5.2). Specifically, let $J(A) = \sum_i \sum_j W_{ij} \left(D_{ij}(\mathcal{X}) - D_{ij}(\mathcal{X}; A)\right)^2$ be our objective function, measuring the weighted squared error between our projected subspace and the full-dimensional space. The direction of the gradient is solved by taking the partial derivative of J w.r.t. a projection matrix A,

$$\frac{\partial}{\partial A} J(A) = \sum_i \sum_j W_{ij} \frac{\partial}{\partial A} \left[D_{ij}^2(\mathcal{X}; A) - 2D_{ij}(\mathcal{X}) D_{ij}(\mathcal{X}; A)\right],$$

which is further evaluated as

$$\frac{\partial}{\partial A} J(A) = \sum_i \sum_j 2W_{ij} \left(D_{ij}(\mathcal{X}; A) - D_{ij}(\mathcal{X})\right) \frac{\partial}{\partial A} D_{ij}(\mathcal{X}; A).$$

In Appendix B.2 we provide the specific numerical formulation of the gradient, for both the Hellinger distance and KL-divergence, as well as a general formulation for

information divergences.

Given the direction of the gradient, the projection matrix can be updated as

$$A = A - \mu \frac{\partial}{\partial A} \tilde{J}(A), \tag{5.4}$$

where

$$\frac{\partial}{\partial A} \tilde{J}(A) = \frac{\partial}{\partial A} J(A) + Q_0 A + \mu Q_1 A$$

is the direction of the gradient, constrained to force A to remain orthonormal. Variables Q_0 and Q_1 are defined as:

$$Q_0 = -\frac{1}{2} \left(\left(\frac{\partial}{\partial A} J(A) \right) A^T + A \left(\frac{\partial}{\partial A} J(A) \right)^T \right)$$

$$Q_1 = \frac{1}{2} \left(\frac{\partial}{\partial A} J(A) + Q_0 A \right) \left(\frac{\partial}{\partial A} J(A) + Q_0 A \right)^T.$$

The full derivation of this constraint can be found in Appendix 5-A. This process is iterated until the error $J(A)$ converges.

5.2.2 IPCA Algorithm

Algorithm 5.1. Unsupervised Information Preserving Component Analysis

Input: Collection of data sets $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ in $\mathbb{R}^d$; the desired projection dimension m; search step size μ

1: Calculate $D(\mathcal{X})$, the Fisher information distance matrix

2: Calculate W, the pairwise weight matrix

3: Initialize $A_1 \in \mathbb{R}^{m \times d}$ as some orthonormal projection matrix

4: Calculate $D(\mathcal{X}; A_1)$, the Fisher information distance matrix in the projected space

5: $l = 1$

6: **while** $|J_l - J_{l-1}| > \epsilon$ **do**

7: Calculate $\frac{\partial}{\partial A_l}\tilde{J}$, the direction of the gradient, constrained to $AA^T = I$

8: $A_{l+1} = A_l - \mu\frac{\partial}{\partial A_l}\tilde{J}$

9: Calculate $D(\mathcal{X}; A_{l+1})$

10: $J = \sum_i \sum_j W_{ij} \left(D_{ij}(\mathcal{X}) - D_{ij}(\mathcal{X}; A_l)\right)^2$

11: $l = l + 1$

12: **end while**

Output: Projection $A \in \mathbb{R}^{m \times d}$, which preserves the information distances between sets in $\mathcal{X}$.

The full method for IPCA is described in Algorithm 5.1. We note that A_1 is often initialized as a random orthonormal projection matrix as to not bias the estimation, but this carries the risk of converging to a local minimum. For certain applications it may be beneficial to initialize near some estimated global minimum if that information is available. At this point we stress that we utilize gradient descent due to its ease of implementation. There may be more efficient methods of optimization, but that is out of the scope of the current contribution and is an area for future work.

5.2.3 Variable Selection

IPCA may be used as a form of variable selection, as the loading vectors in the linear projection matrix A will be appropriately weighted towards the dimensions which best preserve the information distance between sets within the collection. For example, if two multivariate PDFs p and q are independent and identically distributed in a certain dimension, that dimension will offer zero contribution to the information distance between p and q. As such, the information distance is entirely defined by those areas of input space in which p and q differ. When finding a projection which preserves the information distance between p and q, A is going

to be highly weighted towards the variables which contribute most to that distance. Hence, the loading vectors of A essentially give a ranking of the discriminative value of each variable. This form of variable selection is useful in exploratory data analysis.

5.3 Supervised IPCA

While in Section 5.2 we presented IPCA as an unsupervised form of dimensionality reduction, we now make the connection to the supervised framework [20]; forming a direct relation between IPCA and the Chernoff performance bound on classification. Let us first define the classification problem as one of classifying an unknown observation x as one of N potential classes, $\mathcal{C} = \{C_1, \ldots, C_N\}$. We now make the general assumption that the observations from each class are generated by the PDFs $p_1(x), \ldots, p_N(x)$ with the prior probabilities on these classes as $\pi_1, \ldots, \pi_N$. Note we make no restrictions on the complexity of the models $p_i(x)$, so our general assumption holds in all cases. If we define the classifier $f(x) : \mathbb{R}^d \to \mathcal{C}$, the probability of classification error is given by

$$\begin{aligned} P_e &= \sum_{i=1}^{N} \pi_i P(f(x) \neq C_i | C_i) \\ &= \sum_{i=1}^{N} \pi_i \int I(f(x) \neq C_i) p_i(x)\, dx, \end{aligned} \tag{5.5}$$

where $I(\cdot)$ is the standard indicator function. The optimal Bayes classifier is given by

$$f^\star(x) = \arg\max_i \pi_i p_i(x), \tag{5.6}$$

which may be used to determine the minimum error probability

$$P_e^\star = \sum_{i=1}^{N} \int I\left(\max_{j \neq i} \frac{\pi_j p_j(x)}{\pi_i p_i(x)} > 1\right) \pi_i p_i(x)\, dx. \tag{5.7}$$

Note that for the two-class problem, which is easily extended to the multi-class case, (5.7) simplifies to

$$P_e^\star = \int I\left(\frac{\pi_2 p_2(x)}{\pi_1 p_1(x)} > 1\right) \pi_1 p_1(x)\, dx + \int I\left(\frac{\pi_1 p_1(x)}{\pi_2 p_2(x)} > 1\right) \pi_2 p_2(x)\, dx. \tag{5.8}$$

We may bound $P_e^\star$ by

(5.9)

$$P_e^\star \leqslant \pi_2^s \pi_1^{1-s} \exp\left(-D_{ch}(p_1(x), p_2(x); s)\right) + \pi_1^{s'} \pi_2^{1-s'} \exp\left(-D_{CH}(p_2(x), p_1(x); s')\right),$$

where $0 \leqslant s \leqslant 1$ and

$$D_{CH}(p_1(x), p_2(x); s) = -\log \int p_2^s(x) p_1^{1-s}(x)\, dx \tag{5.10}$$

is the Chernoff distance between PDFs $p_1(x)$ and $p_2(x)$ [37]. Note that as D_{CH} increases, the upper bound on the probability of classification error between points in classes C_1 and C_2 decreases.

A special case of the Chernoff distance is when $s = \frac{1}{2}$, and is known as the Bhattacharya distance between PDFs $f(x)$ and $g(x)$,

$$D_B(f, g) = -\log \int \sqrt{f(x) g(x)}\, dx,$$

which has been used to bound the classification error for dimensionality reduction [44]. It is natural, therefore, to find a form of dimensionality reduction which will maximize the Bhattacharya distance between class PDFs, as that will enable control of the error probability. This was done in the parametric sense in [76]. We offer an information geometric approach to the problem, creating a supervised formulation of IPCA.

Specifically, we note that the Bhattacharya distance is a monotonic transformation of the Hellinger distance,

$$D_B(f, g) = -\log\left(1 - \frac{1}{2} D_H^2(f, g)\right).$$

This transformation is important as it allows us to modify our original desire of maximizing the Bhattacharya distance between class PDFs to that of maximizing the Hellinger distance between classes. While seemingly trivial, this transformation is key as it enables us to take an information geometric approach to the problem – with no increase in complexity – due to the convergence of the Hellinger distance to the Fisher information distance.

This information geometric approach fits into the IPCA framework. Consider the following theorem:

Theorem 5.1. *Let RVs $X, X' \in \mathbb{R}^d$ have PDFs f_X and $f_{X'}$, respectively. Using the $m \times d$ matrix A satisfying $AA^T = I_m$, construct RVs $Y, Y' \in \mathbb{R}^m$ such that $Y = AX$ and $Y' = AX'$. The following relation holds:*

$$D_H(f_X, f_{X'}) \geqslant D_H(f_Y, f_{Y'}), \tag{5.11}$$

where f_Y and $f_{Y'}$ are the PDFs of Y, Y', respectively.

The proof of this theorem may be found in Appendix 5-B, with a similar proof for the Kullback-Leibler divergence in Appendix 5-C. What this theorem implies is that maximizing the Hellinger distance in the lower dimensional space is directly related to minimizing the difference (i.e. preserving) between the high and low dimensional distances; they are indeed equivalent statements in the 2 class case. Hence, our objective of finding the projection which maximizes the distance between PDFs is parallel to the objective of preserving the distances between PDFs, albeit with a different formulation. With this knowledge, we will still refer to our supervised framework as IPCA. By maximizing the information distance between class PDFs, we not only ensure an optimal performance bound on classification error, but we also preserve the natural information geometry between classes. This fact is critical

when class PDFs are not linearly separable (e.g. such is the assumption of standard LDA).

5.3.1 Optimization

Let us now define the supervised IPCA projection as one that maximizes the information distance between data sets. Specifically, let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ where $\boldsymbol{X}_i$ consists of all points $x \in \mathbb{R}^d$ in class C_i; estimating the PDF of $\boldsymbol{X}_i$ as $p_i(x)$. Our IPCA formulation now becomes:

$$A = \arg \max_{A:AA^T=I} \sum_i \sum_j W_{ij} D_{ij}(\mathcal{X}; A)^2. \tag{5.12}$$

Note that we use the same gradient methods as described in Section 5.2, with minor modifications. We now employ gradient *ascent* rather than gradient *descent* as objective maximization is now our desire, while our objective function is defined as

$$J(A) = \sum_i \sum_j W_{ij} D_{ij}(\mathcal{X}; A)^2$$

with the direction of the gradient given by

$$\frac{\partial}{\partial A} J(A) = \sum_i \sum_j 2 W_{ij} D_{ij}(\mathcal{X}; A) \frac{\partial}{\partial A} D_{ij}(\mathcal{X}; A).$$

The full method for supervised IPCA, specialized towards the classification task, is described in Algorithm 5.2. Note that this is the same framework as our unsupervised IPCA algorithm (Algorithm 5.1) with objective function changes noted above.

Algorithm 5.2. Supervised Information Preserving Component Analysis

Input: Collection of data classes $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ in $\mathbb{R}^d$; projection dimension m; search step size μ; threshold ϵ

1: Initialize $A_1 \in \mathbb{R}^{m \times d}$ as a random orthonormal projection matrix

2: Calculate W, the pairwise weight matrix

3: Calculate $D(\mathcal{X}; A_1)$, the information distance matrix in the projected space

4: $l = 1$

5: **while** $|J_l - J_{l-1}| > \epsilon$ **do**

6: Calculate $\frac{\partial}{\partial A_l}\tilde{J}$, the direction of the gradient, constrained to $AA^T = I$

7: $A_{l+1} = A_l + \mu\frac{\partial}{\partial A_l}\tilde{J}$

8: Calculate $D(\mathcal{X}; A_{l+1})$

9: $J = \sum_i \sum_j W_{ij} D_{ij}(\mathcal{X}; A_l)^2$

10: $l = l + 1$

11: **end while**

Output: Projection matrix $A \in \mathbb{R}^{m\times d}$, which maximizes the information distances between class PDFs.

5.4 Simulations

We now illustrate the uses of both supervised and unsupervised IPCA. We illustrate the unsupervised case on synthetic data and an analysis of unsolicited email, while we test classification performance using supervised IPCA on satellite imagery data. Note that in all simulations we used weights $W_{ij} = 1$, $\forall i, j$, as we do not apply a sense of locality.

5.4.1 Synthetic Data

As a proof of concept, we now illustrate unsupervised IPCA on a synthetic data set of known structure. An illustration of the data is shown in Fig. 5.1, which is defined as follows: Let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{N_1}, \boldsymbol{X}_{N_1+1}, \ldots, \boldsymbol{X}_{N_1+N_2}\}$ be a collection of sets in which $\boldsymbol{X}_j \in \mathbb{R}^{2\times 400}$ is created by joining two Chi-squared distributions (one flipped about the x-axis). For $j = 1, \ldots, N_1$, let us define $\mathcal{X}_1$ in that fashion while we define $\mathcal{X}_2$ for $j = N_1+1, \ldots, N_1+N_2$ in a similar manner, with the data flipped about

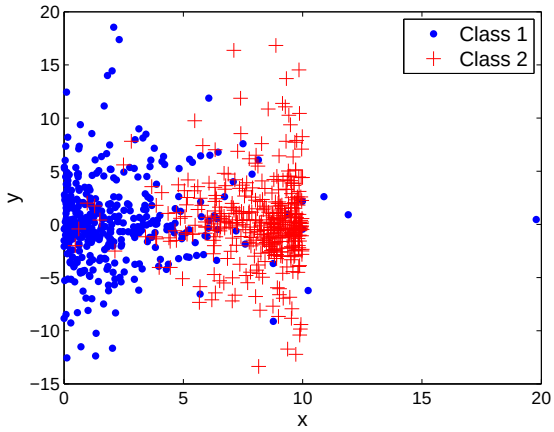


Figure 5.1: An illustration of a sample data set from each class for our synthetic data test. The classes are distributed as 'mirror images' of each other, about the line $x = 5$.

the y-axis and offset by $+10$ units. Essentially, $\mathcal{X}_1$ and $\mathcal{X}_2$ contain 'mirror image' data sets ('mirrored' about the line $x = 5$) with 400 samples each. We wish to find the projection down to a single dimension which optimally preserves the Fisher information between data sets. For this simulation, let $N_1 = N_2 = 5$.

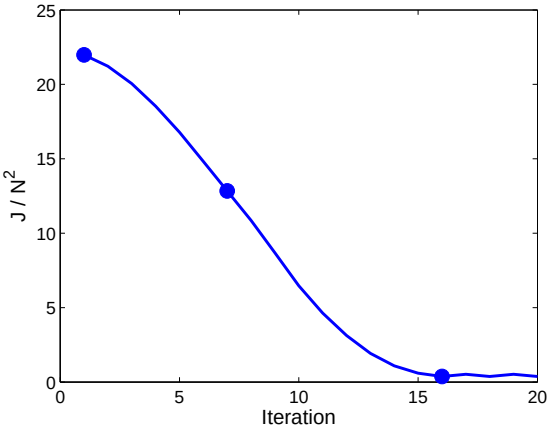
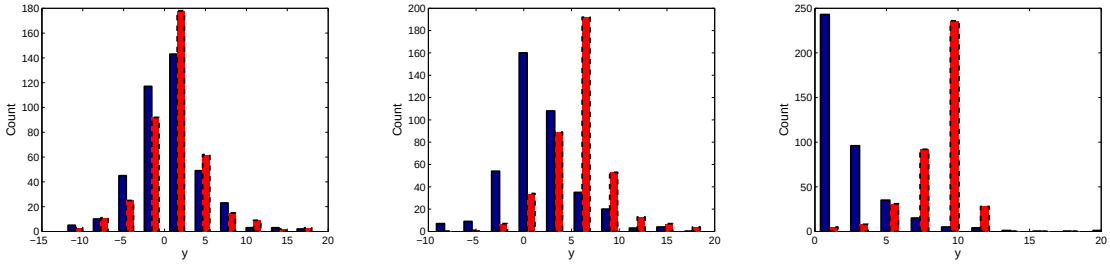


Figure 5.2: The objective function is minimized as we use IPCA to search for the best projection. The circled points correspond to the projections used in Figure 5.3.

Starting with $A_1 \in \mathbb{R}^{1\times 2}$ as a random orthonormal projection matrix, we use IPCA with unity weights to obtain a projection matrix. Figure 5.2 shows the value of the objective function (normalized to a *per pair* value) as a function of gradient descent

iterations. Once the objective function converges, we obtain the projection matrix $A \in \mathbb{R}^{1\times 2}$. This matrix is used to project the data from the 2 original dimensions down to a dimension of 1, such that $y_j = A\boldsymbol{X}_j$.



(a) A_1: Random (b) A_7: Improved (c) A_{16}: Best

Figure 5.3: The evolution of the projection matrix, illustrated on one set from each class. As the objective function is minimized, the statistical separation between sets from differing clusters is increased.

The evolution of the projection matrix is illustrated in Fig. 5.3. One set from each cluster was projected onto the 1-dimensional space defined by A_i (as highlighted in Fig. 5.2). The initial projection matrix A_1, which was randomly generated, offers no distinction between the sets from differing clusters. As the algorithm searches to minimize the objective function, the projection matrix begins to recognize structure within the data, and the sets begin to separate. This process continues until the best projection matrix (in this case A_{16}) is found and the sets are well distinguished. We stress that the distinguishing characteristic is not the Euclidean location of the samples within each data set (as we see they contain some overlap), but the statistics of each set.

5.4.2 Project Honey Pot

We now present the usage of IPCA as a means of exploratory data analysis. The data collection that we utilize in this simulation was provided by Project Honey Pot[1], which is a project designed to learn more about email spammers. Spam is

[1] http://www.projecthoneypot.org

Table 5.1: Data dimensions and corresponding server properties

Dimension	Server Property
1	Total number of emails sent
2	Time elapsed from first email to last email sent
3	BGP server life duration
4	Phishing ratio (number of phishing emails / total emails sent)

typically generated in a two-stage process: first, email addresses are harvested by spammers which troll the web looking for email addresses in html code. Unsolicited messages (i.e. spam) are then sent to these addresses through the use of *spam servers* (e.g. public proxies, unsecured machines, etc), which offer a form of anonymity to the spammer. Hence, it is typically extremely difficult to associate a spam email with the original spammer, as they have taken the necessary steps to prevent detection [68].

Project Honey Pot works by setting up fake websites designed as traps, which have no relevant content when viewed by a standard content browser. However, these 'honey pots' present a unique email address each time the html *source* is viewed and log the IP address which viewed it. By associating each unique email address with an IP address, the project is able to identify the source of the spam email. Whenever an email is received at the phony address, the project is able to connect it to a specific harvester IP address, and extract information about the spam server used by said harvester. By accumulating all of this data, the project hopes to be able to determine some connection between various spammers. Specifically, we are interested in identifying any trends between the spam servers used by phishers, who are criminals seeking to exploit personal information such as passwords and bank account numbers.

This project fits directly into the IPCA framework and it is useful to see if there are any spam server properties which may differentiate types of phishers (i.e. manual vs. automated spammers). Specifically, let $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ be a collection of

harvesters in which each $\boldsymbol{X}_i = [x_1, \ldots, x_{n_i}]$ is a data set consisting of all of the phishing emails[2] received by Project Honey Pot associated with harvester i during the month of October 2006. Each $x_i \in \mathbb{R}^d$ represents an individual email, but the vector values are those corresponding with the properties of spam server from which that email was sent (see Table 5.1). Hence, the values $x \in \boldsymbol{X}$ are typically non-unique, as harvesters send multiple emails over each utilized spam server. We note that within Table 5.1, the 'time elapsed' for dimension 2 is over the entire lifetime of the server while all other properties correspond to over the same one month period. 'BGP duration' [32] is a measure of how long a server is active (i.e. reachable) on the internet. The majority ($\sim 95\%$) of spam servers have a BGP duration of the entire month.

The data dimensions are re-scaled to the 75% quantile to keep differing dimension scales from skewing the IPCA results, which would keep variable selection from being straightforward. We restrict our analysis to harvesters which have sent a phishing email on at least 10 unique spam servers in order to obtain a somewhat smooth PDF estimate using KDE methods.

Given the re-scaled version of $\mathcal{X}$, we apply IPCA with an initialization of a random orthonormal projection $A_1 \in \mathbb{R}^{2\times 4}$, to project the data from 4 dimensions to 2. We utilized the Hellinger distance to approximate the Fisher information distance. The boundedness of the Hellinger distance is extremely useful given the potentially low sample size when estimating PDFs. In Fig. 5.4 we show the value of the cost function (normalized to a *per element pair* value) as IPCA converges on the projection:

$$A = \begin{pmatrix} 0.3318 & 0.9433 & 0.0071 & 0.0057 \\ 0.0359 & -0.0066 & -0.0028 & -0.9993 \end{pmatrix}.$$

[2]We define an email as a phishing email if the subject line contains any words from a dictionary of commonly used phishing terms (e.g. eBay, PayPal, password, account, etc.).

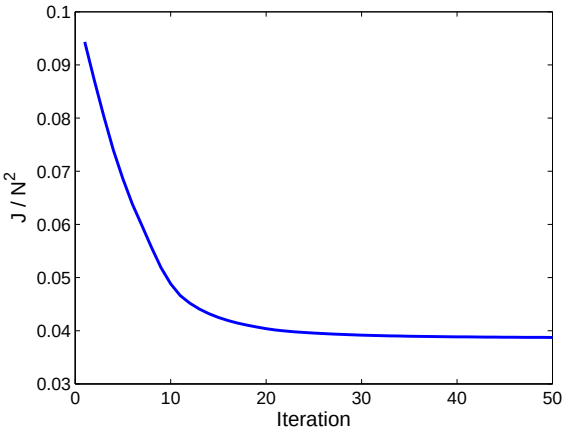
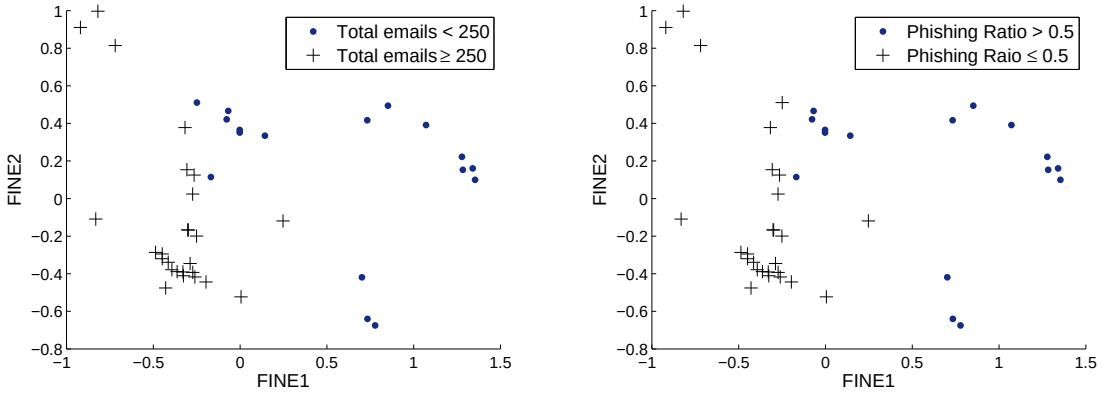


Figure 5.4: The value of the objective function as a function of time, when projecting spam data from 4 to 2 dimensions with IPCA.

We will return to the IPCA projection shortly, but we first embed each harvester into a low-dimensional space using FINE (with cMDS) on the full data. The 2-dimensional embedding in Fig. 5.5 shows a clustering defined by the *harvester* – not server – properties of 'total number of emails sent' and 'phishing ratio'[3]. These clusters are separated by harvesters which, we believe, are essentially manual (i.e. human) and those which are automated (i.e. bots). This conclusion is intuitive as bots send massive amounts of spam (in the thousands), while humans are much more constricted to the time it takes to manually harvest addresses and send spam. Additionally, a manual spammer is predictably more inclined to focus on a specific task (i.e. phishing) while a bot is capable of performing multiple tasks at a high rate, such as both phishing and advertising. These intuitions explain our choice of measures for labeling, and the results – visually defined clusters – confirm this as well.

Returning to the IPCA projection matrix A, we notice that the measures which define the clusters also correspond to the variables of importance found with IPCA. Analysis of the loading vectors shows that the information distance between harvesters is nearly entirely contained in the second and fourth variables, which corre-

[3]Labeling thresholds were defined through a histogram analysis



(a) Labeled by total number of emails sent

(b) Labeled by phishing ratio

Figure 5.5: 2-D FINE embedding of harvesters based on information distance between projected data sets, using different threshold measures for labeling harvesters. These measures form clusters of automated spammers from manual spammers. The labeling measures correspond to properties of each harvester, and are independent of the spam servers used.

spond to the 'Time elapsed from first email to last email sent' and 'Phishing ratio' properties of each spam server. The importance of spam server phishing ratio is intuitive; bots may multi-task as discussed previously, leading to a low phishing ratio on the servers they utilize. Also note that the loadings of A contains some weight with respect to 'total number of emails sent'. As bots send massive amounts of email, it is natural that the spam servers they use will have large values for the total number of emails sent. These results seem to suggest that spammers with manual implementation exploit different servers than those with an automated implementation, as there is not a significant overlap in server usage between clusters.

The indication of 'Time elapsed from first email to last email sent' as an important server property is also of interest. While we do not have the corresponding data available for the harvesters to fully analyze, we suspect that bots send spam in bursts, resulting in shorter server usage periods. This is an intuitive interpretation, but would require additional data to fully validate, and is an area for future work.

5.4.3 LandSAT Imagery

Table 5.2: Distribution of the 4435 training and 2000 test samples in the Landsat data set.

	Soil Type					
	Red	Cotton Crop	Grey	Damp Grey	Vegetable Stubble	Very Damp Grey
Training	1072	479	961	415	470	1038
Test	461	224	397	211	237	470

We now study the performance of IPCA for supervised dimensionality reduction, utilizing the well studied Landsat satellite imagery database [1]. This data set consists of satellite images of 6 differing soil types. Each sample point is a 36-dimensional vector corresponding to the 9 intensity values of a 3×3 pixel region (with overlapping regions) in 4 different spectral bands. The training and test sets (with 4435 and 2000 sample respectively) have been pre-defined with the breakdown described in Table 5.2.

We compare IPCA performance to other methods of linear, supervised dimensionality reduction: linear discriminant analysis (LDA) [42] and quadratic discriminant analysis with slice average variance estimation (QDA-SAVE) [64]. We implement several different classification methods – linear, radial, and quadratic kernel support vector machines (SVMs) [21], and a k-nearest neighbor (k-NN) classifier – as different methods of dimensionality reduction may be optimized specifically for certain classification methods (e.g. LDA and linear classification). In Table 5.3, we illustrate the "best case" classification performance for all simulations, in which the lowest error rate is reported over all projection dimensions with values in the range $m \in [3, 25]$, emphasizing the best performance for each classifier. We see that IPCA outperforms LDA and QDA-SAVE for all classifiers except the quadratic kernel SVM, for which QDA-SAVE narrowly shows better performance.

We further investigate the performance for all classifiers by plotting the classifi-

	Linear	Radial	Quadratic	k-NN
IPCA	**13.60 %**	**9.85 %**	10.05 %	**9.70 %**
LDA	13.70 %	11.35 %	11.25 %	12.60 %
QDA-SAVE	13.65 %	10.15 %	**9.90 %**	10.15 %

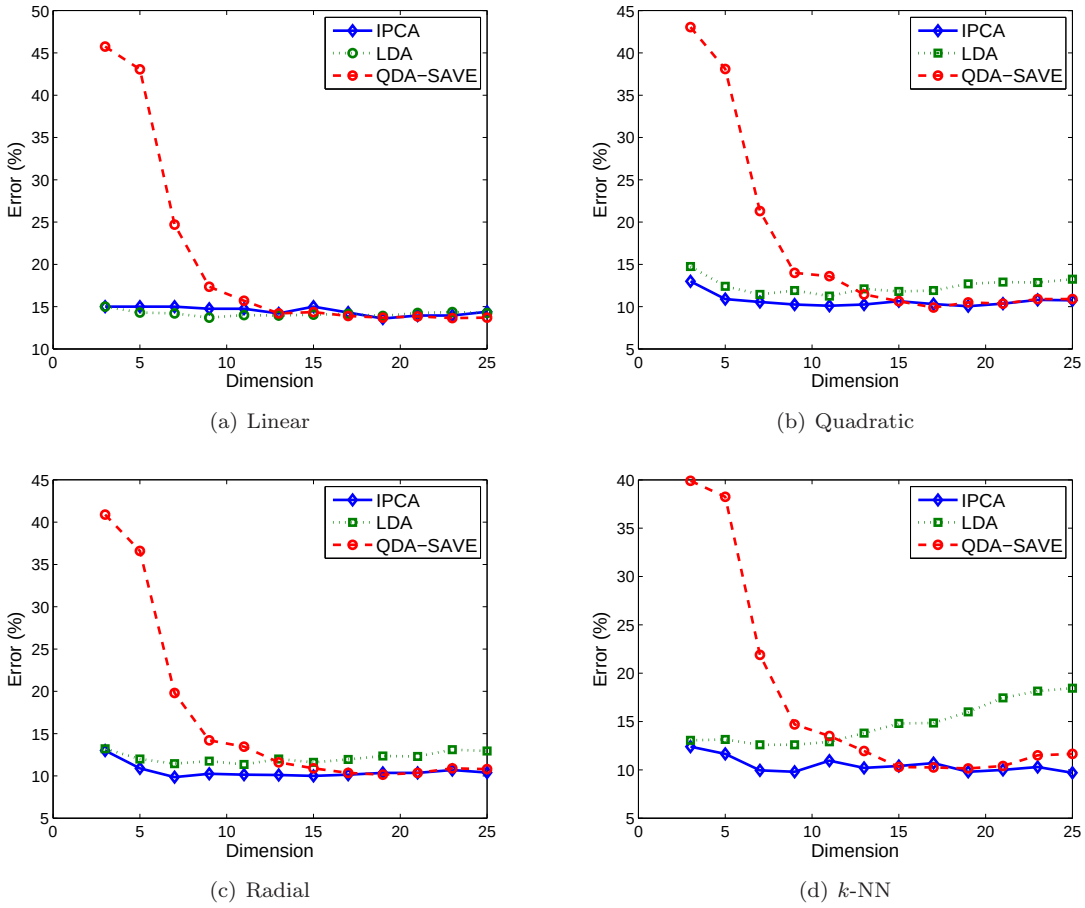
Table 5.3: Classification error probability

cation error as a function of dimension in Fig. 5.6. It is clear that QDA-SAVE has significant difficulties in the low dimensional regime, which may be an issue if significant dimensionality reduction is required (e.g. compression). In contrast, IPCA shows far superior performance in low dimensions, while still maintaining strong competitiveness in high dimensions.

5.5 Conclusions and Future Work

In this chapter, we have offered an information-geometric approach to linear dimensionality reduction through Information Preserving Component Analysis (IPCA). We have shown this method to work in an unsupervised framework by preserving the information distances between high-dimensional data sets in a low-dimensional projection space. Additionally, IPCA may operate as a form of supervised dimensionality reduction by finding the projection space which maximizes the information distances between class PDFs, as this enables control of the probability of classification error.

By finding the low-dimensional space which best preserves/maximizes the dissimilarities between data sets, we have enabled for comparative analysis in how sets from different generative models occupy the projection space. Additionally, analysis of the loading vectors in the IPCA projection matrix allows for a means of variable selection, as the variables which are most crucial to preserving the information distances will have the largest loading values. We have demonstrated this ability with the analysis of spam email, identifying spam server properties which can distinguish



(a) Linear (b) Quadratic

(c) Radial (d) k-NN

Figure 5.6: Classification error probability as a function of dimension when using different classification methods. IPCA show superior performance in nearly all cases, dramatically outperforming QDA-SAVE in the low dimensional regime.

manual spammers from automated spammers. We have additionally demonstrated supervised IPCA towards the classification task using satellite imagery, resulting in superior performance to standard approaches to linear, supervised dimensionality reduction. In the proceeding chapter, we will apply IPCA towards flow cytometric analysis, which was the motivating application for the algorithm.

In future work, we would like to further evaluate the Project Honey Pot database to observe other trends and continue to identify properties of harvester communities. Additionally, we plan to utilize different methods for optimizing the IPCA cost function, as there may exist more efficient methods than gradient descent (e.g. fixed

point iteration). Finally, we would like to continue pursuing applications of interest, specifically in the biomedical fields, and utilize IPCA for exploratory research.

5-A Appendix: Orthonormality Constraint on Gradient Descent

We derive the orthonormality constraint for our gradient descent optimization in the following manner; solving

$$A = \arg \min_{A:AA^T=I} J(A),$$

where I is the identity matrix. Using Lagrangian multiplier M, this is equivalent to solving

$$A = \arg \min_{A} \tilde{J}(A),$$

where $\tilde{J}(A) = J(A) + \mathrm{tr}(A^T M A)$. We can iterate the projection matrix A, using gradient descent, as:

$$A_{i+1} = A_i - \mu \frac{\partial}{\partial A} \tilde{J}(A_i), \tag{5.13}$$

where $\frac{\partial}{\partial A}\tilde{J}(A) = \frac{\partial}{\partial A}J(A) + (M + M^T)A$ is the gradient of the cost function w.r.t. matrix A. To ease notation, let $\Delta \triangleq \frac{\partial}{\partial A}J(A_i)$ and $\tilde{\Delta} \triangleq \frac{\partial}{\partial A}\tilde{J}(A_i)$. Continuing with the constraint $A_{i+1}A_{i+1}^T = I$, we right-multiply (5.13) by A_{i+1}^T and obtain

$$0 = -\mu A_i \tilde{\Delta}^T - \mu \tilde{\Delta} A_i^T + \mu^2 \tilde{\Delta}\tilde{\Delta}^T,$$

$$\mu \tilde{\Delta}\tilde{\Delta}^T = \tilde{\Delta}A^T + A\tilde{\Delta}^T, \tag{5.14}$$

$$\mu(\Delta + (M+M^T)A)(\Delta + (M+M^T)A)^T = (\Delta A(M+M^T)A)A^T + A(\Delta A^T(M+M^T)A).$$

Let $Q = M + M^T$, hence $\tilde{\Delta} = \Delta + QA$. Substituting this into (5.14) we obtain:

$$\mu(\Delta\Delta^T + QA\Delta^T + \Delta A^T Q + QQ^T) = \Delta A^T + A\Delta^T + 2Q.$$

Next we use the Taylor series expansion of Q around $\mu = 0$: $Q = \sum_{j=0}^{\infty} \mu^j Q_j$. By equating corresponding powers of μ (i.e. $\frac{\partial^j}{\partial \mu^j}|_{\mu=0} = 0$), we identify:

$$Q_0 = -\frac{1}{2}(\Delta A^T + A\Delta^T),$$

$$Q_1 = \frac{1}{2}(\Delta + Q_0 A)(\Delta + Q_0 A)^T.$$

Replacing the expansion of Q in $\tilde{\Delta} = \Delta + QA$:

$$\tilde{\Delta} = \Delta - \frac{1}{2}(\Delta A^T + A\Delta^T)A + \mu\, Q_1 A + \mu^2\, Q_2 A + \ldots.$$

Finally, we would like to assure a sufficiently small step size to control the error in forcing the constraint due to a finite Taylor series approximation of Q. Using the L_2 norm of $\tilde{\Delta}$ allows us to calculate an upper bound on the Taylor series expansion:

$$\|\tilde{\Delta}\| \leqslant \|\Delta - \frac{1}{2}(\Delta A^T + A\Delta^T)A\| + \mu\, \|Q_1 A\| + \mu^2\, \|Q_2 A\| + \ldots.$$

We condition the norm of the first order term in the Taylor series approximation to be significantly smaller than the norm of the zeroth order term. If $\mu \ll \|\Delta - \frac{1}{2}(\Delta A^T + A\Delta^T)A\|/\|Q_1 A\|$ then:

$$\frac{\partial}{\partial A}\tilde{J}(A) = \frac{\partial}{\partial A}J(A) + Q_0 A + \mu Q_1 A, \tag{5.15}$$

where

$$Q_0 = -\frac{1}{2}\left(\left(\frac{\partial}{\partial A}J(A)\right)A^T + A\left(\frac{\partial}{\partial A}J(A)\right)^T\right)$$

$$Q_1 = \frac{1}{2}\left(\frac{\partial}{\partial A}J(A) + Q_0 A\right)\left(\frac{\partial}{\partial A}J(A) + Q_0 A\right)^T,$$

is a good approximation of the gradient constrained to $AA^T = I$. We omit the higher order terms as we experimentally find that they are unnecessary, especially as even $\mu^2 \to 0$. We note that while there are other methods for forcing the gradient to obey orthogonality [31, 33], we find our method is straightforward and sufficient for our purposes.

5-B Appendix: Proof of strictly non-increasing property of Hellinger distance w.r.t. an orthonormal data projection

Here we prove that the Hellinger distance between the PDFs of x and x' is greater or equal to the Hellinger distance between the PDFs of $y = Ax$ and $y' = Ax'$, respectively, where A satisfies $AA^T = I$.

Theorem 5.2. *Let RVs $X, X' \in \mathbb{R}^n$ have PDFs f_X and $f_{X'}$, respectively. Using the $d \times n$ matrix A satisfying $AA^T = I_d$, construct RVs $Y, Y' \in \mathbb{R}^d$ such that $Y = AX$ and $Y' = AX'$. The following relation holds:*

$$D_H(f_X, f_{X'}) \geqslant D_H(f_Y, f_{Y'}), \tag{5.16}$$

where f_Y and $f_{Y'}$ are the PDFs of Y, Y', respectively.

The proof is in two parts. First, we show that the Hellinger distance is constant over an arbitrary dimension preserving orthogonal transformation. Next, we show that the same truncation of two random vectors does not increase the Hellinger distance.

Let M be an $n \times n$ orthonormal matrix, i.e., $MM^T = I_n$ and $M^TM = I_n$. Define the random vectors $V, V' \in \mathbb{R}^n$ as follows $V = MX$ and $V' = MX$. By a change of variables, we have

$$f_V \;:\; f_V(v) = f_X(M^Tv) \tag{5.17}$$

$$f_{V'} \;:\; f_{V'}(v') = f_{X'}(M^Tv'). \tag{5.18}$$

Note that the Jacobian of the transformation is 1 and M^T is the inverse of the transformation both due to the orthonormality of M. The squared Hellinger distance between V and V' is given by

$$D_H^2(f_V, f_{V'}) = \int \left(\sqrt{f_V(v)} - \sqrt{f_{V'}(v)}\right)^2 dv. \tag{5.19}$$

Substituting the PDFs from (5.17) into (5.19), we have

$$D_H^2(f_V, f_{V'}) = \int \left(\sqrt{f_X(M^T v)} - \sqrt{f_{X'}(M^T v)}\right)^2 dv. \tag{5.20}$$

Next, using the orthonormality of M we replace $x = M^T v$ and $dx = dv$ in (5.32) and obtain

$$D_H^2(f_V, f_{V'}) = \int \left(\sqrt{f_X(x)} - \sqrt{f_{X'}(x)}\right)^2 dv = D_H^2(f_X, f_{X'}) \tag{5.21}$$

and the squared Hellinger distance remains the same.

We proceed with the second part of the proof. Consider the random vector V as a concatenation of RVs Y and Z: $V^T = [Y^T Z^T]$. If we write matrix M as $M^T = [A^T, B^T]$ where A is $d \times n$ and B is $(n-d) \times n$, then $Y = AX$, $Y' = AX'$, and A satisfies $AA^T = I_d$ (but not $A^T A = I_n$). Since $V^T = [Y^T Z^T]$, we have $D_H^2(f_V, f_{V'}) = D_H^2(f_{YZ}, f_{Y'Z'})$ and by virtue of (5.21)

$$D_H^2(f_X, f_{X'}) = D_H^2(f_{YZ}, f_{Y'Z'}). \tag{5.22}$$

Next, we use the following lemma:

Lemma 5.3. *Let $Y, Y' \in \mathbb{R}^d$ and $Z, Z' \in \mathbb{R}^{n-d}$ be RVs and denote: the joint PDF of Y and Z by f_{YZ}, the joint PDF of Y' and Z' by $f_{Y'Z'}$, the marginal PDF of Y by f_Y, the marginal PDF of Y' by $f_{Y'}$, the conditional PDF of Z by $f_{Z|Y}$, and the conditional PDF of Z' by $f_{Z'|Y'}$. The following holds:*

$$D_H^2(f_{YZ}, f_{Y'Z'}) - D_H^2(f_Y, f_{Y'}) \geqslant 0. \tag{5.23}$$

The proof this Lemma begins as follows:

$$
\begin{aligned}
D_H^2(f(y,z),g(y,z)) - D_H^2(f(y),g(y)) = \\
\int\int \left(\sqrt{f(y,z)} - \sqrt{g(y,z)}\right)^2 dydz - \int \left(\sqrt{f(y)} - \sqrt{g(y)}\right)^2 dy = \\
\int\int f(y,z) + g(y,z) - 2\sqrt{f(y,z)g(y,z)}dydz - \int f(y) + g(y) - 2\sqrt{f(y)g(y)}dy = \\
-2\int\int \sqrt{f(y,z)g(y,z)}dydz + 2\int \sqrt{f(y)g(y)}dy = \\
2\left[\int \sqrt{f(y)g(y)}dy - \int\int \sqrt{f(y,z)g(y,z)}dydz\right].
\end{aligned}
$$

We may now continue by showing $\int\int \sqrt{f(y,z)g(y,z)}\,dy\,dz \leqslant \int \sqrt{f(y)g(y)}\,dy$:

$$
\begin{aligned}
\int\int \sqrt{f(y,z)g(y,z)}\,dy\,dz &= \int\int \sqrt{f(y)f(z|y)g(y)g(z|y)}\,dy\,dz \qquad (5.24)\\
&= \int\int \sqrt{f(y)g(y)}\sqrt{f(z|y)g(z|y)}\,dy\,dz \\
&= \int \sqrt{f(y)g(y)}\left(\int \sqrt{f(z|y)g(z|y)}dz\right)dy \\
&\leqslant \int \sqrt{f(y)g(y)}\left(\int \sqrt{f(z|y)}^2 dz\right)^{\frac{1}{2}}\left(\int \sqrt{g(z|y)}^2 dz\right)^{\frac{1}{2}} dy \qquad (5.25)\\
&= \int \sqrt{f(y)g(y)}\left(\int f(z|y)dz\right)^{\frac{1}{2}}\left(\int g(z|y)dz\right)^{\frac{1}{2}} dy \\
&= \int \sqrt{f(y)g(y)}\,(1)^{\frac{1}{2}}\,(1)^{\frac{1}{2}}\,dy \\
&= \int \sqrt{f(y)g(y)}\,dy.
\end{aligned}
$$

Note that (5.24) used Bayes rule and (5.25) used the Cauchy-Schwartz inequality. We now immediately obtain the following corollary:

Corollary 5.4. *Let $Y, Y' \in \mathbb{R}^d$ and $Z, Z' \in \mathbb{R}^{n-d}$ be RVs and denote: the joint PDF of Y and Z by f_{YZ}, the joint PDF of Y' and Z' by $f_{Y'Z'}$, the marginal PDF of Y by f_Y, and the marginal PDF of Y' by $f_{Y'}$. The following holds:*

$$D_H^2(f_{YZ}, f_{Y'Z'}) \geqslant D_H^2(f_Y, f_{Y'}). \qquad (5.26)$$

This corollary suggests that the squared Hellinger distance must not increase as a result of marginalization. Without loss of generality, due to the monotonic behavior of the square root function, the same may be said for the strict Hellinger distance, yielding the desired result

$$D_H(f_X, f_{X'}) \geqslant D_H(f_Y, f_{Y'}). \tag{5.27}$$

5-C Appendix: Proof of strictly non-increasing property of KL divergence w.r.t. an orthonormal data projection

Here we prove that the KL divergence between the PDFs of x and x' is greater or equal to the KL divergence between the PDFs of $y = Ax$ and $y' = Ax'$, respectively, where A satisfies $AA^T = I$. Note that much of this derivation is repetitive to that of the Hellinger distance in Appendix 5-B, but we include all steps here for completeness.

Theorem 5.5. *Let RVs $X, X' \in \mathbb{R}^n$ have PDFs f_X and $f_{X'}$, respectively. Using the $d \times n$ matrix A satisfying $AA^T = I_d$, construct RVs $Y, Y' \in \mathbb{R}^d$ such that $Y = AX$ and $Y' = AX'$. The following relation holds:*

$$KL(f_X \| f_{X'}) \geqslant KL(f_Y \| f_{Y'}), \tag{5.28}$$

where f_Y and $f_{Y'}$ are the PDFs of Y, Y', respectively.

The proof is in two parts. First, we show that the KL divergence is constant over an arbitrary dimension preserving orthogonal transformation. Next, we show that the same truncation of two random vectors does not increase KL.

Let M be an $n \times n$ orthonormal matrix, i.e., $MM^T = I_n$ and $M^T M = I_n$. Define the random vectors $V, V' \in \mathbb{R}^n$ as follows $V = MX$ and $V' = MX$. By a change of

variables, we have

$$f_V \;:\; f_V(v) = f_X(M^T v) \tag{5.29}$$

$$f_{V'} \;:\; f_{V'}(v') = f_{X'}(M^T v'). \tag{5.30}$$

Note that the Jacobian of the transformation is 1 and M^T is the inverse of the transformation both due to the orthonormality of M. The KL divergence between V and V' is given by

$$KL(f_V \| f_{V'}) = \int f_V(v) \log \frac{f_V(v)}{f_{V'}(v)} dv. \tag{5.31}$$

Substituting the PDFs from (5.29) into (5.31), we have

$$KL(f_V \| f_{V'}) = \int f_X(M^T v) \log \frac{f_X(M^T v)}{f_{X'}(M^T v)} dv. \tag{5.32}$$

Next, using the orthonormality of M we replace $x = M^T v$ and $dx = dv$ in (5.32) and obtain

$$KL(f_V \| f_{V'}) = \int f_X(x) \log \frac{f_X(x)}{f_{X'}(x)} dx = KL(f_X \| f_{X'}) \tag{5.33}$$

and the KL divergence remains the same.

We proceed with the second part of the proof. Consider the random vector V as a concatenation of RVs Y and Z: $V^T = [Y^T Z^T]$. If we write matrix M as $M^T = [A^T, B^T]$ where A is $d \times n$ and B is $(n-d) \times n$, then $Y = AX$, $Y' = AX'$, and A satisfies $AA^T = I_d$ (but not $A^T A = I_n$). Since $V^T = [Y^T Z^T]$, we have $KL(f_V \| f_{V'}) = KL(f_{YZ} \| f_{Y'Z'})$ and by virtue of (5.33)

$$KL(f_X \| f_{X'}) = KL(f_{YZ} \| f_{Y'Z'}). \tag{5.34}$$

Next, we use the following lemma:

Lemma 5.6. *Let $Y, Y' \in \mathbb{R}^d$ and $Z, Z' \in \mathbb{R}^{n-d}$ be RVs and denote: the joint PDF of Y and Z by f_{YZ}, the joint PDF of Y' and Z' by $f_{Y'Z'}$, the marginal PDF of Y by f_Y, the marginal PDF of Y' by $f_{Y'}$, the conditional PDF of Z by $f_{Z|Y}$, and the conditional PDF of Z' by $f_{Z'|Y'}$. The following holds:*

$$\int f(y) KL(f_{Z|Y} \| f_{Z'|Y'}) dy = KL(f_{YZ} \| f_{Y'Z'}) - KL(f_Y \| f_{Y'}). \tag{5.35}$$

This may be proven as follows:

$$\begin{aligned}
\int f(y) KL(f(z|y) \| g(z|y)) dy = \\
\int f(y) \int f(z|y) \log \frac{f(z|y)}{g(z|y)} dz dy = \\
\iint f(y,z) \log \frac{f(y,z)/f(y)}{g(y,z)/g(y)} dz dy = \\
\iint f(y,z) \left(\log \frac{f(y,z)}{g(y,z)} - \log \frac{f(y)}{g(y)} \right) dz dy = \\
\iint f(y,z) \log \frac{f(y,z)}{g(y,z)} dz dy - \iint f(y,z) \log \frac{f(y)}{g(y)} dz dy = \\
KL(f(y,z) \| g(y,z)) - \iint f(y,z) dz \log \frac{f(y)}{g(y)} dy = \\
KL(f(y,z) \| g(y,z)) - \int f(y) \log \frac{f(y)}{g(y)} dy = \\
KL(f(y,z) \| g(y,z)) - KL(f(y) \| g(y)).
\end{aligned} \tag{5.36}$$

Identifying that the LHS of (5.35) is non-negative, we immediately obtain the following corollary:

Corollary 5.7. *Let $Y, Y' \in \mathbb{R}^d$ and $Z, Z' \in \mathbb{R}^{n-d}$ be RVs and denote: the joint PDF of Y and Z by f_{YZ}, the joint PDF of Y' and Z' by $f_{Y'Z'}$, the marginal PDF of Y by f_Y, and the marginal PDF of Y' by $f_{Y'}$. The following holds:*

$$KL(f_{YZ} \| f_{Y'Z'}) \geqslant KL(f_Y \| f_{Y'}). \tag{5.37}$$

This corollary suggests that KL must not increase as a result of marginalization.

Application of (5.37) from Corollary 5.7 to (5.34), yields the desired result

$$KL(f_X \| f_{X'}) \geqslant KL(f_Y \| f_{Y'}). \tag{5.38}$$

CHAPTER VI

Application to Flow Cytometry

6.1 Motivation

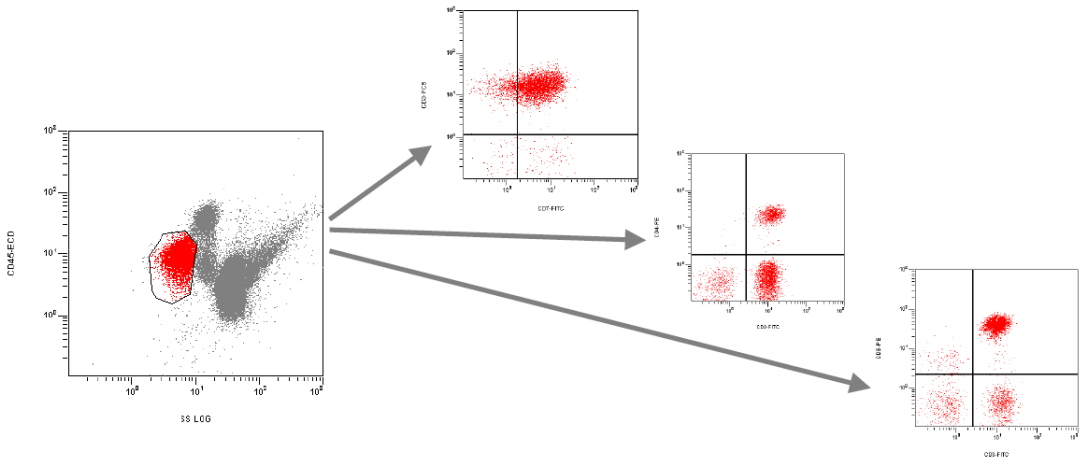


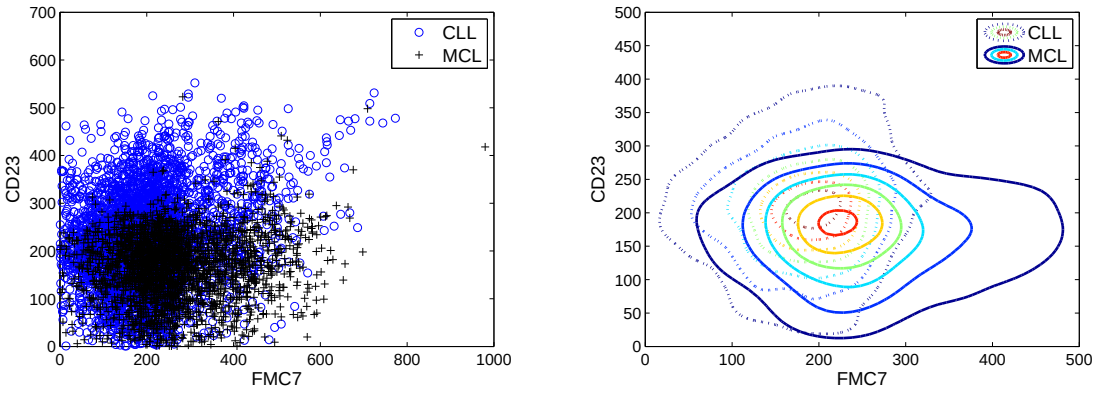
Figure 6.1: Historically, the process of clinical flow cytometry analysis relies on a series of 2-dimensional scatter plots in which cell populations are selected for further evaluation. This process does not take advantage of the multidimensional nature of the problem.

In clinical flow cytometry, cellular suspensions are prepared from patient samples (blood, bone marrow, and solid tissue), and evaluated simultaneously for the presence of several expressed surface antigens and for characteristic patterns of light scatter as the cells pass through an interrogating laser. Antibodies to each target antigen are conjugated to fluorescent markers, and each individual cell is evaluated via detection of the fluorescent signal from each marker. The data from clinical flow cytometry can be considered multidimensional both from the standpoint of multiple characteristics

measured for each cell, and from the standpoint of thousands of cells analyzed per sample. Nonetheless, clinical pathologists generally interpret clinical flow cytometry results in the form of two-dimensional scatter plots in which the axes each represent one of multiple cell characteristics analyzed (Fig. 6.1). By viewing a series of these histograms, a clinician is able to determine a diagnosis for the patient through clinical experience of the manner in which certain leukemias and lymphomas express certain markers.

Given that the standard method of cytometric analysis involves projections onto the axes of the data, the multidimensional nature of the data is not fully exploited. As such, typical flow cytometric analysis is comparable to hierarchical clustering methods, in which data is segmented on an axis-by-axis basis. Marker combinations have been determined through years of clinical experience, leading to relative confidence in analysis given certain axes projections. These projection methods, however, contain the underlying assumption that marker combinations are independent of each other, and do not utilize the dependencies which may exist within the data. Ideally, clinicians would like to analyze the full-dimensional data, but this cannot be visualized outside of 3 dimensions.

An example of the difficulty in analysis of 2-dimensional scatter plots (essentially plots of the marginal PDFs) is illustrated in Fig. 6.2. Two distinct but immunophenotypically similar forms of lymphoid leukemia – mantle cell lymphoma (MCL) and chronic lymphocytic leukemia (CLL) – are illustrated with both scatter and contour plots. These diseases exhibit similar expression patterns to many surface antigens, but are generally distinct with respect to antigens CD23 and FMC7. The significant similarity and overlapping nature in the marginal plots illustrates the difficulty in traditional 2-dimensional flow cytometry analysis. It would be potentially beneficial,



(a) Scatter Plot

(b) Contour Plot

Figure 6.2: 2-dimensional plots of disease classes CLL and MCL, in which each point represents a unique blood cell. The overlapping nature of the scatter plots makes it difficult for pathologists to differentiate disease classes using primitive 2-dimensional axes projections.

therefore, to develop systems for clustering and visualization of clinical flow cytometry data that utilize all dimensions of data derived for each cell during routine clinical analysis.

There have been previous attempts at using machine learning to aid in flow cytometry diagnosis. Some have focused on clustering in the high-dimensional space [78,79], while others have utilized information geometry to identify differences in sample subsets and between data sets [69, 70]. These methods have not satisfied the problem because they do not significantly approach the aspect of visualization for 'human in the loop' diagnosis, and the ones that do [60,61] only apply dimensionality reduction to a single set at a time.

In this chapter we utilize the methods and techniques described thus far towards the problems of diagnosis, verification, and visualization in flow cytometric analysis. While the expression of various markers may be highly variable over different patients, the general characterization of the multivariate PDF underlying each patient sample is much less variable. Hence, each distribution exists on some statistical

manifold with a much lower dimensional parameterization, and this application is appropriate for FINE. By embedding each patient data set into a single low-dimensional Euclidean space, we enable pathologists to visually identify relative similarities between multiple patients of differing disease diagnoses. This provides a simple and efficient means of determining which data sets may need further investigation (e.g. for possible misdiagnosis). This work has produced promising results which we have recently published [15, 16, 34].

We have determined that unsupervised IPCA would be an ideal form of dimensionality reduction for flow cytometric visualization of the data domain. This is expected as the IPCA algorithm was originally motivated by the cytometry problem. The specific properties of IPCA which are beneficial toward cytometric analysis are listed as such:

- Orthonormal: The data needs to be preserved without scaling or skewing, as this is most similar to the current methods in practice (i.e. axes projections).
- Unsupervised: This requirement is straightforward as the dimensionality reduction would be an aid for diagnosis, so no labels would be available. Learning should be based entirely on the geometry of the data.
- Linear: Once the projection is determined, the subspace is constant and does not need to be recomputed when adding new data. New data is easily projected and analyzed as desired.
- Relationship Preserving: Patients in the same disease class should show similar expressions in the low-dimensional space, while differing disease classes should be visually distinct from one another.

IPCA provides a low-dimensional representation which is a linear combination of

the various markers, enabling clinicians to visualize all of the data simultaneously, rather than the current process of axes projections, which only relays information in relation to two markers at a time. An important facet of the IPCA projection matrix is its variable selection, which relays information describing which marker combinations yield the most information. This has the significant benefit of allowing clinicians and researchers to experiment with new marker combinations, and obtain a measure of their diagnostic ability in certain disease classes.

In the rest of this chapter we present several case studies for flow cytometric analysis. Section 6.2 presents a lymphoid leukemia study involving CLL and MCL. We isolate the CLL disease for further interrogation in Section 6.3, focusing on prognosis. In Section 6.4 we focus on the disease classes of acute lymphoblastic leukemia and hematogone hyperplasia. Finally, we discuss areas for future work in the area of cytometric analysis in Section 6.6.

All work presented in this chapter was done in collaboration with the Department of Pathology at the University of Michigan, which provided data and diagnoses for all patients. Please note that in all studies the matrix of Fisher information distances $D(\mathcal{X})$ was approximated with the symmetric Kullback-Leibler divergence using densities estimated with KDE methods (see Appendix A). We have obtained similar results when using the Hellinger distance as well.

6.2 Lymphoid Leukemia Study

For our first study, we will compare patients with two distinct but immunophenotypically similar forms of lymphoid leukemia – mantle cell lymphoma (MCL) and chronic lymphocytic leukemia (CLL), as illustrated in Fig. 6.2. These diseases display similar characteristics with respect to many expressed surface antigens, but are

generally distinct in their patterns of expression of two common B lymphocyte antigens: CD23 and FMC7 (a distinct conformational epitope of the CD20 antigen). Typically, CLL is positive for expression of CD23 and negative for expression of FMC7, while MCL is positive for expression of FMC7 and negative for expression of CD23. These distinctions should lead to a difference in densities between patients in each disease class.

6.2.1 The Data

Dimension	Marker
1	Forward Light Scatter
2	Side Light Scatter
3	FMC7
4	CD23
5	CD45
6	Empty

Table 6.1: Data dimensions and corresponding markers for analysis of CLL and MCL.

The data set $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{43}\}$ consists of 43 patients, 23 of which have been diagnosed with CLL and 20 diagnosed with MCL. Each $\boldsymbol{X}_i$ is a 6-dimensional matrix corresponding to the flow cytometer output of the i^{th} patient; each dimension corresponding to a different marker (see Table 6.1), and each element representing a unique blood cell, totaling $n_i \sim 5000$ total cells per patient.

Utilizing the MLE method on the matrix of Fisher information distances (geodesically approximated with the symmetric KL-divergence), we estimate (with smoothing) the local intrinsic dimension of each patient PDF. The results are shown in Fig. 6.3, where we can see the intrinsic dimension is $m = \{2, 3\}$. This result can be interpreted as recognizing the 2 specific markers which most significantly differentiate between classes (i.e. $m = 2$), but also accounting for the fact that there still exists subtle differences between members of the same class, and some patients may not exhibit the expected response to specific antigens as strongly as others (i.e. $m = 3$).

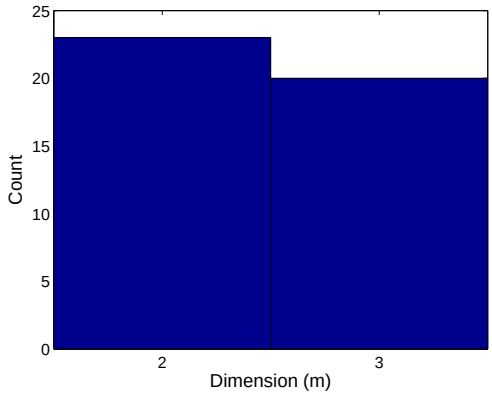


Figure 6.3: Histogram of local dimension estimates for the statistical manifold defined by flow cytometry results of the lymphoid leukemia study.

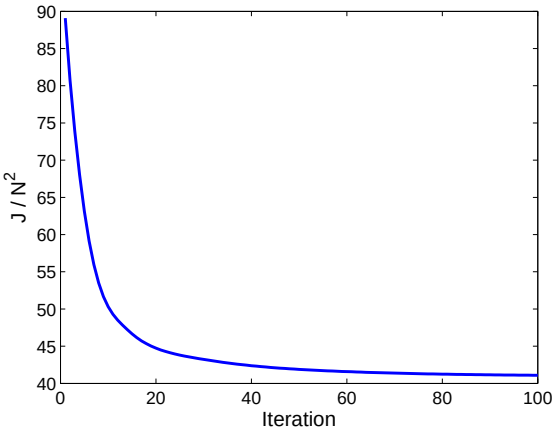


Figure 6.4: CLL and MCL Study: Evaluating the IPCA objective as a function of time. As the iterations increase, the objective function eventually converges.

Hence, we may appropriately embed the statistical manifold in 2 or 3 dimensions. Note that the k-NN algorithm yielded identical results.

6.2.2 IPCA

We found the IPCA projection as

$$A = \begin{pmatrix} -0.1177 & 0.0693 & 0.8979 & 0.2513 & 0.3346 & -0.0032 \\ 0.0077 & -0.2678 & -0.1541 & 0.9243 & -0.2224 & 0.0270 \end{pmatrix}. \tag{6.1}$$

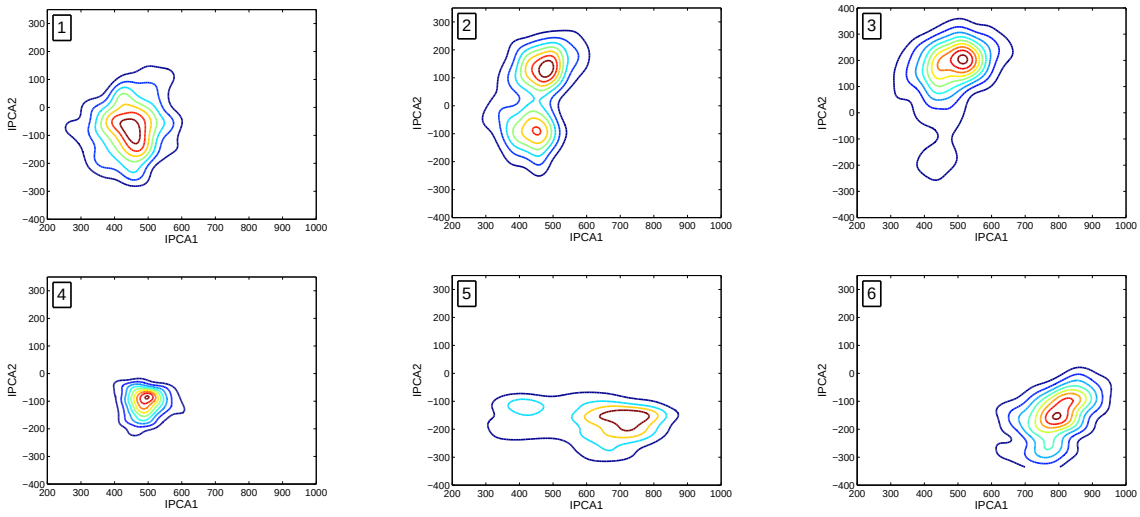
This projection was calculated by minimizing the objective function with respect to A, as illustrated in Fig. 6.4 in which the squared error (per element pair) is plotted

as a function of time. As the iteration i increases, J converges and A_i is determined to be the IPCA projection matrix. We note that while dimension 6 corresponds to no marker (it is a channel of just noise), we do not remove the channel from the data sets, as the projection determines this automatically (i.e. loading values approach 0). Additionally, due to computational complexity issues, each data set was randomly subsampled such that $n_i = 500$. While we would not necessarily suggest this decimation in practice, we have found it to have a minimal effect during experimentation.

Given the IPCA projection, we illustrate the 2-dimensional PDFs of several different patients in the projected space in Fig. 6.5. We selected patients based on the symmetric KL-divergence values between patients of different disease class. Specifically, we selected the CLL and MCL patients with a small divergence (i.e. most similar PDFs), patients with a large divergence (i.e. least similar PDFs), and patients which represented the centroid of each disease class. These low-dimensional PDFs, which are what would be utilized by a diagnostician, are visibly different between disease classes. While the most similar CLL and MCL patients do share much similarity in their IPCA PDFs, there is still a significant enough difference to distinguish them, especially given the similarities to other patient PDFs.

Using the IPCA projection matrix (6.1) for variable selection, the loading vectors are highly concentrated towards the 3^{rd} and 4^{th} dimensions, which correspond to fluorescent markers FMC7 and CD23. We acknowledge that this marker combination is well known and currently utilized in the clinical pathology community for differentiating CLL and MCL[1]. We stress, however, that what had previously been determined through years of clinical experience was able to be independently

[1]CD45 and light scatter characteristics are often used as gating parameters for selection of lymphocytes among other cell types prior to analysis, but CD23 and FMC7 are the main analytical biomarkers in this 3-color assay.



(a) Most Similar (b) Centroids (c) Least Similar

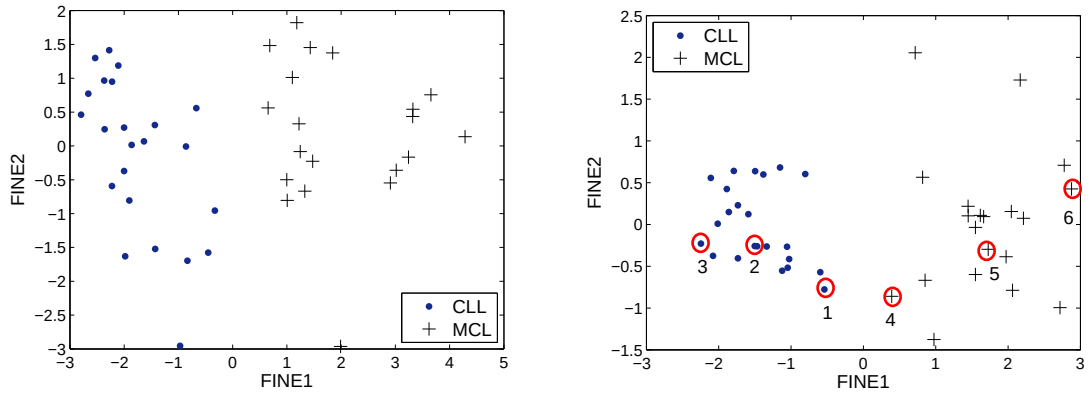
Figure 6.5: CLL and MCL Study: Contour plots (i.e. PDFs) of the IPCA projected data. The top row corresponds to the PDFs of the CLL patients, while the bottom row represents PDFs of MCL patients. The selected patients are those most similar between disease classes, the centroids of disease classes, and those least similar between disease classes, as highlighted in Fig. 6.6(b).

validated quickly using IPCA.

6.2.3 FINE

We now illustrate the 2-dimensional embedding obtained with FINE (using cMDS) of the projected data. The embedding results are shown in Fig. 6.6(a), in which each point in the plot represents an individual patient. It should be noted that there exists a natural separation between the disease types, as the implementation was entirely unsupervised. As such, we can conclude that there is a natural difference in probability distribution between the disease classes as well. Although this is known through years of clinical experience, we were able to determine this without any a priori knowledge; simply through information geometry.

The separation between classes is preserved in Fig. 6.6(b) when using the IPCA projected data as compared to using the full-dimensional data. An important byproduct of this natural clustering is the ability to visualize the cytometric data in a man-



(a) Full Dimension (b) IPCA Projection

Figure 6.6: CLL and MCL Study: Comparison of embeddings, obtained with FINE, using the full dimensional data and the data projected with IPCA. IPCA preserves the separation between disease classes. The circled points correspond to the density plots in Fig. 6.5, numbered respectively.

ner which allows comparisons between patients. The circled points correspond to the PDFs shown in Fig. 6.5; in the IPCA projection space, these patients are well distinguished between disease classes, but it is still difficult to instantaneously compare several patients. In the space defined by FINE, the patients are easily compared and differentiated within the clusters of each disease type.

6.3 Chronic Lymphocytic Leukemia Study

Continuing our study of patients with chronic lymphocytic leukemia (CLL), we wish to determine subclasses within the CLL disease class. Specifically, we now use IPCA to find a low-dimensional space which preserves the differentiation between patients with good and poor prognoses (i.e. favorable and unfavorable immunophenotypes). Literature [29] has shown that patients whose leukemic cells are strong expressors of CD38 have significantly worse survival outcome. Genotypic studies have shown that the absence of somatic mutation within immunoglobulin genes of CLL cells (a so-called "pre-follicular" genotype) is a potent predictor of worse outcome. High levels of CD38 expression are an effective surrogate marker for the absence of

somatic immunoglobulin gene mutation, and also have been shown to be an independent predictor of outcome in some studies. Since patients can generally be stratified by CD38 expression levels, and CD38 has been shown to emerge as a defining variable of CLL subsets in hierarchical immunophenotypic clustering [40], we would expect IPCA to localize the CD38 variable as one of importance when analyzing CLL data.

6.3.1 The Data

Dimension	Marker
1	Forward Light Scatter
2	Side Light Scatter
3	CD5
4	CD38
5	CD45
6	CD19

Table 6.2: Data dimensions and corresponding markers for analysis of CLL.

Using the same patients (those diagnosed with CLL) as in the lymphoid leukemia study, we define $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{23}\}$, where each $\boldsymbol{X}_i$ was analyzed with by the series of markers in Table 6.2. Local dimension estimation suggests an intrinsic dimension of $m = 3$ for all samples, which will be further discussed shortly.

6.3.2 IPCA

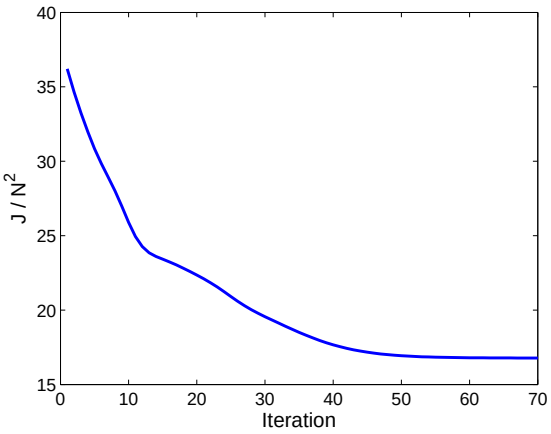
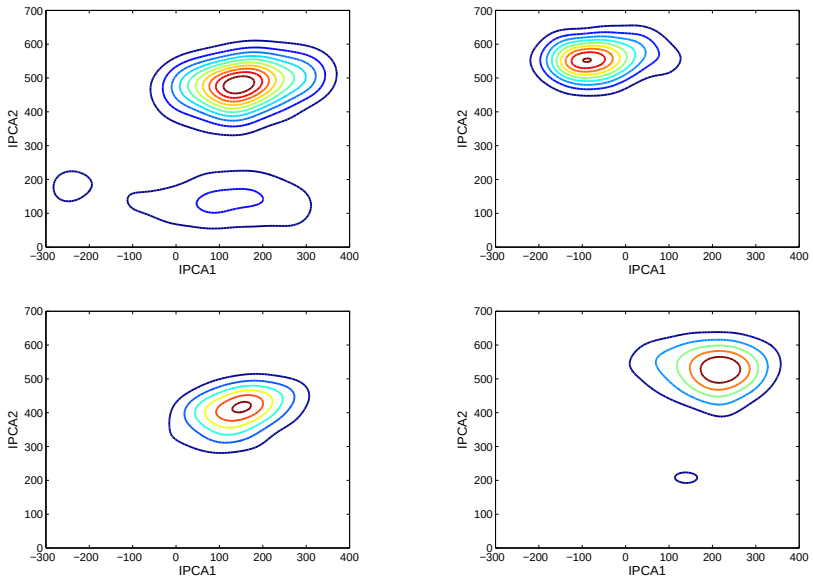


Figure 6.7: CLL Prognosis Study: The value of the IPCA objective function v.s. time.



(a) Most Similar (b) Least Similar

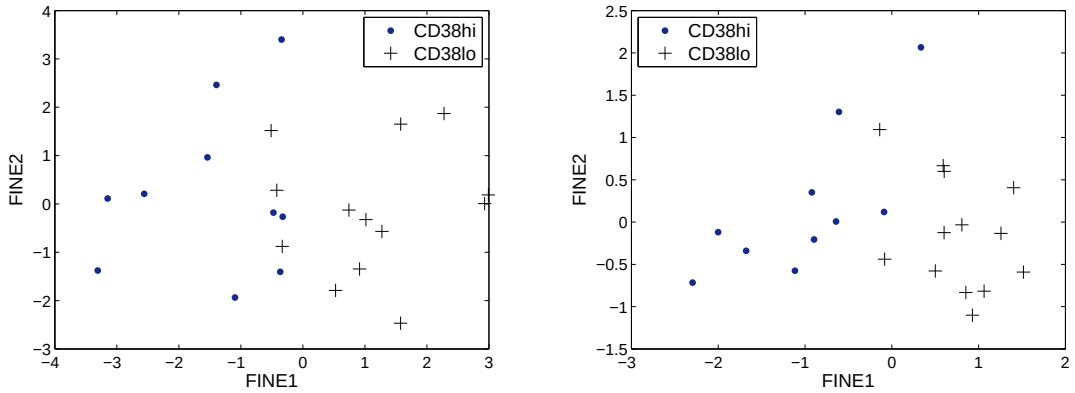
Figure 6.8: CLL Prognosis Study: Contour plots (i.e. PDFs) of the IPCA projected data. The top row corresponds to the PDFs of patients with a poor immunophenotype (CD38hi), while the bottom row represents PDFs of patients with a favorable immunophenotype (CD38lo). The selected patients are those most similar between prognosis classes and those least similar between classes.

Minimizing the objective function (see Fig. 6.7), we calculate the IPCA projection matrix as

$$A = \begin{pmatrix} -0.0700 & 0.0950 & 0.5006 & -0.8361 & 0.1834 & -0.0519 \\ -0.1705 & -0.0434 & -0.3775 & -0.0988 & 0.6992 & 0.5727 \end{pmatrix}.$$

This projection matrix has very high loadings in variables 4, 5, and 6, which correspond to markers CD38, CD45, and CD19 respectively. This identifies the isolation of B cells by CD19 expression (a B lymphocyte restricted antigen always expressed on CLL cells) and assessment of CD38 on these B cells. As expected, we identify CD38 as a marker of importance in differentiating patient groups. We also identify the possibility that CD45 and CD19 expression are also areas which may help prognostic ability. Note that this seems to agree with the intrinsic dimension estimate of $m = 3$. The potential importance of these markers is an area for further interrogation.

We plot the projected patient PDFs in Fig. 6.8. Due to the small sample of



(a) Full Dimensional (b) IPCA Projection

Figure 6.9: CLL Prognosis Study: Comparison of embeddings, obtained with FINE, using the IPCA projection matrix A and the full dimensional data. The patients with a poor immunophenotype (CD38hi) are generally well clustered against those with a favorable immunophenotype (CD38lo) in both embedddings.

patients and their significant similarity to one another, even between prognosis groups (this is evident in Fig. 6.9), we illustrate only those patients with high similarity and low similarity (by information divergence) between the prognosis classes. One can see there is still some difficulty forming a significant discernment between patient prognoses, but the usage of FINE will further help with this task.

6.3.3 FINE

Using FINE to embed the data (Fig. 6.9) for comparative visualization, we see that the different prognosis groups are very similar, although decent clusters are formed when labels are applied. These clusters are not well separated, however, which further illustrates the difficulties in forming an appropriate prognosis. There are also issues of sample size, as a larger database of patients may lead to a more clear separation of clusters. Nonetheless, IPCA and FINE were able to appropriately identify the important markers for assigning prognosis, and group patients accordingly with respect to immunophenotype.

We note that CD38hi versus CD38lo for each patient was determined using cutoff

values endorsed in the literature [29]. Although complete follow-up data for this retrospective cohort were not available, the findings were indirectly further validated by the fact that, of the patients with follow-up information available, zero of 6 CD38lo patients died, while 4 of 9 CD38hi patients died within a median follow-up interval of 25 months (range 1 to 102 months).

6.4 Acute Lymphoblastic Leukemia vs. Hematogone Hyperplasia Study

We now demonstrate a study involving the diseases acute lymphoblastic leukemia (ALL) and a benign condition known as hematogone hyperplasia (HP). ALL is marked by the neoplastic proliferation of abnormal lymphocyte precursors (lymphoblasts). Our study specifically focused upon ALL consisting of B cell precursor lymphoblasts (B-precursor ALL), the most common form of this disease, since the normal counterpart to B-precursor lymphoblasts, termed hematogones, are detectable in the bone marrow of most healthy individuals, and hematogones can proliferate in benign reversible fashion in numerous clinical states [62]. The distinction between hematogones and leukemic B-precursor lymphoblasts is highly relevant in clinical practice since these cell types exhibit substantial immunophenotypic overlap, many transient conditions associated with hematogone hyperplasia can present with clinical suspicion for leukemia, and patients with ALL can develop HP during recovery from chemotherapy for their leukemia.

6.4.1 The Data

For this study, let us define the data set $\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{54}\}$, which consists of 54 patients, 31 of which have been diagnosed with ALL and 23 diagnosed with HP. Patient samples were analyzed with a series of markers (see Table 6.3) designed for the isolation of hematogones and aberrant lymphoblast populations, based on known

Dimension	Marker
1	Forward Light Scatter
2	Side Light Scatter
3	CD38
4	CD19
5	CD45
6	CD10

Table 6.3: Data dimensions and corresponding markers for analysis of ALL and HP.

differential patterns of these markers in these cell types. Specific details of how the data was retrieved can be found in [34]. Intrinsic dimension estimation yields an estimate of $m = 3$ for all samples, suggesting that visualization is indeed plausible with minimal loss of information (even in 2 dimensions).

6.4.2 IPCA

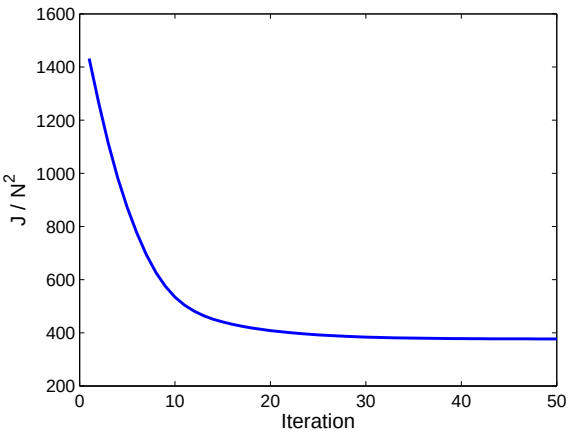
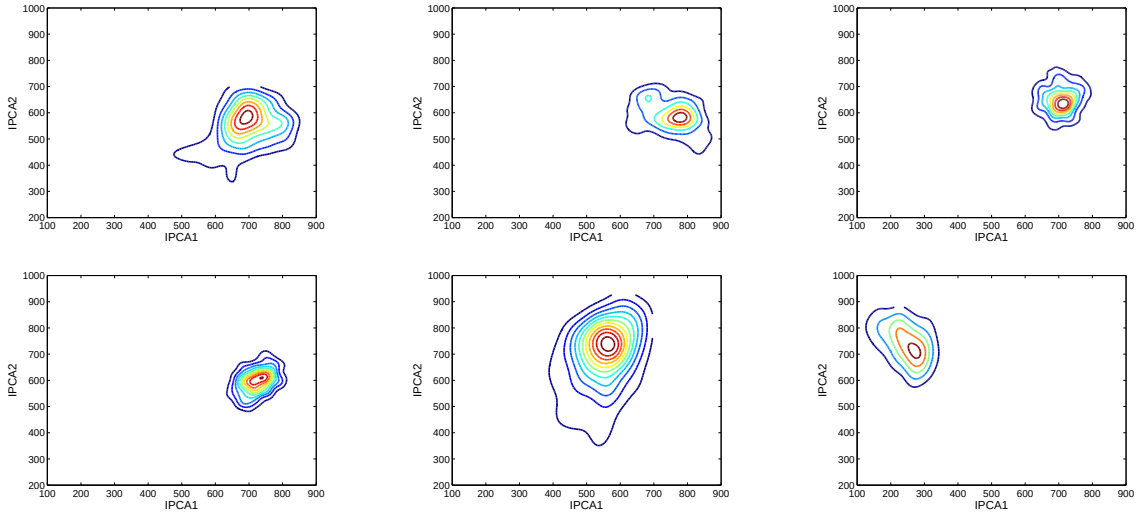


Figure 6.10: ALL and HP Study: The value of the IPCA objective function v.s. time

Minimizing the objective function (Fig. 6.10), we calculate the IPCA projection as

$$A = \begin{pmatrix} -0.1805 & -0.1448 & 0.8691 & 0.0848 & 0.4084 & 0.1310 \\ -0.0336 & 0.1143 & -0.0291 & 0.2506 & -0.2608 & 0.9242 \end{pmatrix}.$$

We observe that the projection matrix has strong loadings corresponding to markers CD38 and CD10. In clinical practice, it is often noted that hematogones have a very uniform and strong CD38 expression pattern, while lymphoblasts can have



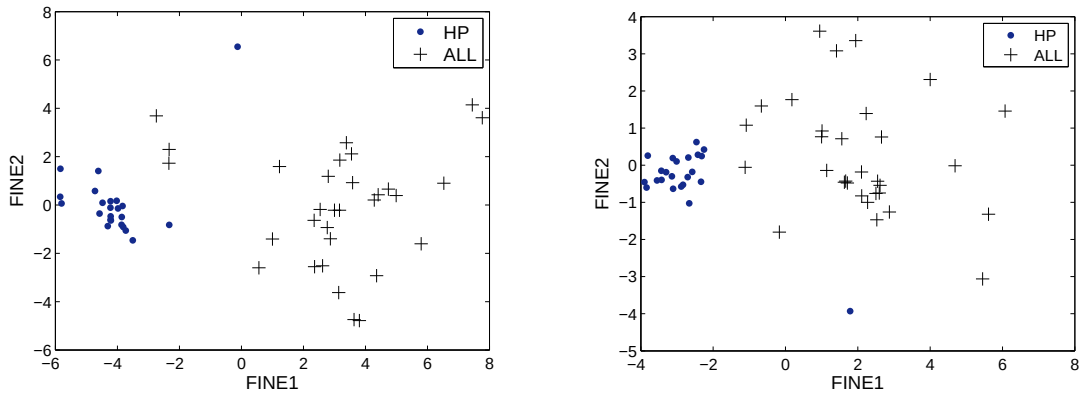
(a) Most Similar (b) Centroids (c) Least Similar

Figure 6.11: ALL and HP Study: Contour plots (i.e. PDFs) of the IPCA projected data. The top row corresponds to the PDFs of the HP patients, while the bottom row represents PDFs of ALL patients. The selected patients are those most similar between disease classes, the centroids of disease classes, and those least similar between disease classes.

quite a range of CD38 expression [62]. This analysis seems to provide independent validation for that observation. Furthermore, this analysis identifies CD10 as a principal distinguishing marker among the others analyzed in this 4-color assay. This finding is not intuitive, since in day-to-day practice CD10 is not obviously of greater distinguishing value than marker such as CD45 or side angle light scatter. These markers, like CD10, are used for their different expression patterns in lymphoblasts versus hematogones, but that may show considerable overlap in expression intensity between these two cell types. Our identification of CD10 as a marker of importance identifies an area for further clinical investigation.

We illustrate the projected data in Fig. 6.11, once again selecting the most and least similar patients between disease classes, as well as the class centroids. We see that the projected PDFs match the clinical experience, as the HP patients have limited variability to their distributions, while the patients with ALL show a wide range of expression patterns.

6.4.3 FINE



(a) Full Dimensional

(b) IPCA Projection

Figure 6.12: ALL and HP Study: Comparison of embeddings, obtained with FINE, using the full dimensional data and the IPCA projection matrix A. The embedding is very similar when using the projected data, which preserves the similarities between patients.

Using FINE, we compare the embedding of the full-dimensional data to that of the projected data in Fig. 6.12. The embeddings are very similar, which illustrates once again that IPCA preserves the similarities between different sets. This allows for a low-dimensional analysis in the projected space with the security of knowing the relationships between patients have been minimally effected.

It is important to note the natural clustering which occurs between the disease classes. Though admittedly more easily viewed given the class labels, there are still two distinct clusters with only a few ambiguous patients. We note that the clusters are even more easily distinguished in 3 dimensions when rotations are available to the viewer, which agrees with our dimension estimation results. We also point out that the behavior of the clusters is indicative of the excepted behavior given clinical experience. Specifically, the ALL cluster of patients is generally spread out while the HP cluster is tightly formed. This is in direct correlation to the clinical experience where, as mentioned previously, it is often noted that hematogones have a very

uniform and strong CD38 expression pattern, while lymphoblasts can have quite a range of CD38 expression [62]. Once again, our methods seem to independently validate this assertion.

6.5 Performance Comparison

We now compare IPCA to the LDA, PCA, and ICA [46] projection matrices for the preceding studies. Given that an ultimate task is visualization for diagnosis and validation, it is important that the disease classes are easily distinguished. For our comparison, we utilize the Bhattacharya distance to measure how distinguishable the "worst case" scenarios are in the projected space – essentially we desire the most similar patients in differing disease classes (i.e. "worst case") to have as little similarity as possible. As previously detailed in Chapter V, the Bhattacharya distance has been used to bound classification error in dimension reduction problems [44], and is directly related to the Chernoff performance bound [37].

Study	DR Method			
	IPCA	PCA	ICA	LDA
Lymphoid	**0.1573**	0.0821	0.0220	0.1097
CLL	**0.0550**	0.0409	0.0326	0.0363
ALL/HP	**0.0624**	0.0532	0.0335	0.0428

Table 6.4: 'Worst case' performance comparison of dimension reduction (DR) methods for flow cytometry studies. Results reported for each case study are of the lowest values of the Bhattacharya distance between patient pairs with differing diseases in the projected space. IPCA outperforms LDA, PCA, and ICA in all cases.

Results are illustrated in Table 6.4, where the best performance is emphasized (larger numbers are more desirable). It is clear that IPCA consistently outperforms both other methods of dimension reduction; concluding that the projection subspace defined by IPCA is best at distinguishing between disease types. Although we do not present them here, we have observed similar results with several other measures of probabilistic distance and cluster similarity. Note that LDA was performed by

assigning a unique class label for each patient. ICA was performed using the FastICA algorithm [46], and the data was pre-processed by whitening and PCA in accordance with [47].

6.5.1 Subsampling Performance

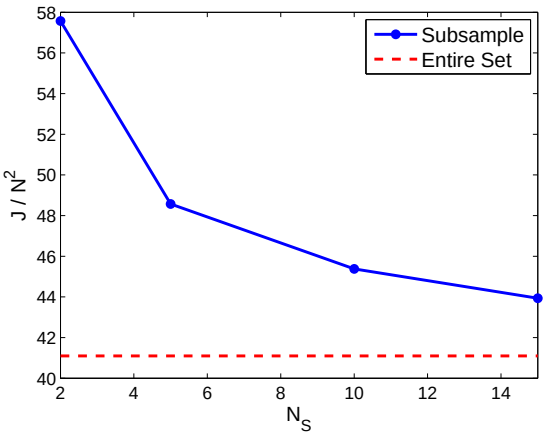


Figure 6.13: IPCA performance using subset of patients $\mathcal{X}_S \subset \mathcal{X}$ from the lymphoid leukemia collection, where N_S is the number of randomly selected patients from each disease class. Results shown over a 10-fold cross validation, with the IPCA projection determined by $\mathcal{X}$ shown as a lower bound with the dotted line.

One concern when implementing IPCA is the number of data sets necessary to find a proper projection. Specifically, given a subset of patients $\mathcal{X}_S \subset \mathcal{X}$, how close does IPCA approach the value of the objective function obtained when utilizing the entire patient collection $\mathcal{X}$? To determine this, we return to our lymphoid leukemia study and subsample from $\mathcal{X}$, with N_S patients randomly selected from each disease class ($N_S \in [2, 5, 10, 15]$), and use IPCA to determine the projection matrix A. We then calculate the value of the objective function on the entire set $\mathcal{X}$ projected by A. The mean results over a 10-fold cross validation are illustrated in Fig. 6.13, where we signify the value of the objection function when using IPCA on the entire data set with the dashed line. Given that the value of the objection function with the initial random projection matrix was $\frac{J}{N^2} = 89.0802$, the relative performance of IPCA with

few available data sets is promising.

6.6 Conclusions and Future Work

In this chapter we have applied our dimensionality reduction techniques towards the field of clinical flow cytometry. By modeling each patient data set as a realization of some PDF lying on a statistical manifold, we have shown that we are able to embed all patients within the same low-dimensional Euclidean space with FINE. This enables a visualization in which direct geometric comparisons may be made between patients of differing disease class in order to validate clinical diagnoses. We used IPCA to discover a low-dimensional projection that allows for visualization in which the data is discernable between immunophenotypes. We have demonstrated these abilities on numerous case studies of differing leukemias and lymphomas, and have shown superiority to other standard methods of dimensionality reduction.

As discussed in Chapter V, analysis of the loading vectors in the IPCA projection matrix allows for a means of variable selection. We have shown independent verification for determining optimal marker combinations in distinguishing immunophenotypically similar cancers, as well as validating variables which help to identify prognostic groups. Verifying these known results through independent methods provides a solid *proof-of-concept* for the ability to utilize IPCA for exploratory research of different marker assays.

The combination of FINE and IPCA has proven useful for verification of cancer diagnosis. The pathology community has shown much interest in this work, and we look forward to continuing our studies. Specifically, we are interested in testing on larger marker-assays, as we feel the true power of our framework is obtained in the high-dimensional regime. When dealing with research grade cytometers, which

reach up to 15-colors, standard methods of flow cytometric analysis are no longer viable and our information-geometric approach will greatly help pathologists and diagnosticians.

CHAPTER VII

Conclusions and Future Work

In this thesis we have approached the problem of dimensionality reduction on statistical manifolds. As opposed to the standard approaches to manifold learning, which aim to reconstruct a Riemannian sub-manifold of Euclidean space, we have taken an information-geometric approach to the problem. We view data sets as realizations of probability density functions lying on a statistical manifold, and aim to reconstruct that manifold in a low-dimensional Euclidean space. This enables the usages of many standard learning algorithms which operate in a Euclidean space, which would not normally be easily applied to PDFs.

We first began by presenting work on local intrinsic dimension estimation. Rather than assuming a constant dimension over an entire data set, we have accounted for multiple supporting manifolds of varying dimensionality, and estimated the dimension of the supporting manifold for each sample. Not only has this proven useful for dimensionality reduction, but we have also presented several practical applications for which the information gained from intrinsic dimension is worthwhile. We have shown the ability to detect anomalies in time-series data sets, cluster data based on complexity, and segment images over various levels of detail.

Given the intrinsic dimension of the statistical manifold, we have shown the abil-

ity to reconstruct that manifold in a low-dimensional Euclidean space. Standard methods of multidimensional scaling, which have been thoroughly utilized in manifold learning, are easily adapted towards the reconstruction of statistical manifolds given an appropriate measure of dissimilarity between PDFs. Although the Fisher information distance – which is the natural metric of distance on a statistical manifold – cannot be calculated without knowledge of the manifold parameterization, we have formed good approximations of the distance using graphical methods alongside one of several information divergences. This accurate measure of dissimilarity, in conjunction with standard MDS methods, enabled the embedding of PDFs into an open Euclidean space; a process we refer to a *Fisher Information Nonparametric Embedding*. If the additional constraint of embedding onto the surface of a sphere is desired, we offer *Spherical Laplacian Information Maps*. Both FINE and SLIM may be used for a comparative visualization of PDFs, as well as additional learning techniques which are not available in the probabilistic space. This has been shown for document classification as well as object recognition.

Finally, when dimensionality reduction is desirable in the data domain, we have presented an algorithm deemed *Information Preserving Component Analysis* which finds the optimal low-dimensional subspace which preserves the high-dimensional Fisher information distances between PDFs. Contrary to standard unsupervised projection methods, which aim to find the optimal low-dimensional representation of a single data set, the IPCA projection space ensures the best representation of a data set with respect to all of the available data sets to which it will be compared. The projection matrix itself is useful as a means of variable selection, as the dimensions with the highest loading values will be those which contribute most to the information distance between PDFs. This was demonstrated on the analysis of spam patterns,

in which we were able to potentially identify community properties. We have also shown the ability to use IPCA as a supervised method for dimensionality reduction for the classification task, as the formulation of IPCA has a direct correlation to the Chernoff bound for classification error. IPCA has shown superior performance to standard supervised methods of dimensionality reduction on real data.

We have used all of the methods presented here in collaboration towards the problems of diagnosis and visualization of flow cytometry data. Current forms of analysis, which rely on 2-dimensional axes projections, are both antiquated and prone to user error. Through the use of FINE, we were able to embed patients into a common space and enable diagnosticians to view a patient with an unspecified disease in relation to a database of patients with potentially similar immunophenotype. As opposed to traditional axes projections, IPCA offers a projection space which is formed through a linear projection of all available markers, weighted according to importance in preserving information distance. In this low-dimensional space, visualization is simple and we have shown that similar immunophenotypes are distinguishable, even in the worst case scenarios. Additionally, when using the IPCA projection matrix for variable selection, we have found agreement with standard clinical knowledge and practice. This is critical as it offers a *proof-of-concept* for the use of IPCA towards exploratory research, aiming to find new and distinguishing marker assays.

In future work, we would like to offer convergence proofs for the geodesic approximation of the Fisher information distance. While we have shown empirical results, illustrating the asymptotic convergence properties, we would like to also prove these results analytically, providing details on specific conditions for which these properties hold. This is similar to the way Tenenbaum *et al.* have provided convergence proofs for their Isomap algorithm [7]. Additionally, we would like to continue studying the

benefits of our framework on flow cytometry analysis. Specifically, we aim to eventually run a clinical study, polling pathologists to obtain information on the potential for clinical usage of our methods. This could be both qualitative in terms of ease of use, as well as quantitative with respect to rate of misdiagnosis. Finally, we would like to perform exploratory analysis with research-grade cytometers (which offer much higher dimensionality) to potentially discover new and distinguishing marker assays towards leukemias and lymphomas which are currently undistinguishable through flow cytometry.

APPENDICES

APPENDIX A

Kernel Density Estimation

Kernel methods are nonparametric techniques used for estimating probability densities of data sets. These methods are similar to mixture-models in that they are defined by the normalized sum of multiple densities. Unlike mixture models, however, kernel methods are nonparametric and are comprised of the normalized sum of identical densities centered about each data point within the set (A.1). This yields a density estimate for the entire set in that highly probable regions will have more samples, and the sum of the kernels in those areas will be large, corresponding to a high probability in the resultant density.

We now illustrate the derivation of the kernel density estimate (KDE) of the PDF $f(x)$ of the realization $\boldsymbol{X}_f$. We utilize Gaussian kernels as the quadratic properties will be useful in implementation. Specifically, the KDE of a PDF is defined as

$$\hat{f}(x) = \frac{1}{n_f \cdot h} \sum_{j=1}^{n_f} K\left(\frac{x - x_j}{h}\right), \tag{A.1}$$

where n_f is the number of sample points $x_j \in \boldsymbol{X}_f$, K is some kernel satisfying the properties

$$K(x) \geqslant 0, \, \forall x \in \mathcal{X},$$

$$\int K(x)\, dx = 1,$$

and h is the bandwidth or smoothing parameter. By utilizing the Gaussian kernel

$$K(x) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right),$$

where Σ is the covariance of the kernel, we may combine the smoothing parameter vector h with Σ (ie. $\Sigma = I$) such that $H_f = \text{diag}(h)$. Note that we implement a vector bandwidth such that our Gaussian kernels are ellipses rather than spheres. There are a variety of methods for determining this bandwidth parameter; we choose to implement the maximal smoothing principle [75]. This yields the a final kernel density estimate of

$$\hat{f}(x) = \frac{1}{n_f}\sum_{j=1}^{n_f} \frac{1}{\sqrt{|2\pi H_f|}} \exp\left(-\frac{1}{2}(x-x_j)^T H_f^{-1}(x-x_j)\right). \tag{A.2}$$

Let us now make the following definitions:

$$\begin{aligned} D_j^{(f)} &= (x-x_j^{(f)})^T H_f^{-1}(x-x_j^{(f)}) \\ W^{(f)} &= \mathrm{e}^{-D^{(f)}/2}, \end{aligned} \tag{A.3}$$

where $D_j^{(f)}$ is a Mahalanobis distance between the point x and sample points $x_j^{(f)} \in \boldsymbol{X}_f$, and $D^{(f)}$ is the vector with elements $D_j^{(f)}$. Substituting (A.3) into (A.2), we obtain

$$\hat{f}(x) = \frac{1}{n_f}\sum_{j=1}^{n_f} \frac{1}{\sqrt{|2\pi H_f|}} W^{(f)}, \tag{A.4}$$

the KDE approximation of the PDF generating $\boldsymbol{X}_f$.

We note that the mean squared error of a KDE decreases only as $n^{-\mathrm{O}(1/\mathrm{d})}$, which becomes extremely slow for large d. As such, it may be difficult to calculate good kernel density estimates. However, for our purposes, the estimation of densities is secondary to the estimation of the divergence between them. As such, the issues with MSE of density estimates in large dimensions, while an area for future work, is not of immediate concern.

APPENDIX B

Implementation Details

We now detail the calculation of the approximation of the Fisher information distance between two realizations of PDFs. Specifically, let $\boldsymbol{X}_f$ and $\boldsymbol{X}_g$ be realizations of PDFs $f(x)$ and $g(x)$ respectively. Our goal is to calculate both an approximation of the Fisher information distance between the PDFs as well as the direction of the gradient with respect to a projection matrix A. Let us now illustrate the difficulties with these computations.

Recall that the Hellinger distance (squared) is defined as

$$D_H^2(f(x), g(x)) = \int \left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 dx. \tag{B.1}$$

Given the limited (and often unknown) support of x in both $f(x)$ and $g(x)$, it is appropriate to reformat this definition in terms of an expected value with respect to a single density $f(x)$ or $g(x)$:

$$D_H^2(f(x), g(x)) = \begin{cases} \int \left(1 - \sqrt{\frac{g(x)}{f(x)}}\right)^2 f(x)dx \\ \int \left(1 - \sqrt{\frac{f(x)}{g(x)}}\right)^2 g(x)dx \end{cases}. \tag{B.2}$$

These equations may be numerically approximated as follows:

$$\hat{D}_H^2(f(x), g(x)) = \begin{cases} \frac{1}{n_f} \sum_{i=1}^{n_f} \left(1 - \sqrt{\frac{\hat{g}\left(x_i^{(f)}\right)}{\hat{f}\left(x_i^{(f)}\right)}}\right)^2 \\ \frac{1}{n_g} \sum_{i=1}^{n_g} \left(1 - \sqrt{\frac{\hat{f}\left(x_i^{(g)}\right)}{\hat{g}\left(x_i^{(g)}\right)}}\right)^2 \end{cases},$$

in which n_f and n_g are the number of samples $x_i^{(f)} \in \boldsymbol{X}_f$ and $x_i^{(g)} \in \boldsymbol{X}_g$, and $\hat{f}(x)$ and $\hat{g}(x)$ are the kernel density estimates of PDFs $f(x)$ and $g(x)$ (see Appendices B.4 and A). The problem with these approximations is that they yield a non-symmetric estimate of the Hellinger distance $\hat{D}_H^2(f(x), g(x)) \neq \hat{D}_H^2(g(x), f(x))$. Additionally, the estimate is unbounded from above. By definition the Hellinger distance should be symmetric and bounded by 2 (for the squared distance).

When approximating the Kullback-Leibler divergence, a similar approach of formatting as an expectation may seem natural. The definition of the KL divergence

$$KL(f(x)\|g(x)) = \int f(x) \log \frac{f(x)}{g(x)}\, dx \tag{B.3}$$

in turn would be approximated as

$$\hat{KL}(f\|g) = \frac{1}{n_f} \sum_{i=1}^{n_f} \log \frac{\hat{f}\left(x_i^{(f)}\right)}{\hat{g}\left(x_i^{(f)}\right)}.$$

Note that by definition the KL divergence is not necessarily symmetric, however it is strictly non-negative. This numerical approximation does not guarantee this non-negativity.

B.1 Metric Calculation

We now detail our approximations which do not suffer from the aforementioned pitfalls. Define

$$T(x) = \frac{f(x)}{f(x) + g(x)}$$

and

$$\hat{T}(x) = \frac{\hat{f}(x)}{\hat{f}(x) + \hat{g}(x)}. \tag{B.4}$$

Note that $0 \leqslant T(x) \leqslant 1$.

For simplicity, let us write $f = f(x)$, $g = g(x)$, and $T = T(x)$. The Hellinger distance (squared) may be computed as follows:

$$\begin{aligned} D_H^2(f,g) &= \int \left(\sqrt{f} - \sqrt{g}\right)^2 dx \\ &= \int \left(\sqrt{\frac{f}{f+g}} - \sqrt{\frac{g}{f+g}}\right)^2 (f+g)dx \\ &= \int \left(\sqrt{T} - \sqrt{1-T}\right)^2 f\, dx + \int \left(\sqrt{T} - \sqrt{1-T}\right)^2 g\, dx. \end{aligned} \tag{B.5}$$

Hence, we may now define our numerical approximation of the squared Hellinger distance as:

$$\begin{aligned} \hat{D}_H^2(f,g) &= \frac{1}{n_f}\sum_{i=1}^{n_f}\left(\sqrt{\hat{T}\left(x_i^{(f)}\right)} - \sqrt{1-\hat{T}\left(x_i^{(f)}\right)}\right)^2 \\ &\quad + \frac{1}{n_g}\sum_{i=1}^{n_g}\left(\sqrt{\hat{T}\left(x_i^{(g)}\right)} - \sqrt{1-\hat{T}\left(x_i^{(g)}\right)}\right)^2, \end{aligned} \tag{B.6}$$

which is both symmetric and bounded above by 2.

The same formulation may be implemented when approximating the Kullback-Leibler divergence. Specifically,

$$\begin{aligned} KL(f\|g) &= \int f \log \frac{f}{g}\, dx \\ &= \int \frac{f}{f+g} \log\left(\frac{f}{f+g} / \frac{g}{f+g}\right)(f+g)\, dx \\ &= \int T \log \frac{T}{1-T} f\, dx + \int T \log \frac{T}{1-T} g\, dx. \end{aligned} \tag{B.7}$$

Hence

$$\begin{aligned} \hat{KL}(f\|g) &= \frac{1}{n_f}\sum_{i=1}^{n_f} \hat{T}\left(x_i^{(f)}\right) \log \frac{\hat{T}\left(x_i^{(f)}\right)}{1-\hat{T}\left(x_i^{(f)}\right)} \\ &\quad + \frac{1}{n_g}\sum_{i=1}^{n_g} \hat{T}\left(x_i^{(g)}\right) \log \frac{\hat{T}\left(x_i^{(g)}\right)}{1-\hat{T}\left(x_i^{(g)}\right)}, \end{aligned} \tag{B.8}$$

which no longer suffers from the issue of potential negativity. This value is still non-symmetric, and when dealing with metric learning it is often desirable to have a

symmetric dissimilarity metric. Hence, we implement the symmetric KL divergence (2.10) which maybe calculated in a similar manner:

$$\begin{aligned} D_{KL}(f,g) &= KL(f\|g) + KL(g\|f) \\ &= \int f \log \frac{f}{g}\, dx + \int g \log \frac{g}{f}\, dx \\ &= \int (f-g) \log \frac{f}{g}\, dx \\ &= \int (2T-1) \log \frac{T}{1-T} f\, dx + \int (2T-1) \log \frac{T}{1-T} g\, dx, \end{aligned} \tag{B.9}$$

yielding

$$\begin{aligned} \hat{D}_{KL}(f,g) &= \frac{1}{n_f} \sum_{i=1}^{n_f} (2\hat{T}\left(x_i^{(f)}\right) - 1) \log \frac{\hat{T}\left(x_i^{(f)}\right)}{1-\hat{T}\left(x_i^{(f)}\right)} \\ &\quad + \frac{1}{n_g} \sum_{i=1}^{n_g} (2\hat{T}\left(x_i^{(g)}\right) - 1) \log \frac{\hat{T}\left(x_i^{(g)}\right)}{1-\hat{T}\left(x_i^{(g)}\right)}. \end{aligned} \tag{B.10}$$

The Bhattacharya distance may be numerically formulated in the same manner as the KL divergence and Hellinger distance:

$$\begin{aligned} D_B(f,g) &= -\log \int \sqrt{f}\sqrt{g}\, dx \\ &= -\log \int \sqrt{f}\sqrt{g}\, \frac{f+g}{f+g}\, dx \\ &= -\log \left[\int \sqrt{T(1-T)} f\, dx + \int \sqrt{T(1-T)} g\, dx \right] \\ \hat{D}_B(f,g) &= -\log \left[\frac{1}{n_f} \sum_{i=1}^{n_f} \sqrt{\hat{T}\left(x_i^{(f)}\right)\left(1-\hat{T}\left(x_i^{(f)}\right)\right)} \right. \\ &\quad \left. - \frac{1}{n_g} \sum_{i=1}^{n_g} \sqrt{\hat{T}\left(x_i^{(g)}\right)\left(1-\hat{T}\left(x_i^{(g)}\right)\right)} \right]. \end{aligned} \tag{B.11}$$

In order to numerically approximate these information divergences, we calculate $\hat{T}(x)$ as described in Section B.4.

B.2 Gradient Calculation

In Appendix B.1 we detailed expressions of the form

$$D = \frac{1}{n_f}\sum_{i=1}^{n_f} G\left(\hat{T}(x_i^{(f)})\right) + \frac{1}{n_g}\sum_{i=1}^{n_g} G\left(\hat{T}(x_i^{(g)})\right),$$

which were used to numerically approximate the Hellinger distance and Kullback-Leibler divergence between PDFs $f(x)$ and $g(x)$. For simplicity, we write

$$D = \frac{1}{n_f}\sum_{i=1}^{n_f} G(\hat{T}) \mid_{x_i^{(f)}} + \frac{1}{n_g}\sum_{i=1}^{n_g} G(\hat{T}) \mid_{x_i^{(g)}} .$$

Note that we do not continue with the Bhattacharya distance as we are mainly interested in that measure for final comparison to other dimension reduction methods, and we do not calculate the gradient w.r.t. this distance. If desired, the gradient may be calculated by a analytic transformation of the Hellinger distance gradient.

The gradient of D w.r.t. some parameter θ, to which T yields some dependency, is defined as

$$\frac{\partial D}{\partial \theta} = \frac{1}{n_f}\sum_{i=1}^{n_f} \frac{\partial G}{\partial T}\frac{\partial T}{\partial \theta} \mid_{x_i^{(f)}} + \frac{1}{n_g}\sum_{i=1}^{n_g} \frac{\partial G}{\partial T}\frac{\partial T}{\partial \theta} \mid_{x_i^{(g)}}, \tag{B.12}$$

where

$$\frac{\partial T}{\partial \theta} = T(1-T)\left(\frac{\partial}{\partial \theta}\log f - \frac{\partial}{\partial \theta}\log g\right). \tag{B.13}$$

This derivation is explained in Appendix B.5. Substituting (B.13) into (B.12), the gradient may be numerically approximated as

$$\begin{aligned}\frac{\partial \hat{D}}{\partial \theta} &= \frac{1}{n_f}\sum_{i=1}^{n_f} \hat{T}(1-\hat{T})\frac{\partial G}{\partial T}\left(\frac{\partial}{\partial \theta}\log f - \frac{\partial}{\partial \theta}\log g\right) \mid_{x_i^{(f)}} \\ &+ \frac{1}{n_g}\sum_{i=1}^{n_g} \hat{T}(1-\hat{T})\frac{\partial G}{\partial T}\left(\frac{\partial}{\partial \theta}\log f - \frac{\partial}{\partial \theta}\log g\right) \mid_{x_i^{(g)}}.\end{aligned} \tag{B.14}$$

Given this general setting, it is important to recognize that the only difference between the formulation of the Hellinger distance, KL divergence, and symmetric

KL divergence is the definition of $G(T)$. Hence, the formulation of the gradient is unchanged for all metrics, given a different definition of $G(T)$. We now derive the value of $T(1-T)\frac{\partial G}{\partial T}$ for each metric.

B.2.1 Hellinger Distance

From (B.5) we see that $G(T) = \left(\sqrt{T} - \sqrt{1-T}\right)^2$. Therefore,

$$\begin{aligned}\frac{\partial G}{\partial T} &= (\sqrt{T} - \sqrt{1-T})\left(\frac{1}{\sqrt{T}} + \frac{1}{\sqrt{1-T}}\right) \\ &= \sqrt{\frac{T}{1-T}} - \sqrt{\frac{1-T}{T}} \\ &= \frac{2T-1}{\sqrt{(1-T)T}}.\end{aligned}$$

Hence,

$$T(1-T)\frac{\partial G}{\partial T} = \sqrt{T(1-T)}\,(2T-1). \tag{B.15}$$

B.2.2 Kullback-Leibler Divergence

From (B.7) we see that $G(T) = T\log\frac{T}{1-T}$. Therefore,

$$\frac{\partial G}{\partial T} = \log\left(\frac{T}{1-T}\right) + \frac{1}{1-T}. \tag{B.16}$$

Hence,

$$T(1-T)\frac{\partial G}{\partial T} = T(1-T)\log\left(\frac{T}{1-T}\right) + T. \tag{B.17}$$

For the symmetric KL divergence, (B.9) yields that $G(T) = (2T-1)\log\frac{T}{1-T}$. Therefore,

$$\frac{\partial G}{\partial T} = 2\log\left(\frac{T}{1-T}\right) + \frac{2T-1}{T(1-T)}.$$

Hence,

$$T(1-T)\frac{\partial G}{\partial T} = 2T(1-T)\log\left(\frac{T}{1-T}\right) + 2T - 1. \tag{B.18}$$

B.3 Matrix Gradient

We now specify our abstraction to the specific task for IPCA, which is calculating the gradient of D w.r.t. the projection matrix A. First, let us derive $\frac{\partial}{\partial\theta}\log f = \frac{\partial}{\partial A}\log f(Ax)$ in which $\hat{f}(Ax)$ may be estimated with kernel density estimation methods described in Appendix A, with kernel locations $x_j^{(f)} \in \boldsymbol{X}_f$.

$$\begin{aligned}\frac{\partial}{\partial A}\log\hat{f}(Ax)|_{x^{(f)}} &= \frac{\partial}{\partial A}\log\left(\frac{1}{n_f}\sum_{j=1}^{n_f}\frac{1}{\sqrt{|2\pi H_f|}}\,\mathrm{e}^{-\frac{1}{2}(x-x_j^{(f)})^T A^T H_f^{-1} A(x-x_j^{(f)}))}\right)\\ &= \sum_{j=1}^{n_f}\bar{W}_j^{(f)}\left(-H_f^{-1}A(x-x_j^{(f)})(x-x_j^{(f)})^T\right)\\ &= -H_f^{-1}AC^{(f)}(x), \qquad \text{(B.19)}\end{aligned}$$

where H_f is the kernel bandwidth for set $\boldsymbol{X}_f$,

$$\bar{W}_j^{(f)} = \frac{\exp\left(-\frac{1}{2}(x-x_j^{(f)})^T A^T H_f^{-1} A(x-x_j^{(f)})\right)}{\sum_{l=1}^{n_f}\exp\left(-\frac{1}{2}(x-x_l^{(f)})^T A^T H_f^{-1} A(x-x_l^{(f)})\right)},$$

and

$$C^{(f)}(x) = \sum_{j=1}^{n_f}\bar{W}_j^{(f)}(x-x_j^{(f)})(x-x_j^{(f)})^T$$

is the weighted sample covariance around x. In the same manner,

$$\frac{\partial}{\partial A}\log\left(\hat{g}(Ax)\right)|_{x^{(g)}} = -H_g^{-1}AC^{(g)}(x),$$

by evaluating the KDE with points $x_j^{(g)} \in \boldsymbol{X}_g$.

Finally, we may now define the gradient $\frac{\partial}{\partial A}\hat{D}$ as follows

$$\begin{aligned}\frac{\partial}{\partial A}\hat{D} &= \frac{1}{n_f}\sum_{i=1}^{n_f}\hat{T}(x_i^{(f)})(1-\hat{T}(x_i^{(f)}))\frac{\partial G}{\partial T}(x_i^{(f)})\left[(-H_f^{-1}AC^{(f)}(x_i^{(f)})) - \ldots\right.\\ &\quad \left.-(-H_g^{-1}AC^{(g)}(x_i^{(f)}))\right]\\ &\quad +\frac{1}{n_g}\sum_{i=1}^{n_g}\hat{T}(x_i^{(g)})(1-\hat{T}(x_i^{(g)}))\frac{\partial G}{\partial T}(x_i^{(g)})\left[(-H_f^{-1}AC^{(f)}(x_i^{(g)})) - \ldots\right.\\ &\quad \left.-(-H_g^{-1}AC^{(g)}(x_i^{(g)}))\right]. \qquad \text{(B.20)}\end{aligned}$$

B.4 Numerical Implementation

B.4.1 PDFs and $T(x)$

To estimate the PDFs $f(x)$ and $g(x)$, let us begin with the following definitions:

$$\begin{aligned} D_{ij}^{(f,g)} &= (x_i^{(f)} - x_j^{(g)})^T H_f^{-1} (x_i^{(f)} - x_j^{(g)}) \\ W_{ij}^{(f,g)} &= \mathrm{e}^{-D_{ij}^{(f,g)}/2} \\ W^{(f,g)} &= \left[W_{ij}^{(f,g)} \right], \end{aligned} \tag{B.21}$$

which in conjunction with the Gaussian KDE illustrated in Appendix A, yield the density estimates:

$$\begin{aligned} \hat{f}(x)|_{x^{(\star)}} &= \frac{1}{n_\star} \frac{1}{\sqrt{|2\pi H_f|}} W^{(f,\star)} \mathbf{1} \\ \hat{g}(x)|_{x^{(\star)}} &= \frac{1}{n_\star} \frac{1}{\sqrt{|2\pi H_g|}} W^{(g,\star)} \mathbf{1}, \end{aligned} \tag{B.22}$$

where $\mathbf{1}$ is the vector of all ones and $\star$ is either f or g. Essentially, given the limited support of each $\boldsymbol{X}_f$ and $\boldsymbol{X}_g$, we approximate the densities and their derivatives w.r.t. the samples in an appropriate set. Hence, $\hat{f}(x)|_{x^{(\star)}}$ is an $n_\star$ element vector with elements equal to $\hat{f}(x^{(\star)})$, $x^{(\star)} \in \boldsymbol{X}_\star$. A similar interpretation holds for $\hat{g}(x)|_{x^{(\star)}}$.

Plugging the density estimates (B.22) into (B.4), we calculate our final estimates of $T(x)$:

$$\begin{aligned} \hat{T}(x^{(f)}) &= \frac{1}{\sqrt{|2\pi H_f|}} W^{(f,f)} \mathbf{1} ./ \left(\frac{1}{\sqrt{|2\pi H_f|}} W^{(f,f)} \mathbf{1} + \frac{1}{\sqrt{|2\pi H_g|}} W^{(g,f)} \mathbf{1} \right) \\ \hat{T}(x^{(g)}) &= \frac{1}{\sqrt{|2\pi H_f|}} W^{(f,g)} \mathbf{1} ./ \left(\frac{1}{\sqrt{|2\pi H_f|}} W^{(f,g)} \mathbf{1} + \frac{1}{\sqrt{|2\pi H_g|}} W^{(g,g)} \mathbf{1} \right) \end{aligned}$$

where the notation ./ signifies element-by-element vector division.

B.4.2 Gradient

We now describe the implementation of (B.20). Specifically, let us numerically calculate

$$Z(f,g) = \frac{1}{n_f}\sum_{i=1}^{n_f} \hat{T}(x_i^{(f)})(1-\hat{T}(x_i^{(f)}))\frac{\partial G}{\partial T}(x_i^{(f)})\left(H_g^{-1}AC^{(g)}(x_i^{(f)})\right),$$

and extend towards the 3 other similar formulations such that

$$\frac{\partial}{\partial A}\hat{D} = Z(f,g) - Z(f,f) + Z(g,g) - Z(g,f). \tag{B.23}$$

Let us continue (B.21) with the following additional definitions:

$$\begin{aligned} \bar{W}_{ij}^{(f,g)} &= W_{ij}^{(f,g)}/(W_{ij}^{(f,g)}\mathbf{1}\mathbf{1}^T) \\ S_{ij}^{(f,g)} &= \hat{T}(1-\hat{T})\frac{\partial G}{\partial T}|_{x_i^{(f)}} \cdot \bar{W}_{ij}^{(f,g)} \\ S^{(f,g)} &= \left[S_{ij}^{(f,g)}\right]. \end{aligned} \tag{B.24}$$

The formulation continues as follows:

$$\begin{aligned} Z(f,g) &= \frac{1}{n_f}H_g^{-1}A\sum_{i=1}^{n_f}\sum_{j=1}^{n_g}\hat{T}(1-\hat{T})\frac{\partial G}{\partial T}|_{x_i^{(f)}} \cdot \bar{W}_{ij}^{(f,g)}(x_i^{(f)}-x_j^{(g)})(x_i^{(f)}-x_j^{(g)})^T \\ &= \frac{1}{n_f}H_g^{-1}A\sum_{i=1}^{n_f}\sum_{j=1}^{n_g}S_{ij}^{(f,g)}(x_i^{(f)}-x_j^{(g)})(x_i^{(f)}-x_j^{(g)})^T \\ &= \frac{1}{n_f}H_g^{-1}A\left[\boldsymbol{X}_f\mathrm{diag}(S^{(f,g)}\mathbf{1})\boldsymbol{X}_f^T - \boldsymbol{X}_f S^{(f,g)}\boldsymbol{X}_g^T\right. \\ &\quad \left. - \boldsymbol{X}_g(S^{(f,g)})^T\boldsymbol{X}_f^T + \boldsymbol{X}_g\mathrm{diag}((S^{(f,g)})^T\mathbf{1})\boldsymbol{X}_g^T\right]. \end{aligned} \tag{B.25}$$

Equation (B.25) and similar formulations (replacing f and g where appropriate) may be substituted into (B.23) to obtain the final calculation of the gradient $\frac{\partial}{\partial A}\hat{D}$.

B.5 Gradient of T

Calculation of the gradient of $T = \frac{f}{f+g}$ w.r.t. some parameter θ. Let $F_\theta = \frac{\partial}{\partial\theta}F$ for some arbitrary function F.

$$\begin{aligned}
\frac{\partial T}{\partial \theta} &= T\frac{\partial}{\partial\theta}\log T \\
&= T\frac{\partial}{\partial\theta}\left(\log f - \log\left(f+g\right)\right) \\
&= T\left(\frac{f_\theta}{f} - \frac{f_\theta + g_\theta}{f+g}\right) \\
&= T\left(\frac{f_\theta g - g_\theta f}{f(f+g)}\right) \\
&= T\left((1-T)\frac{f_\theta}{f} - \frac{g}{f+g}\frac{g_\theta}{g}\right) \\
&= T\left((1-T)(\log f)_\theta - (1-T)(\log g)_\theta\right) \\
&= T(1-T)\left(\frac{\partial}{\partial\theta}\log f - \frac{\partial}{\partial\theta}\log g\right)
\end{aligned}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] *UCI Machine Learning Repository: Statlog (Landsat Satellite) Data Set.* available at `http://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)`.

[2] S. Amari and H. Nagaoka. *Differential-geometrical methods in statistics.* Springer, 1990.

[3] S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191. American Mathematical Society and Oxford University Press, 2000. Translations of mathematical monographs.

[4] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *Proceedings IEEE Conf. On Computer Vision and Pattern Recognition*, pages 581–588, June 2005.

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems, Volume 14*. MIT Press, 2002.

[6] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[7] M. Bernstein, V. de Silva, J. C. Langford, and J. B. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Department of Psychology, Stanford University, 2000.

[8] J. C. Bezdec. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[9] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[10] F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(10):1404–1407, October 2002.

[11] K. M. Carter and A. O. Hero. Variance reduction with neighborhood smoothing for local dimension estimation. In *Proc. IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3917–3920, April 2008.

[12] K. M. Carter, A. O. Hero, and R. Raich. De-biasing for intrinsic dimension estimation. In *Proc. IEEE Statistical Signal Processing Workshop*, pages 601–605, August 2007.

[13] K. M. Carter, C. Kyung min Kim, R. Raich, and A. O. Hero III. Information preserving embeddings for discrimination. In *Proc. of IEEE Signal Processing Society DSP Workshop*, Jan. 2009.

[14] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero. Fine: Fisher information non-parametric embedding. *IEEE Transactions on Pattern Analyis and Machine Intelligence*, 2008. submitted.

[15] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero. Information preserving component analysis: Data projections for flow cytometry analysis. *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Digital Image Processing Techniques for Oncology*, Feb. 2009. to appear.

[16] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero III. Dimensionality reduction of flow cytometric data through information preservation. In *Proc. of IEEE Machine Learning for Signal Processing Workshop*, Oct. 2008.

[17] K. M. Carter, R. Raich, and A. O. Hero. On local intrinsic dimension estimation and its applications. *IEEE Transactions on Signal Processing.* in review.

[18] K. M. Carter, R. Raich, and A. O. Hero. Learning on statistical manifolds for clustering and visualization. In *Proc. of 45th Annual Allerton Conf. on Communication, Control, and Computing*, pages 526–533, September 2007.

[19] K. M. Carter, R. Raich, and A. O. Hero. Fine: Information embedding for document classification. In *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 1861–1864, April 2008.

[20] K. M. Carter, R. Raich, and A. O. Hero III. An information geometric approach to supervised dimensionality reduction. In *Proc. of IEEE Signal Processing Society DSP Workshop*, April 2009. in review.

[21] C.-C. Chang and C.-J. Lin. *LIBSVM: A library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[22] B. B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):72–77, Jan. 1995.

[23] J. Costa, A. Girotra, and A. Hero. Estimating local intrinsic dimension with k-nearest neighbor graphs. *IEEE Workshop on Statistical Signal Processing (SSP)*, July 2005.

[24] J. Costa and A. O. Hero. *Statistics and analysis of shapes*, chapter Learning Intrinsic Dimension and Entropy of High-Dimensional Shape Spaces, pages 231–252. Birkhauser, 2006.

[25] J. A. Costa and A. O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. on Signal Processing*, 52(8):2210–2221, August 2004.

[26] S.I.R. Costa, S. Santos, and J. Strapasson. Fisher information matrix and hyperbolic geometry. In *Proceedings of IEEE ITSOC Information Theory Workshop on Coding and Complexity*, August 2005.

[27] T. Cox and M. Cox. *Multidimensional Scaling.* Chapman & Hall, London, 1994.

[28] I. Csiszár. Information type measures of differences of probability distribution and indirect observations. *Studia Sci. Math. Hungarica 2*, pages 299–318, 1967.

[29] R. N. Damle, T. Wasil, F. Fais, F. Ghiotto, A. Valetto, S. L Allen, and et. al. Ig v gene mutation status and cd38 expression as novel prognostic indicators in chronic lymphocytic leukemia. *Blood*, 95(7):1840–1847, 1999.

[30] T. Dietterich. Ai seminar. Carnegie Mellon, 2002.

[31] S. C. Douglas. On the design of gradient algorithms employing orthogonal matrix constraints. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1401–1404, April 2007.

[32] Z. Duan, K. Gopalan, and X. Yuan. Behavioral characteristics of spammers and their network reachability properties. In *Proc. of IEEE Int. Conf. on Comm.*, pages 164–171, June 2007.

[33] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, April 1999.

[34] W. G. Finn, K. M. Carter, R. Raich, and A. O. Hero. Analysis of clinical flow cytometric immunophenotyping data by clustering on statistical manifolds: Treating flow cytometry data as high-dimensional objects. *Cytometry Part B: Clinical Cytometry*, 76B(1), Jan. 2009.

[35] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, March 1989.

[36] J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. Technical report, Stanford University, 2005.

[37] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990. 2nd edition.

[38] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Neural Information Processing Systems*, number 17, pages 513–520, 2004.

[39] S. Grikshchat, J. A. Costa, A. O. Hero, and O. Michel. Dual rooted-diffusions for clustering and classification on manifolds. In *Proc. 2006 IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, volume 5, May 2006.

[40] L. K. Habib and W. G. Finn. Unsupervised immunophenotypic profiling of chronic lymphocytic leukemia. *Cytometry Part B: Clinical Cytometry*, 70B(3):124–135, 2006.

[41] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[42] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[43] M. E. Hellman and J. Raviv. Probability of error, equivocation, and the chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372, July 1970.

[44] P. F. Hsieh, D. S. Wang, and C. W. Hsu. A linear feature extraction for multiclass classification problems based on class mean and covariance discriminant information. *IEEE Transactions on Pattern Analalysis and Machine Intelligence*, 28(2):223–235, Feb. 2006.

[45] Shiping Huang, Matthew O. Ward, and Elke A. Rundensteiner. Exploration of dimensionality reduction for text visualization. In *Proc. IEEE Third Intl. Conf. on Coordinated and Multiple Views in Exploratory Visualization*, pages 63–74, July 2005.

[46] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons, NY, USA, 2001.

[47] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, May 2000.

[48] P. Jaccard. The distribution of flora in the alpine zone. *New Phytologist*, 11:37–50, 1912.

[49] R. Kass and P. Vos. *Geometrical Foundations of Asymptotic Inference*. Wiley Series in Probability and Statistics. John Wiley and Sons, NY, USA, 1997.

[50] B. Kégl. Intrinsic dimension estimation using packing numbers. In *Neural Information Processing Systems: NIPS*, Vancouver, CA, Dec. 2002.

[51] Hyunsoo Kim, Peg Howland, and Haesun Park. Dimension reduction in text classification with support vector machines. In *Journal of Machine Learning Research 6*, pages 37–53, January 2005.

[52] J. Kim. *Nonparametric statistical methods for image segmentation and shape analysis*. PhD thesis, Massachusetts Institute of Technology, February 2005.

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

[54] S.N. Lahiri. *Resampling Methods for Dependent Data*. Springer, NY, USA, 2003.

[55] G. Lebanon. Information geometry, the embedding principle, and document classification. In *Proceedings of the 2nd International Symposium on Information Geometry and its Applications*, 2005.

[56] S. Lee, A. Abbott, N. Clark, and P. Araman. Active contours on statistical manifolds and texture segmentation. In *International Conference on Image Processing 2005*, volume 3, pages 828–831, 2005.

[57] S-M. Lee, A. L. Abbott, and P. A. Araman. Dimensionality reduction and clustering on statistical manifolds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–7, June 2007.

[58] E. Levina and P. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Neural Information Processing Systems: NIPS*, Vancouver, CA, Dec. 2004.

[59] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, San Fransisco, 1982.

[60] R. C. Mann. On multiparameter data analysis in flow cytometry. *Cytometry*, 8(2):184–189, 1987.

[61] R. C. Mann, D. M. Popp, and R. E. Hand Jr. The use of projections for dimensionality reduction of flow cytometric data. *Cytometry*, 5(3):304–307, 1984.

[62] R. W. McKenna, L. T. Washington, D. B. Aquino, L. J. Picker, and S. H. Kroft. Immunophenotypic analysis of hematogones (b-lymphocyte precursors) in 662 consecutive bone marrow specimens by 4-color flow cytometry. *Blood*, 98(8):2498–2507, 2001.

[63] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In *Proc. IEEE Neural Networks for Signal Processing Workshop*, 1999.

[64] I. Pardoe, X. Yin, and R. D. Cook. Graphical tools for quadratic discriminant analysis. *Technometrics*, 49(2), May 2007.

[65] A. P. Petland. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:661–674, 1984.

[66] R. Raich, J. A. Costa, S. B. Damelin, and A. O. Hero. Classification constrained dimensionality reduction. *IEEE Transactions on Signal Processing*, 2008. to be submitted.

[67] R. Raich, J. A. Costa, and A. O. Hero. On dimensionality reduction for classification and its applications. In *Proc. IEEE Intl. Conference on Acoustic Speech and Signal Processing*, May 2006.

[68] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proc. of ACM SIGCOMM*, pages 291–302, Sept. 2006.

[69] M. Roederer and R. Hardy. Frequency difference gating: A multivariate method for identifying subsets that differ between samples. *Cytometry*, 45(1):56–64, 2001.

[70] M. Roederer and R. Hardy. Probability binning comparison: A metric for quantitating multivariate distribution differences. *Cytometry*, 45(1):47–55, 2001.

[71] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(1):2323–2326, 2000.

[72] J. Salojarvi, S. Kaski, and J. Sinkkonen. Discriminative clustering in fisher metrics. In *Artificial Neural Networks and Neural Information Processing - Supplementary prodceedings ICANN/ICONIP 2003*, pages 161–164, June 2003.

[73] A. Srivastava, I.H. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, June 2007.

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

[75] George Terrell. The maximal smoothing principle in density estimation. *Journal of the American Statistical Association*, 85(410):470–477, June 1990.

[76] M. Thangavelu and R. Raich. Multiclass linear dimension reduction via a generalized chernoff bound. In *IEEE Machine Learning for Signal Processing Workshop*, Oct. 2008.

[77] Y. Vardi and C.-H Zhang. The multivariate $L1$-median and associated data depth. *Proceedings of the National Academy of Science USA*, 97:1423–1426, 2000.

[78] E. Zamir, B. Geiger, N. Cohen, Z. Kam, and B. Katz. Resolving and classifying haematopoietic bone-marrow cell populations by multi-dimensional analysis of flow-cytometry data. *British Journal of Haematology*, 129:420–431, 2005.

[79] Q. T. Zeng, J. P. Pratt, J. Pak, D. Ravnic, H. Huss, and S. J. Mentzer. Feature-guided clustering of multi-dimensional flow cytometry datasets. *Journal of Biomedical Informatics*, 40:325–331, 2007.

[80] S.K. Zhou and R. Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):917–929, June 2006.