Dimensionality Detection and the Geometric Median on Data Manifolds

by

L. R. Goetz-Weiss

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Applied Mathematics

2017

This thesis entitled:
Dimensionality Detection and the Geometric Median on Data Manifolds
written by L. R. Goetz-Weiss
has been approved for the Department of Applied Mathematics

_____

Prof. François Meyer

_____

Prof. Bengt Fornberg

_____

Prof. Jem Corcoran

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Goetz-Weiss, L. R. (MS, Applied Mathematics)

Dimensionality Detection and the Geometric Median on Data Manifolds

Thesis directed by Prof. François Meyer

In many applications high-dimensional observations are assumed to arrange on or near a low-dimensional manifold embedded in an ambient Euclidean space. In this thesis, ideas from differential geometry are extended to equation-free data analysis to better understand high-dimensional datasets. In particular, two questions are addressed: (1) how can the intrinsic dimensionality of a manifold-valued dataset be determined? and (2) how can this intrinsic dimensionality be leveraged to obtain a better notion of centrality? For (1), two common methods for estimating global dimensionality are stated and a novel approach is proposed to obtain an estimator for local dimensionality. Then, for (2), a novel approach is presented to estimate the geometric median on manifolds of which no prior knowledge of the underlying geometry is known. These methods are first applied to synthetic datasets and then to real world neurological measurements to create a biomarker for the development of epilepsy in an animal model.

Acknowledgements

First and foremost, I would like to thank Prof. François Meyer for his excellent advisement over the past few years. Prof. Meyer consistently encouraged me to form my own ideas while ensuring that I was moving in the correct direction. His mastery of the subject matter and genuine enthusiasm towards his work has been invaluable while completing this thesis. It has been an honor learning from Prof. Meyer. Thank you.

I would next like to thank the Applied Mathematics department. While it is impossible to list all of the department members who have enhanced my time at CU, I would like to particularly extend my gratitude to my committee members, Prof. Bengt Fornberg and Prof. Jem Corcoran.

During my time working on this thesis I was fortunate to receive funding from NSF's EXTREEMS grant as well as the Applied Mathematics department's TA program.

Finally, I would like to thank my friends and family for their continuing support throughout this endeavor.

Contents

Chapter

1 Introduction **1**

2 Differential Geometry Preliminaries **4**

2.1 Manifolds 4

2.2 Tangent Plane 7

2.3 Path Length and Manifold Distance 8

2.4 Conclusion 9

3 Dimensionality Estimation **10**

3.1 Background and Related Work 10

3.2 Multiscale Singular Value Decomposition (MSVD) 11

3.3 Kernelized Correlation Coefficient 12

3.4 Experiments 15

3.5 Going from Global to Local 17

3.6 Conclusion 18

4 The Geometric Median on Data Manifolds **19**

4.1 Introduction 19

4.2 Background and Related Work 20

4.2.1 Geometric p-mean 20

4.2.2 Geometric Median . . . 21
4.2.3 The Weiszfeld Algorithm for Manifolds . . . 21
4.2.4 Moving to Data Manifolds . . . 22
4.3 The Geometric Median on Data Manifolds . . . 22
4.3.1 Approximating the Logarithmic and Exponential Maps . . . 23
4.3.2 Approximating Manifold Distance . . . 24
4.3.3 Determining an Appropriate Initial Guess . . . 28
4.3.4 Adjusting for Finite Sampling . . . 29
4.3.5 An Algorithm for Estimating the Geometric Median on Data Manifolds . . . 29
4.4 Experiments . . . 29
4.5 Conclusion . . . 32

5 Creating a Biomarker for Epileptogenesis **35**
5.1 Background . . . 35
5.1.1 Epilepsy . . . 35
5.1.2 Experiment and Data . . . 35
5.2 Intrinsic Dimensionality . . . 36
5.2.1 Visualizing Dataset . . . 36
5.2.2 Local Dimensionality Estimates . . . 36
5.2.3 Dimensionality as Disease Progresses . . . 40
5.3 Decoding the Progression of Epileptogenesis . . . 40
5.3.1 Biomarker Results . . . 43
5.4 Conclusion . . . 43

6 Conclusion **46**

Bibliography **49**

Appendix

A Notation **52**

Tables

Table

3.1 The synthetic datasets used to compare the MSVD and kernelized correlation coefficient estimators. 16

3.2 The estimated dimensionality of each dataset by the two methods discussed. 16

5.1 The standard deviation and mode prevalence of the dimensionality estimate using random dimension median partitioning with l partitioning steps. 38

5.2 The standard deviation and mode prevalence of the dimensionality estimate using k_D-NN. 40

A.1 Notation. 52

A.2 Notation (cont). 53

Figures

Figure

2.1 An illustration of a smooth d-manifold. 5

2.2 An illustration of the tangent plane of $\mathcal{M}$, $T_p\mathcal{M}$. 7

2.3 The distance in example 2.3. 9

3.1 The singular values associated with a four-dimensional sphere embedded in $\mathbb{R}^{60}$ as a function of scale. The solid line represents the mean over data points and the dashed lines represents one standard deviation above and below the mean. We see that at a ball size of 1 the singular value spectrum does indeed reflect the true dimensionality of the manifold. 13

3.2 The singular values associated with a four-dimensional ellipsoid embedded in $\mathbb{R}^{60}$ as a function of scale. The solid line represents the mean over data points and the dashed lines represents one standard deviation above and below the mean. We see that there is no scale which accurately represents the true dimensionality. 13

3.3 The mean and standard deviation of the dimensionality estimate for the synthetic mouse dataset over 200 trials. 18

4.1 An example of manifold-valued data with a mean that is not manifold-valued. 20

4.2 A comparison of several estimators' sensitivity to sample size. $d = 3$, noise level = 0.05. 33

4.3 A comparison of several estimators' sensitivity to noise. $d = 3$, sample size = 30. . . 33

4.4 A comparison of several estimators' sensitivity to dimension. noise level = 0.05, sample size = 45. 34

5.1 Laplacian Eigenmaps of hAEP data for each condition. Blue - Baseline, Cyan - Silent, Green - Latent, Red - Chronic. 37

5.2 The mean (top) and standard deviation (bottom) of the dimensionality estimate for the hAEP dataset over 100 trials using $l = 2$. 39

5.3 The mean and standard deviation of the local intrinsic dimensionality estimate as the disease progresses over 100 trials using $l = 2$. 41

5.4 Recovered $\tau_i(t)$ for each rat colored by condition (left) and histogram of τ values by condition (right). Cyan - Silent, Green - Latent, Red - Chronic. 44

Chapter 1

Introduction

In recent years we have become surrounded by an abundance of data. One implication of this is the need for new mathematical approaches to visualize, understand, and leverage massive data products. Not surprisingly, these fields of research have exploded in recent years. In general these big data problems belong to one of two categories: first, the data may come from a system for which underlying mathematical relationships are known or second, the data may come from a system which is difficult or impossible to model. For the first situation knowledge of the underlying system can be leveraged to understand the data. In the second case, however, a model must be inferred purely from data and not from known laws describing the system's behavior. The focus of this thesis is the second, equation-free case.

Data from equation-free systems can be roughly divided into two cases. In the first, there is an abundance of observations, each of which consists of a relatively small number of values (i.e. observations are of relatively low dimension). For instance, if a survey is conducted over a large population, data is collected on several thousand people but each participant's response consists of relatively few answers. In this context, statistics can be used to understand the relationships between variables. The second case is where the dimensionality of observations is significantly larger than the total number of observations. An example of this is observations of dynamical systems. Typically when the output of a dynamical system is sampled, thousands of data points are collected for each observation. The goal is then to relate these high-dimensional observations to configurations of the system in its implicit state space. In this second situation it is typically useful

to find a parameterization of the observation space by a set of significantly fewer parameters. Here, the underlying assumption is that observations will arrange on or near a low-dimensional manifold embedded in the ambient Euclidean space.

A classic tool for learning such a parameterization is Principal Component Analysis (PCA). In PCA a set of coordinate axes is found that accounts for as much variance as possible. An observation is then projected onto these principal components to get a low-dimensional representation. This however assumes that the full set of observations may be written as a linear combination of basis vectors which in general is not true. A modern approach is to instead assume that the observations arrange on or around a low-dimensional manifold. In this case, one seeks a nonlinear parameterization of the observation space to recover this underlying manifold. Approaches to this end include Local Linear Embedding (LLE) [34], where linear models are fit to local regions of the observation space and are then patched together to ensure a smooth parameterization, and Laplacian Eigenmaps [7] which approximates the Laplacian operator on the data manifold and uses its eigenvectors as a set of global coordinates.

One challenge of using nonlinear dimensionality reductions is finding the appropriate number of coordinates to use in a global parameterization. In the case of noise-free data observed on a linear subspace of the ambient space, the number of nonzero singular values of the data matrix gives the dimensionality of the subspace [16]. For nonlinear subspaces however, curvature of the space will cause the number of nonzero singular values to overestimate the data's implicit dimensionality. In the noise-free case, this issue can be resolved by running local PCA on small neighborhoods of data, assuming that the sampling density is sufficiently high. The presence of noise however adds an additional difficulty to estimating data dimensionality. Even in the linear case, noise in the data will cause noise in the singular values of the data matrix, requiring a threshold to define when a principal component should be considered noise. In the nonlinear case noise adds a more severe difficultly. In this situation it is crucial to select an appropriate neighborhood size to analyze the data. If the neighborhood is too large, manifold curvature will cause the estimator to overestimate dimensionality and if the neighborhood is too small, noise may cause the estimator

to overestimate the dimensionality. For this reason the most successful estimators have taken a multiscale approach to dimensionality estimation. One final complication is that the data manifold may have different dimensionality in different regions, motivating the use of local estimators. In chapter 3, two common global estimators of dimensionality are discussed and a novel approach to convert a generalized global estimator into a local estimator is presented.

Once an appropriate dimensionality estimate has been reached, a natural goal for time series data is to track the progression of observations over time. In other words, one would like to find a smooth trajectory that minimizes the residual variance. A simple yet common approach to this problem is to look at moving window averages. In the case of high-dimensional observations arranging on or around a low-dimensional manifold, one should be careful in defining the average of a set of points. In particular, the mean taken in the ambient Euclidean space will typically not belong to the manifold. In this context it is more useful to define the *geometric median* as the point which minimizes the total manifold distance to each point. Most work on estimating the geometric median ([11], [15]) has assumed prior knowledge of manifold geometry, which is typically not available in equation-free models. In chapter 4, a novel extension of prior work on estimating the geometric median to data manifolds is presented.

The methods discussed in this thesis are largely motivated by an interdisciplinary effort to construct a computational biomarker for the development of epilepsy following a traumatic shock (epileptogenesis) [27]. In this project, an animal model is injected with a drug known to induce epilepsy. As the animal progresses through the stages of epileptogenesis, a measurement of auditory evoked potential in the animal's hippocampus is recorded. These hippocampal auditory evoked potentials (hAEPs) serve as a measurement of the rats current status in epileptogenesis. In chapter 5, the methods from chapters 3 and 4 are used to construct a computational biomarker that tracks the progression of epileptogenesis based solely on these measurements.

Chapter 2

Differential Geometry Preliminaries

In this chapter relevant concepts from differential geometry are formalized. In general, differential geometry applies methods from calculus and linear algebra to study problems in geometry, particularly understanding smooth manifolds (defined below). Unlike differential topology, which is concerned with spaces up to a homeomorphism, differential geometry is concerned with spaces up to an isometry. In other words, differential topology deals with the structure of a space, whereas differential geometry also deals with metrics associated with the space.

Differential geometry is a useful tool for a wide variety of disciplines ranging from general relativity, to stochastic processes, computer vision, and econometrics. For the purpose of this thesis we will focus on a specific subset of differential geometry that is relevant to our application for high-dimensional observations. In particular, we will only consider extrinsic geometries, in other words geometric structures that are embedded in a high but finite-dimensional Euclidean space. We focus here because most high-dimensional observations can be interpreted as samples in such a Euclidean space. The definitions and examples in this chapter are taken from [33].

2.1 Manifolds

Differential geometry is largely concerned with understanding smooth manifolds, so defining such an object is perhaps the most logical place to begin.

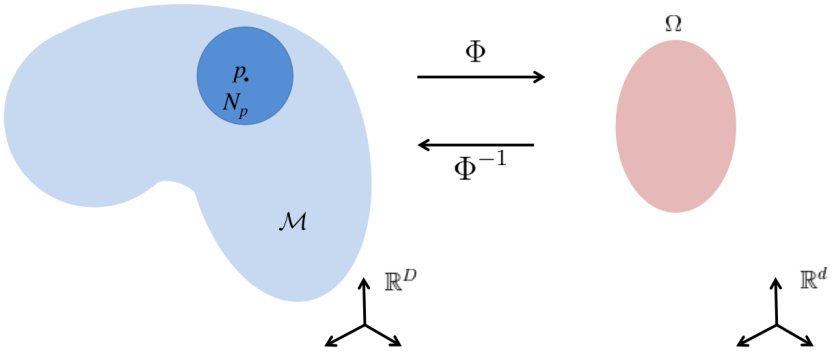

Figure 2.1: An illustration of a smooth d-manifold.

Definition 2.1 *Let D and d be positive integers. A subset $\mathcal{M} \subset \mathbb{R}^D$ is called a* **smooth d-submanifold of $\mathbb{R}^D$** *if for each $p \in \mathcal{M}$, there exists an open neighborhood $N_p \subset \mathbb{R}^D$ such that there is an diffeomorphism (i.e. a smooth invertible bijective map) $\Phi : N_p \cap \mathcal{M} \to \Omega$ where Ω is an open subset of $\mathbb{R}^d$. Φ is called a* **coordinate chart** *of $\mathcal{M}$, while its inverse $\Phi^{-1} : \Omega \to N_p \cap \mathcal{M}$ is called a* **parametrization** *of $N_p \cap \mathcal{M}$. The collection of all of the coordinate charts of $\mathcal{M}$ is called the* **altas** *of $\mathcal{M}$ and d is called the* **dimensionality** *of $\mathcal{M}$.*

Example 2.1.1 Consider the unit circle $S_1 = \{x \in \mathbb{R}^2 \mid \|x\|_2 = 1\}$ where $\|x\|_2^2 := \sum_i x_i^2$. For any $p \in S_1$ an open neighborhood of p can be mapped to an open interval on the real line. In particular, let $\theta \in [0, 2\pi)$ be the angle between p and the horizontal axis. Then, the set $S_1 \cap N_p = \{(\cos(\eta), \sin(\eta)) \mid \eta \in (\theta - \frac{\pi}{2}, \theta + \frac{\pi}{2})\}$ is diffeomorphic to the open interval $(0, \pi)$. Thus the unit circle is a smooth 1-manifold.

Example 2.1.2 The unit circle is a special case of the n-sphere, $S_{n-1} = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$. S_n also satisfies the definition of a smooth manifold by considering the map

$$\begin{aligned} \Phi : \{x \in S_{n-1} \mid x_n > 0\} &\to \mathbb{R}^{n-1} \\ \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} &\mapsto \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}, \end{aligned} \tag{2.1}$$

and its inverse,

$$\Phi^{-1} : \mathbb{R}^{n-1} \to \{x \in S_{n-1} \mid x_n > 0\}$$
$$\begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ \sqrt{1 - \sum_{i=1}^{n-1} x_i^2} \end{bmatrix}. \tag{2.2}$$

This map satisfies the coordinate chart definition for any $p \in \{x \in S_{n-1} \mid x_n > 0\}$. By defining similar maps for $x_1 > 0, x_1 < 0, \ \ldots \ , x_{n-1} > 0, x_{n-1} < 0$, and $x_n < 0$ we see that any $p \in S_{n-1}$ has a neighborhood for which there exists a diffeomorphism to $\mathbb{R}^{n-1}$. Thus S_{n-1} is a smooth $(n-1)$-manifold.

Example 2.1.3 Consider the space of special orthogonal 2×2 matrices, $SO(2) = \{M \in \mathbb{R}^{2\times 2} \mid M^T M = M M^T = I \ , \ det(M) = 1\}$. Any element of $SO(2)$ can be written as

$$M = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & cos(\theta) \end{bmatrix}. \tag{2.3}$$

From this parameterization one can construct a diffeomorphism from $SO(2)$ to the unit circle as

$$\Phi : SO(2) \to S_1$$
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \mapsto \begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix}, \tag{2.4}$$

which has the inverse

$$\Phi^{-1} : S_1 \to SO(2)$$
$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \mapsto \begin{bmatrix} b_1 & b_2 \\ -b_2 & b_1 \end{bmatrix}. \tag{2.5}$$

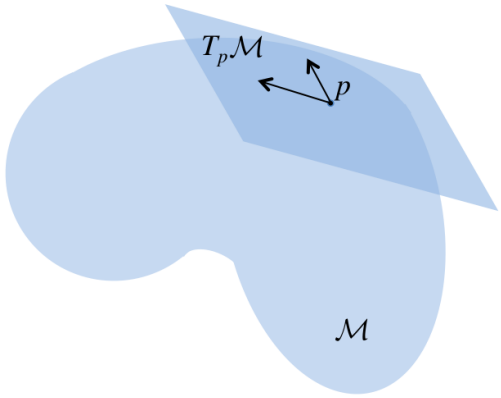

Figure 2.2: An illustration of the tangent plane of $\mathcal{M}$, $T_p\mathcal{M}$.

Then, by composing this map with the map from S_1 to $\mathbb{R}$, one sees that $SO(2)$ is a smooth 1-manifold.

2.2 Tangent Plane

When dealing with a smooth manifold $\mathcal{M}$ it is often useful to understand the local geometry of $\mathcal{M}$ around a point $p \in \mathcal{M}$. To formalize this notion we define the tangent plane of $\mathcal{M}$ at p as follows.

Definition 2.2 *Let $\mathcal{M} \subset \mathbb{R}^D$ be a smooth d-manifold and let $p \in \mathcal{M}$. A vector $v \in \mathbb{R}^D$ is called a* **tangent vector** *of $\mathcal{M}$ at p if there exists a smooth curve $\gamma : [0,1] \to \mathcal{M}$ such that $\gamma(0) = p$ and $\dot{\gamma}(0) = v$. The set of all tangent vectors at p is called the* **tangent space** *of $\mathcal{M}$ at p and denoted by $T_p\mathcal{M}$. In other words,*

$$T_p\mathcal{M} = \{\dot{\gamma}(0) \mid \gamma : [0,1] \to \mathcal{M} \text{ is smooth}, \gamma(0) = p\}. \tag{2.6}$$

One should note that since there is a diffeomorphism from $N_p \cap \mathcal{M}$ to $\mathbb{R}^d$, it follows that $T_p\mathcal{M}$ is a d-dimensional space. Further, $T_p\mathcal{M}$ is an affine (linear) subspace of $\mathbb{R}^D$ (the reader is referred to [33] for a proof of this).

Example 2.2 Consider the tangent plane T_pS_{n-1} of S_{n-1} at a point $p \in S_{n-1}$. Since

$p \in S_{n-1}$, $p^T p = 1$. Further, for any path $\gamma(t) \subset S_{n-1}$, $\|\gamma(t)\|^2 = 1$ so $\frac{d}{dt}\|\gamma(t)\|^2 = 0$. That is, $2\dot{\gamma}(t)^T\gamma(t) = 0$. Since $\gamma(0) = p$, it follows that $\dot{\gamma}(0)^T p = 0$. In other words,

$$T_p S_{n-1} = \{x \in \mathbb{R}^n \mid x^T p = 0\}. \tag{2.7}$$

2.3 Path Length and Manifold Distance

The final tool that we need from differential geometry is a notion of distance. First, let us recall the length of a path $\gamma : [0,1] \to \mathbb{R}^D$, $L(\gamma)$,

$$L(\gamma) := \int_0^1 \|\dot{\gamma}(t)\| dt. \tag{2.8}$$

It is important to note that this definition does not depend on the parameterization of γ but rather the image $\gamma([0,1])$. To see this, consider a reparameterization of γ, $\hat{\gamma}(t) = \gamma(s(t))$ where $s : [0,1] \to [0,1]$, $s(0) = 0$, $s(1) = 1$, and $\dot{s} \geq 0$. Then

$$\begin{aligned} L(\hat{\gamma}) &= \int_0^1 \|\frac{d}{dt}\gamma(s(t))\| dt \\ &= \int_0^1 \|\dot{\gamma}(s(t))\|\dot{s}(t) dt. \end{aligned} \tag{2.9}$$

By performing the change of variable $u = s(t)$, we see from the second integral that $L(\hat{\gamma}) = L(\gamma)$.

We can now define a notion of distance on a smooth manifold $\mathcal{M}$ based on the shortest path between points.

Definition 2.3 *Let $p, q \in \mathcal{M}$. The* **manifold distance** *between p and q is defined as*

$$d(p,q) = \inf_{\gamma:[0,1]\to\mathcal{M}, \gamma(0)=p, \gamma(1)=q} L(\gamma). \tag{2.10}$$

The reader should note that $L(\gamma) \geq \|\gamma(1) - \gamma(0)\|$, so it follows that $d(p,q) \leq \|p - q\|$. Further, one can show that this distance forms a metric on $\mathcal{M}$, i.e.

(i) *If $p, q \in \mathcal{M}$ and $d(p,q) = 0$, then $p = q$*

(ii) *For all $p, q \in \mathcal{M}$, $d(p,q) = d(q,p)$*

(iii) *For all $p, q, r \in \mathcal{M}$, $d(p,q) \leq d(p,r) + d(r,q)$*

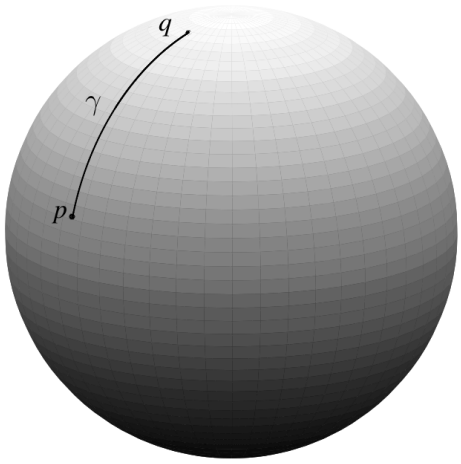

Figure 2.3: The distance in example 2.3.

For a proof of this result, the reader is referred to [33].

Example 2.3 On the 2-sphere, $S_2 \subset \mathbb{R}^3$, the distance between two points $p, q \in S_2$ is the length of the shortest path between p and q on S_2, i.e. an arc on a great circle between p and q (figure 2.3). The distance on S_2 is thus $d(p,q) = \cos^{-1}(\langle p, q \rangle)$ where $\langle \cdot, \cdot \rangle$ is the dot product.

2.4 Conclusion

In this chapter we have formalized essential ideas from differential geometry. Specifically, we have defined a smooth d-manifold, $\mathcal{M} \subset \mathbb{R}^D$, the tangent plane at any point $p \in \mathcal{M}$, and the distance metric on $\mathcal{M}$. In the following chapters these ideas are applied to equation-free models. In particular, given a set of manifold-valued data, we discuss methods for estimating dimensionality, the tangent plane and the distance metric and demonstrate that these estimates can be used for creating equation-free models based solely on the underlying geometric structure of a dataset.

Chapter 3

Dimensionality Estimation

3.1 Background and Related Work

In many cases high-dimensional observations exhibit a dramatically lower-dimensional structure. In other words, the observations lie on or around a low-dimensional manifold embedded in a high-dimensional ambient space. Consider images of a ball rolling across a table. Every pixel in the image corresponds to a distinct coordinate, so each image can be represented as a square $n \times n$ matrix. Therefore, the space of all possible images has n^2 dimensions. The images however depend on at most five parameters describing the ball's location and orientation in three-dimensional space, so the images will indeed concentrate around a five-dimensional manifold embedded in $\mathbb{R}^{n^2}$.

In the case of a linear manifold, i.e. a manifold closed under addition and scalar multiplication, the underlying parameters can be recovered using Principal Component Analysis (PCA). In most cases however (e.g. the images of a ball rolling across a table) the underlying manifold is nonlinear. There has been significant work on recovering a parameterization of nonlinear manifolds [35], [34], [7], [39] but each of these approaches assumes that the submanifold's dimension is know *a priori*. Estimating the intrinsic dimensionality of a dataset is an active field of research [26], [19], [24]. Some datasets also exhibit different dimensionality in different regions of the ambient space [12] and in this case an estimator for the local intrinsic dimensionality is desirable.

The goal of this chapter is to describe two modern approaches for estimating the global dimensionality and to then propose a novel approach to estimate the local dimensionality of a point cloud. The first estimator, which is motivated by linear algebra, is based on applying the Singular

Value Decomposition (SVD) on neighborhoods of varying size [26]. The second estimator, which is inspired by geometry, is based on the number of points inside the ball $B_r(p) = \{x \in \mathbb{R}^D \mid \|x-p\| < r\}$ as r varies [19]. These two methods are explained in the next two sections respectively. In the fourth section, experiments are performed on synthetic data that demonstrate the strengths and weaknesses of the two methods. Finally, we describe a novel approach to estimating the local dimensionality is presented.

3.2 Multiscale Singular Value Decomposition (MSVD)

Fukunaga [16] first proposed using Principle Component Analysis to determine the intrinsic dimensionality of linear subspaces embedded in a high-dimensional space. Let $X = [x_1 \ ... \ x_N]$ be the D by N data matrix with columns corresponding to data points. If this data lies on a $d < D$ dimensional linear subspace of $\mathbb{R}^D$, then only the first d singular values of X will be non-zero. In practice, data is usually obscured by some amount of high-dimensional noise, so instead of looking for nonzero singular values, we look for singular values above a certain threshold.

In the case where the data lies on a nonlinear manifold, the magnitude of the singular values will be influenced by manifold curvature. Consider for example S_2, the unit sphere in $\mathbb{R}^3$. As discussed earlier, this is a two-dimensional manifold. Indeed, if you look at the sphere on a small enough scale it will look like a two-dimensional plane and only the first two singular values will be non-zero. However, as the scale increases curvature will cause the third singular value to become non-zero, making the manifold appear three-dimensional. In the case of a noisy sampling from the manifold, the Singular Value Decomposition may also mistake noise in the ambient space as an additional dimension. We thus need to consider the data at a scale large enough such that the singular values from noise may be differentiated from the singular values from the manifold's true geometry, but small enough such that manifold curvature does not cause us to overestimate the intrinsic dimensionality.

Motivated by this thought, the authors in [26] suggests a multiscale SVD approach (MSVD) where one computes the singular values of neighborhoods of varying size and then selects an

appropriate scale to determine the intrinsic dimensionality. The multiscale SVD method first finds for each x_i, $i = 1, ..., N$ and for each scale $r = r_{min}, ..., r_{max}$ the neighborhood given by $N_r(x_i) = \{x_j \mid x_j \in B_r(x_i)\}$ where $B_r(z)$ is a ball of radius r centered at z. We then denote $\sigma_k^{i,r}$ as the k^{th} singular value of the data matrix corresponding to the data points in $N_r(x_i)$ sorted in decreasing order, $\sigma_k^{i,r} \geq \sigma_{k+1}^{i,r}$.

As an example, consider the unit sphere $S_4 = \{x \in \mathbb{R}^5 \mid \sum_{i=1}^5 x_i^2 = 1\}$ embedded in $\mathbb{R}^{60}$ and obscured by 60-dimensional noise. Note that after this embedding, samples still arrange around a four-dimensional manifold embedded in $\mathbb{R}^{60}$. Figure 3.1 shows the singular value spectrum as a function of the ball size r. We indeed see that at a small scale there is not a significant gap in singular values while at a large scale the largest gap is between the fifth and sixth eigenvalue, suggesting that the manifold is five dimensional. Between these cases however, there is a scale (ball size of roughly 1) where the spectrum does indeed reflect the true intrinsic dimensionality of four.

Unfortunately, this method falls apart with a slight deviation from this ideal experiment. Consider replacing the sphere in the previous experiment with the ellipsoid $\{x \in \mathbb{R}^5 | \sum_{i=1}^5 x_i^2/i = 1\}$. Figure 3.2 shows the singular value spectrum of neighborhoods as a function of the ball size r. In this example there is no scale which correctly captures the intrinsic dimensionality. Even considering the rate at which the singular values grow as suggested in [26], there is no clear distinction between singular values corresponding to the true geometry and those corresponding to noise. Indeed, applying the algorithm described in [26] estimates the intrinsic dimensionality to be 1.6, significantly underestimating the true dimensionality of four. It seems that although this method can handle noisy samples quite well, it suffers in the case of anisotropic curvature.

3.3 Kernelized Correlation Coefficient

The second estimator considered is motivated by the behavior of the number of points within a ball of varying radius. Specifically, one expects the number of points in a ball of size r centered at x_i to grow as r^d where d is the true dimensionality of the manifold. We define the *sample*

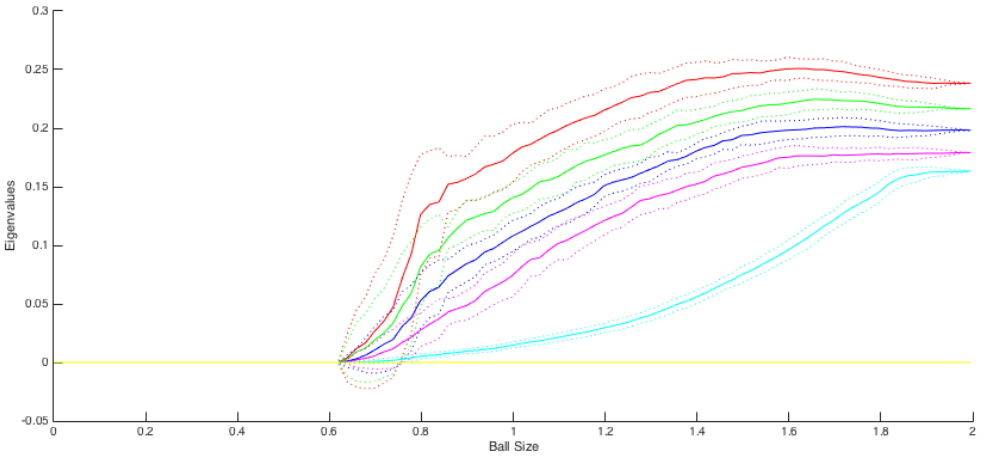

Figure 3.1: The singular values associated with a four-dimensional sphere embedded in $\mathbb{R}^{60}$ as a function of scale. The solid line represents the mean over data points and the dashed lines represents one standard deviation above and below the mean. We see that at a ball size of 1 the singular value spectrum does indeed reflect the true dimensionality of the manifold.

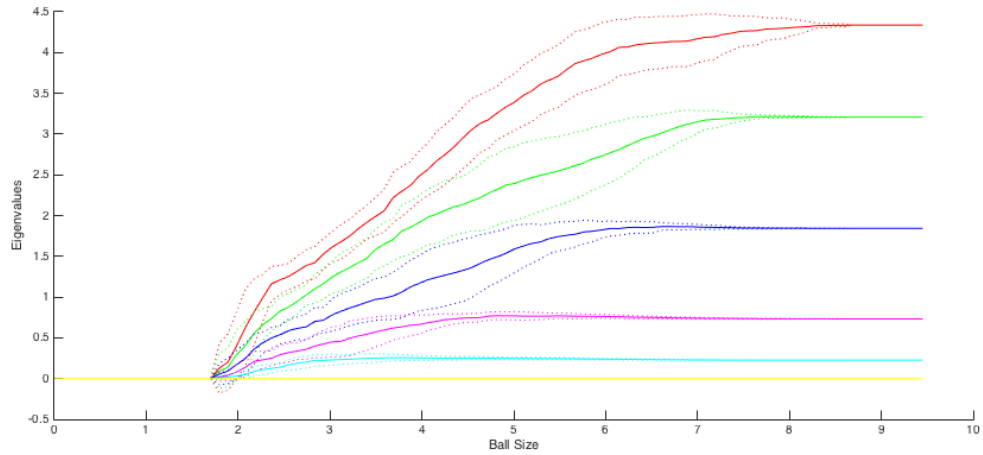

Figure 3.2: The singular values associated with a four-dimensional ellipsoid embedded in $\mathbb{R}^{60}$ as a function of scale. The solid line represents the mean over data points and the dashed lines represents one standard deviation above and below the mean. We see that there is no scale which accurately represents the true dimensionality.

correlation sum as

$$C_n(s) = \frac{2}{n(n-1)} \sum_{i<j}^{n} \mathbb{I}(||x_i - x_j|| < s), \tag{3.1}$$

where n is number of data points in the dataset and $\mathbb{I}$ is the indicator function. Noting that $\frac{2}{n(n-1)} = \binom{n}{2}^{-1}$, $C_n(s)$ has the simple interpretation as the fraction of points with a pairwise distance less than s.

We can then approximate the dimensionality of the dataset as

$$d \approx \nu := \lim_{s \to 0} \lim_{n \to \infty} \frac{\log C_n(s)}{\log s}. \tag{3.2}$$

Since of course we cannot let $s \to 0$ and $n \to \infty$, we instead compute ν for varying values of s and fit a line to estimate the behavior of $C_n(s)$ as $s \to 0$.

Instead of using the correlation method directly, we apply the kernel version proposed in [19]. The authors replace the correlation sum with the U-statistic

$$U_{n,h}(k) = \frac{2}{n(n-1)} \sum_{i<j}^{n} k_h(||x_i - x_j||^2), \tag{3.3}$$

where

$$k_h(z) = \frac{1}{h^l} k(z/h^2). \tag{3.4}$$

For some kernel k. After making several assumption on the probability density function describing the data, its supporting manifold, and k, the authors in [19] show that if $h \to 0$ and $nh^l \to \infty$ then $\lim_{n\to\infty} U_{n,h}(k)$ converges if and only if $l = d$, where d is the true dimension of the manifold.

As discussed in [19] we would like $h_l(n)$ to approach zero at the fastest possible rate while meeting the constraints that $nh^l \to \infty$ so we fix $h_l(n)$ as a function of dimension and sample size,

$$nh_l(n)^l = \frac{1}{c^l} \log n \Rightarrow h_l(n) = \frac{1}{c} \left(\frac{\log n}{n} \right)^{1/l}, \tag{3.5}$$

where c is a constant. We choose c in such a way that when looking at the full dataset of N points we use the same scale independent of dimension. Specifically, we set $h_l(N)$ to the mean distance

of a point to its nearest neighbor,

$$h_l(N) = \frac{1}{N}\sum_{i=1}^{N} \min_{j \neq i} ||x_i - x_j||. \tag{3.6}$$

Finally we have

$$h_l(n) = h_l(N)\left(\frac{N}{n}\frac{\log n}{\log N}\right)^{1/l}. \tag{3.7}$$

Note that this function evaluated at $n = N$ has no dependency on l.

For each dimension $l = 1, ..., l_{max}$ the authors divides the data into r partitions $r = 1, ..., 5$ and compute $U_{\lfloor N/r \rfloor, h_l(\lfloor N/r \rfloor)}(k)$. To make the algorithm more robust, they look not only at the U-statistic of subsamples individually but also at the U-statistic between subsamples. The two-sample U-statistic is defined as

$$U_{n,h}(X, Y) = \frac{1}{n^2}\sum_{i,j=1}^{n} k_h(||x_i - y_j||). \tag{3.8}$$

The U-statistic is then averaged for all $r(r+1)/2$ combinations of the r subsamples and a line is fit through the points $\left[\log h_l(\lfloor N/r \rfloor), U_{\lfloor N/r \rfloor, h_l(\lfloor N/r \rfloor)}(k)\right]$ using weighted least squares with weights $1/r$. The analysis in [19] shows that, under certain assumptions, the slope of $\log U_{n,h_l(n)}(k)$ is given by $(d-l)\log h_l(n)$ as $n \to \infty$ and $h \to 0$. The dimensionality is thus estimated by the line with the smallest slope in absolute value.

3.4 Experiments

Estimating the intrinsic dimensionality of a dataset is made difficult by two main factors: noise and curvature. To study the effect of these two factors on the performance of the two methods described, we construct the datasets shown in table 3.1 and apply both estimators. In each dataset, the points are embedded in $\mathbb{R}^{60}$. The noise in datasets 2 and 4 is normally distributed with standard deviation 0.2. Table 3.2 shows the estimated dimensionality of each of these datasets by both methods.

This experiment demonstrates the strengths and weaknesses of the two methods. First, as discussed previously, we notice that in the presence of anisotropic curvature, the MSVD method begins to underestimate the intrinsic dimensionality. Next, we note that the presence of noise

Dataset	Description
1	Noiseless samples drawn from a 4-dimensional unit sphere
2	Noisy samples drawn from a 4-dimensional unit sphere
3	Noiseless samples drawn from the 4-dimensional ellipsoid defined by $\sum_{i=1}^{5} x_i^2/i = 1$
4	Noisy samples drawn from the 4-dimensional ellipsoid defined by $\sum_{i=1}^{5} x_i^2/i = 1$

Table 3.1: The synthetic datasets used to compare the MSVD and kernelized correlation coefficient estimators.

Dataset	Manifold	Noise	MSVD	Ustat
1	Sphere	0	4	4
2	Sphere	0.2	4	10
3	Ellipsoid	0	2	5
4	Ellipsoid	0.2	1.6	9

Table 3.2: The estimated dimensionality of each dataset by the two methods discussed.

causes the kernelized correlation coefficient to severely overestimate the intrinsic dimension. The authors of [19] do not claim that their method generalizes to noisily sampled manifold and mentions that noisy data can be considered to be sampled from a highly curved manifold.

3.5 Going from Global to Local

In general a manifold could be of different dimensionality in different regions. For instance, consider the synthetic "mouse" dataset shown in figure 3.3. Here the tail is of dimension 1 while the body is of dimension 2. In this case we would like an estimator for the *local* manifold dimensionality.

Carter [12] suggests using a k nearest neighbor (kNN) approach to determine local dimensionality. For each point they determine the k nearest neighbors and run an intrinsic dimensionality estimator on the neighborhood. We however use a different approach for turning a generalized global estimator into a local estimator based on the medians of randomly selected dimensions. In chapter 5 we will see that applying this randomized method on real data leads to a more locally consistent estimator. The approach is summarized in algorithm 1

for $i = 1, \ldots, EPOCHS$ **do**
  for $j = 1, \ldots, l$ **do**
    Randomly select a dimension
    Use this dimension's median to divide the dataset into two equal-size subsets
  end
  Run the global estimate on each of the 2^l subsets
  Assign each point in each subset a dimension based on this global estimate
end
Assign a local dimensionality estimate to each point from the average over epochs

Algorithm 1: Partitioning by the median of a random dimension to turn a generalized global dimensionality estimator into a local one.

As a proof of concept consider the following example. Let S_2 be the unit sphere in $\mathbb{R}^3$ (recall that S_2 is a 2-dimensional manifold) and let $T = \{x = [t \ \sin(3(t-1)) \ 0] \mid t \in (1, 5)\}$. Now consider the dataset defined as $S_3 \cup T$ rotated by some random orthogonal matrix R. This rotation is only necessary because of issues that arise when partitioning the dataset by the median of x_3. Specifically, since all of the points in T have $x_3 = 0$, the median is poorly defined.

Figure 3.3 shows the mean and standard deviation of the estimated dimensionality for each

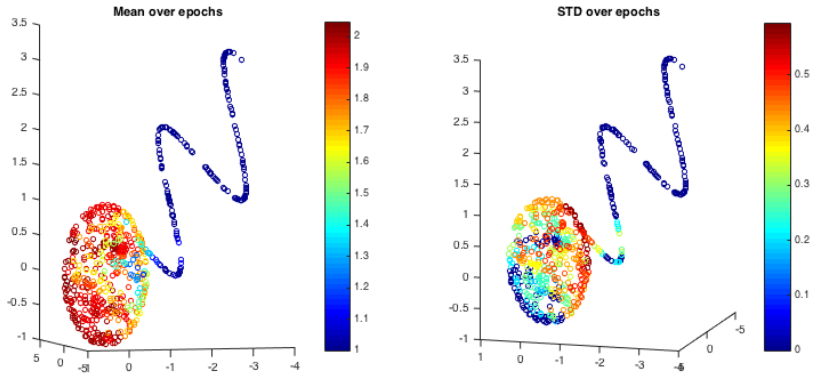

Figure 3.3: The mean and standard deviation of the dimensionality estimate for the synthetic mouse dataset over 200 trials.

point over 200 trails. The dataset here has 500 samples from S_3 and 200 from T. $l = 3$ is used. We see indeed that points sampled from S_3 are estimated to have dimension 2 and points sampled from T are estimated to have dimension 1. Further, we see that points on the boundary have dimension between 1 and 2, and a high standard deviation. This demonstrates the limitations of our method near the boundary between regions of different dimensionality.

3.6 Conclusion

In this chapter we have developed multiscale SVD and the kernelized correlation coefficient, two methods to estimate the intrinsic dimensionality of a set of observations. These estimators were compared through a controlled experiment on synthetic data and their respective strengths and weaknesses were discussed. In particular, we found that while MSVD handles noise well the quality of the estimate suffers in the presence of anisotropic curvature. The kernelized correlation coefficient estimate on the other hand was shown to handle anisotropic curvature, but not noise. A method to make a local dimensionality estimator from a generalized global estimator was then presented and finally, this local dimensionality estimator was demonstrated on a synthetic dataset.

Chapter 4

The Geometric Median on Data Manifolds

4.1 Introduction

In this chapter an estimator for the geometric median of a manifold-valued dataset is developed. For a set of points sampled from a manifold $\mathcal{M}$, the mean computed in the ambient Euclidean space may no longer belong to $\mathcal{M}$. As an example, consider a set of orthogonal $n \times n$ matrices. Although each matrix in this set is orthogonal, the mean of these matrices is typically non-orthogonal. A more reliable notion of centrality would require the mean to be an element of the underlying manifold. The common approach here is to define the geometric p-mean on the manifold as the point which minimizes a manifold distance based cost function. Several methods have been proposed to estimate the geometric p-mean based on closed form knowledge of the manifold logarithm and exponential maps. In equation-free models however, no knowledge of these maps is known.

In what follows we extend recent work in estimating the geometric median [15] to manifolds implicitly defined by a set of observations. This work focuses on the geometric median (i.e. the geometric 1-mean) due to the median's low sensitivity to outliers [15]. The result is an algorithm that uses the geometry of a dataset to create a meaningful notion of centrality for high-dimensional data with low intrinsic dimensionality. In the next section the geometric median is defined and a method for estimating it is described. In section 4.2, this method is extended to data manifolds and in section 4.3, the approach is demonstrated on synthetic datasets.

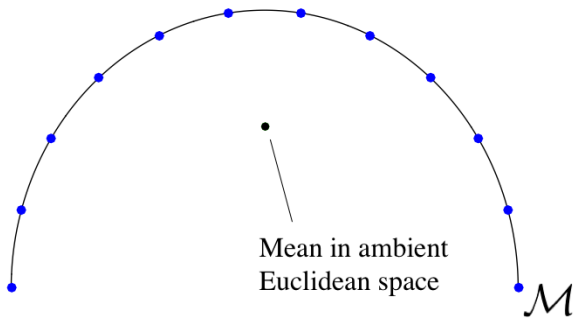

Figure 4.1: An example of manifold-valued data with a mean that is not manifold-valued.

4.2 Background and Related Work

4.2.1 Geometric p-mean

In what follows let $x_i \in \mathcal{M}$, $i = 1, ..., n$ denote a set of observations where $\mathcal{M} \subset \mathbb{R}^D$ is a d-dimensional submanifold equipped with a distance metric $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$. The p-mean is defined as the minimizer of the cost function

$$C_p(m) = \sum_{i=1}^{n} w_i \left(d(m, x_i)\right)^p \tag{4.1}$$

where w_i is a set of weights with $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$.

In the Euclidean case $\mathcal{M} = \mathbb{R}^d$, the p-mean can be found in closed form for $p \in [2, \infty)$ while for $p = 1$ no such closed form exists. Nonetheless, there are gradient descent methods that have been shown to quickly converge to the 1-mean (i.e. the median) [41], [22], [31]. In the non-Euclidean case however there are typically no closed form expressions of the p-mean for any p. Recently, there has been significant progress on iterative methods to find the geometric p-mean. The authors in [4] derive a stochastic gradient decent algorithm and shows almost sure convergence. In [2] a deterministic approach is taken and conditions for convergence are given based on manifold properties and step size. The authors in [18] take a slightly different approach and views the problem as finding the zero of a vector field on the manifold. In [17], [6], [38], and [23] the special cases of

positive definite matrices, the circle, finite-dimensional Lie groups, and Kendal shape spaces are considered, respectively. Applications include animation [32], and shape analysis [9]. The authors in [9] also discuss the benefit of extrinsic analysis over intrinsic analysis.

4.2.2 Geometric Median

In Euclidean spaces it is well known that the sample mean (in our context the 2-mean) is sensitive to outliers. In [15] this problem is discussed extensively and the authors derive an algorithm to find the geometric median (1-mean). Due to the robustness of the median to outliers, the current work will focus on the $p = 1$ case.

In a Euclidean space, $\mathcal{M} = \mathbb{R}^d$, C_1 can easily be shown to be convex by the convexity of the distance metric. However, for the more general Riemannian case we cannot guarantee the convexity of C_1. If we assume that the x_i lie in a convex subset $U \subset \mathcal{M}$, then there is a unique shortest-path geodesic from x_i to x_j for each i, j and the notion of manifold distance is well defined. Under this assumption, the authors in [15] show the following result:

Theorem 4.2:

If the sectional curvatures of $\mathcal{M}$ are non positive or are bounded above by $\Delta > 0$ and $diam(U) < \pi/(2\sqrt{\Delta})$, then $C_1(x)$ has a unique minimizer.

4.2.3 The Weiszfeld Algorithm for Manifolds

In the Euclidean case the algorithm first proposed by Weiszfeld [41] (later improved by [22], [31]) can be used to find the median. First consider the gradient of the cost function,

$$\nabla C_1(m) = \nabla \left(\sum_{i=1}^{n} w_i ||x_i - m|| \right) = \sum_{i=1}^{n} \frac{w_i x_i}{||m - x_i||}. \tag{4.2}$$

The iteration presented by Ostresh [31] is a scaled gradient descent method,

$$m^{(k+1)} = m^{(k)} - \alpha \left(\sum_{i=1}^{n} \frac{w_i x_i}{||m^{(k)} - x_i||} \right) \left(\sum_{i=1}^{n} \frac{w_i}{||m^{(k)} - x_i||} \right)^{-1}. \tag{4.3}$$

Ostresh showed that this iteration will converge when $0 \leq \alpha \leq 2$ and the points are not colinear.

In the general Riemannian case the gradient of the cost function is

$$\nabla C_1(m) = \nabla \left(\sum_{i=1}^{n} w_i d(x_i, m) \right) = - \sum_{i=1}^{n} \frac{w_i \mathrm{Log}_m(x_i)}{d(x_i, m)}. \tag{4.4}$$

Fletcher [15] presents a gradient descent iteration for finding the median on a manifold,

$$m^{(k+1)} = \mathrm{Exp}_{m^{(k)}}(\alpha \nu_k) \ ; \ \nu_k = \left(\sum_{i=1}^{n} \frac{w_i \mathrm{Log}_m(x_i)}{d(x_i, m)} \right) \left(\sum_{i=1}^{n} \frac{w_i}{d(x_i, m)} \right)^{-1}. \tag{4.5}$$

Fletcher showed that this iteration converges to the unique geometric median if $0 \leq \alpha \leq 2$ and the conditions of theorem 4.2 are met.

4.2.4 Moving to Data Manifolds

In all prior work on estimating the geometric median on Riemannian manifolds, either explicit knowledge of the logarithmic and exponential maps is assumed or an approximation based on application specific assumptions is used. In data-driven, equation-free models, however, the logarithmic and exponential map are typically unknown. Nonetheless, a notion of centrality is useful in a variety of applications. In what follows the algorithm given by (4.5) is extended to manifolds implicitly defined by a set of samples. The result is a method for finding the geometric median for an arbitrary set of points with no closed form expressions describing the underlying manifold.

4.3 The Geometric Median on Data Manifolds

To apply the algorithm defined in (4.5) four issues must be addressed. First, an appropriate approximation for the logarithmic and exponential maps must be defined. Second, a reliable estimate of the manifold distance function $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ must be found. Third, an appropriate initial guess for the median must be determined, and fourth, the effect of finite sampling of the manifold must be considered. These four issues are the topics of the subsequent sections.

4.3.1 Approximating the Logarithmic and Exponential Maps

Inferring the geometry of a manifold $\mathcal{M}$ from a finite set of samples is challenging for three main reasons. First, separating the structure of $\mathcal{M}$ from noise in the ambient space is difficult when the sampling density is low. Indeed, there is a fundamental principal that limits how large curvature and noise can become before the underlying local linear structure is no longer recoverable [21]. Second, an appropriate scale must be chosen to recover the local linear structure and third, the deviation from this local linear structure, i.e. the curvature of $\mathcal{M}$, must be understood. For the last issue, methods in multivariate interpolation have proven to be useful for approximating the nonlinear structure of $\mathcal{M}$ [29].

In what follows, let $\mathcal{M}$ be a d-dimensional manifold embedded in $\mathbb{R}^D$. At each point $x \in \mathcal{M}$ let $T_x\mathcal{M}$ denote the tangent plane of $\mathcal{M}$ at x. The logarithm at x is the projection of a neighborhood of x onto $T_x\mathcal{M}$,

$$\text{Log}_x : N(x) \cap \mathcal{M} \to T_x\mathcal{M}. \tag{4.6}$$

When $\mathcal{M}$ is embedded in $\mathbb{R}^D$ the tangent plane and thus Log_x can be approximated by applying Principal Component Analysis to a neighborhood of points centered on x. There are two common methods for selecting the neighborhood $N(x)$: the epsilon ball,

$$N_\epsilon(x) = \{y \in \mathbb{R}^D \mid ||y - x||_2 < \epsilon\}, \tag{4.7}$$

and by taking the k-nearest neighbors (kNN) of x,

$$N_k(x) = \arg \min_{y_1,\ldots,y_k} \sum_{i=1}^{k} ||x - y_i||_2. \tag{4.8}$$

In the current work we focus on $N_k(x)$ as it is typically more robust to nonuniform sampling. The d coordinates of $T_x\mathcal{M}$ are then given by the first d left singular vectors of the mean centered data matrix $X_{N(x)} = [x_{i_1} \ \ldots \ x_{i_k}] \in \mathbb{R}^{D\times k}$ where $i_1, \ldots, i_k$ are the indices of the data points in $N_k(x)$. $X_{N(x)}$ is centered such that each row has mean zero.

The exponential map, $\text{Exp}_x : T_x\mathcal{M} \to N(x) \cap \mathcal{M}$ is defined as the inverse of Log_x. Since the projection given by PCA is not invertible, an approximate exponential map is not directly available.

Indeed, to compute the exponential map knowledge of the geometry of $\mathcal{M}$ is typically needed. In the equation-free case, however, the local geometry of $\mathcal{M}$ may be approximated by interpolating the known values of the exponential map at each sample point. If we let $x'_1, ..., x'_k$ denote the set of data points within $N(x)$ (i.e. $x'_j = x_{i_j}$) and denote $y'_i = \text{Log}_x(x'_i)$ then we have $x'_i = \text{Exp}_x y'_i$ for $i = 1, ..., k$. We then employ multivariate interpolation to approximate the exponential function at any $y_q \neq y'_1, ..., y'_k$.

While there are many effective interpolation methods, only a few generalize well to multiple dimensions. Two of the most successful methods for multivariate interpolation are Radial Basis Functions (RBF) and Inverse Distance Weighting (IDW), also known as Shepard's method, a common tool in computer graphics. RBF's have proven to perform very well on multivariate interpolation when provided a sufficient set of samples [14]. In our context, however, the query point need not be in the region of known function values. To see this, note that the image $\text{Log}_x N(x)$ may not be convex and ν_k from (4.5) may not be in $\text{Log}_x N(x)$. For this reason we seek an interpolation scheme that is well-behaved outside of the sampled region, i.e. one that provides stable extrapolation away from the sampled region. To this end we use Shepard's method. The IDW scheme presented by Shepard [36] produces an interpolant from a weighted average of nearby points with weight given by $1/||x - y||_2^p$, for some $p > 0$. The interpolant at y_q is then defined as

$$\text{Exp}_x(y_q) = \left(\sum_{i=1}^{k} \frac{\text{Exp}_x(y'_i)}{||y_q - y'_i||_2^p} \right) \left(\sum_{i=1}^{k} \frac{1}{||y_q - y'_i||_2^p} \right)^{-1} \tag{4.9}$$

when $y_q \neq y'_1, ..., y'_k$ and $\text{Exp}_x(y_q) = \text{Exp}_x(y'_j)$ if $y_q = y'_j$. The power parameter p controls the scale of the interpolation. As p approaches infinity, Shepard's method provides interpolants given by the value of the function at the nearest data point. On the other hand, as p approaches 0 Shepard's interpolation approaches a constant function defined as the average of known function values.

4.3.2 Approximating Manifold Distance

When the data points x_i lie in a convex subset $U \subset \mathcal{M}$ there is a unique shortest path from x_i to x_j for each i, j. Determining the length of this geodesic in the absence of a closed form expression

for the manifold metric is however a nontrivial task. In this work we consider two approaches. First, as motivated by [8], we construct a data graph $G = (V, E)$ where each $v_i \in V$ corresponds to x_i, and $(i, j) \in E$ if $x_i \in N(x_j)$ or $x_j \in N(x_i)$. Here $N(x)$ can be defined via either the ϵ-ball definition or the k-NN definition. Each edge (i, j) is then assigned a weight given by $w_{ij} = ||x_i - x_j||_2$. The shortest path from v_i to v_j on this graph will approach the manifold distance from x_i to x_j as the number of samples approaches infinity and the volume of $N(x)$ approaches zero. For a detailed argument that this approximation converges, the reader is referred to [8].

The above approach gives a good estimate of the true distance in the case of high sampling density and low noise. Unfortunately, this estimate of the distance is typically quite sensitive to noise [25]. Also, computing all of the pairwise shortest paths on G takes roughly $O(n^2 \log n)$ operations using Dijkstra's algorithm with binary heaps. As an alternative, we consider estimating the manifold distance by first finding an approximate global parameterization of $\mathcal{M}$ and then computing the distance between each x_i and x_j in parameter space. There have recently been many algorithms proposed for the problem of manifold learning such as Laplacian Eigenmaps [7], Local Linear Embedding [34], and ISOMAP [39] which provide approximate global maps $\hat{\Phi} : \mathcal{M} \to \mathbb{R}^d$. These methods typically seek to preserve local distances at the cost of introducing some global distortion, causing them to be non-isometric at large scales. Since the algorithm in (4.5) weights data points like $1/d(x_i, m)$, we suspect that the large scale distortion will have relatively little effect on the quality of the estimator. This intuition is confirmed in our experiments as discussed in section 4.4. In addition to being more robust to noise, methods such as Laplacian Eigenmaps and Local Linear Embedding only require solving one eigendecomposition of a sparse matrix and can typically be computed in less time than the all pair shortest paths problem.

In the current work we focus on Laplacian Eigenmaps for a global parameterization to use as a proxy for manifold distance. One way to motivate Laplacian Eigenmaps is through the minimal distortion embedding of a weighted, undirected graph $G = (V, W)$ into $\mathbb{R}^d$. The vertex set $V = [1, ..., n]$ here corresponds to the n samples in the dataset and the weight between two vertices, w_{ij} reflects the spatial similarity between x_i and x_j. A common choice here is to use the Gaussian

kernel,

$$w_{ij} = \exp\left(-\|x_i - x_j\|/\sigma^2\right), \tag{4.10}$$

where σ is a scale parameter. In our context we are interested in preserving local distance, so we apply a neighborhood scheme to the graph so that each vertex is only connected to its k nearest (i.e. most similar) neighbors.

We then seek an embedding $\hat{\Phi} : V \to \mathbb{R}^d$ such that two points are close in the parameter space, $\mathbb{R}^d$, if and only if they are close in the ambient space, $\mathbb{R}^D$. To this end we consider the objective function

$$\min_{\phi} \sum_{i,j \in V} ||\hat{\Phi} v_i - \hat{\Phi} v_j||_2^2 w_{ij} := \min_{Y \in \mathbb{R}^{d \times n}} \sum_{i,j \in V} ||y_i - y_j||_2^2 w_{ij}, \tag{4.11}$$

where $Y \in \mathbb{R}^{d \times n}$ is a matrix with columns $y_i = \hat{\Phi} v_i$.

The summand is zero whenever v_i and v_j are not connected (i.e. $w_{ij} = 0$) so this objective function is a measure of the total distance in the embedded space between connected vertices, weighted by their similarity, w_{ij}.

It is important to note that the objective function is trivially minimized by the constant vector. This is not helpful in our context so we introduce the condition that each y_i must be orthogonal to the vector of all ones, i.e. $y_i^T \mathbf{1} = 0$. An additional trivial solution to the optimization problem is to choose Y to be a matrix full of zeros. To avoid this we force each column of Y to have unit length i.e. $y_i^T y_i = 1$.

Claim : For a connected graph G, the optimal $\hat{\Phi}$ in (4.11) is given by mapping each vertex v_i to the i^{th} coordinate of the d eigenvectors of the combinatorial graph Laplacian L that correspond to the d smallest magnitude nonzero eigenvalues. i.e if L has the eigendecomposition $L = Q\Lambda Q^T$ where the columns of Q correspond to eigenvectors and Λ is a diagonal matrix of eigenvectors with $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq ...$ then the optimal embedding is $Y = [q_2\ q_3\ ...\ q_{d+1}]$.

Here the combinatorial graph Laplacian is defined as $L = D - W$ where D is a diagonal matrix with $d_{ii} = \sum_{j=1}^{n} w_{ij}$.

Proof - First note that when $d = 1$ we have

$$\begin{aligned}\sum_{i,j=1}^{n} ||y_i - y_j||_2^2 w_{ij} &= \sum_{i,j=1}^{n} (y_i - y_j)^2 w_{ij} \\ &= \sum_{i,j=1}^{n} (y_i^2 - 2y_i y_j + y_j^2) w_{ij} \\ &= 2\sum_{i=1}^{n} d_{ii} y_i^2 - 2\sum_{i,j=1}^{n} y_i y_j w_{ij} \\ &= 2\left(y^T D y - y^T W y\right) \\ &= 2y^T L y.\end{aligned} \tag{4.12}$$

Now we let $L = Q\Lambda Q^T$, and

$$\begin{aligned}y^T L y &= (Q^T y)^T \Lambda (Q^T y) \\ &= \sum_{i=1}^{n} (q_i^T y)^2 \lambda_i.\end{aligned} \tag{4.13}$$

Since L is symmetric we know that $\{q_i\}_{i=1}^n$ form an orthogonal basis for $\mathbb{R}^n$ so we can write $y = \sum_{i=1}^n c_i q_i$ with $\sum_{i=1}^n c_i^2 = 1$. It follows that $q_i^T y = c_i$ (assuming each q_i has unit length). Our expression becomes

$$y^T L y = \sum_{i=1}^{n} c_i \lambda_i. \tag{4.14}$$

We also note that since $L = D - W$, $\mathbf{1}$ is always an eigenvector of L with eigenvalue 0. So the y that minimizes the above expression and is orthogonal to $\mathbf{1}$ is given by setting y to the eigenvector corresponding to λ_2. For the $d > 1$ case, an inductive argument based on orthogonal subspaces shows the original claim.

For reasons as discussed in [40] and [30] we introduce the Normalized Combinatorial Graph Laplacian, $\hat{L}$, defined as

$$\hat{L} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{4.15}$$

where $(D^{-\frac{1}{2}})_{ii} = d_{ii}^{-\frac{1}{2}}$. This normalization adjusts for nonuniform sampling density on the underlying manifold. [37] showed that under this normalization, $\hat{L}$ converges to the Laplacian

operator defined on the manifold (i.e. the Laplace-Beltrami operator) as the number of points goes to infinity under some light assumption on the sampling distribution.

As seen in (4.11), this embedding in $\mathbb{R}^d$ preserves small scale distance in the data graph G, which in turn serves as a proxy for the true manifold distance. For this reason, we choose to approximate the true distance $d(x_i, x_j)$ with the distance in this embedded space $\|y_i - y_j\|$. Since this embedding only introduces distortion at large scales and the gradient descent algorithm uses weights that are inversely proportionate to distance, we suspect (and later confirm) that this proxy for distance will perform as well as the shortest path distance on the data graph.

4.3.3 Determining an Appropriate Initial Guess

Under appropriate assumptions on $\mathcal{M}$, (4.5) will converge to the true median if the initial guess, $m^{(0)}$, is sufficiently close to the true median. Finding a suitable initialization is therefore very important when estimating the geometric median. Unfortunately, taking the median of the sample points in the ambient space may give an initial estimate that is quite far from the manifold. To determine a more accurate initial guess we again note that the distance in the parameter space given by Laplacian Eigenmaps can serve as a proxy for manifold distance at small scales and, since the median is not sensitive to distance points, we suspect that this proxy may provide a suitable initial guess.

In what follows, let y_i denote the embedding assigned to data point x_i by Laplacian Eigenmaps, $y_i = \hat{\Phi} x_i$. We then have that $d(x_i, x_y) \sim ||y_i - y_j||$ when $d(x_i, y_i)$ is small. It is reasonable to assume that the embedding of the true median $m^\star$, $\hat{\Phi} m^\star$ would not be too far from the median of $y_1, ..., y_n$. We use this notion to get an initial guess, $m^{(0)}$, by first finding a locally quasi-isometric map $\hat{\Phi} : \mathcal{M} \to \mathbb{R}^d$, computing the median of the embedded coordinates $y_i = \Phi x_i$, $i = 1, ..., n$, denoted $\bar{y}$, and then assigning $m^{(0)} = \hat{\Phi}^{-1} \bar{y}$. Of course, methods such as Laplacian Eigenmaps do not provide an inverse map directly. We can however use multivariate interpolation to approximate the inverse map from the known samples $x_i = \hat{\Phi}^{-1} y_i$. This approach to inverting nonlinear dimensionality reductions has been studied in [29]. Again, we suggest using Shepard's method as an

interpolation scheme as $\hat{\Phi}\mathcal{M}$ is not necessarily convex, and we are not guaranteed that $\bar{y} \in \hat{\Phi}\mathcal{M}$. As discussed earlier, Shepard's method is more stable when the query value is outside of the sampling domain.

4.3.4 Adjusting for Finite Sampling

A common issue with applying gradient descent techniques to non-differentiable cost functions on a finite set of samples is that as the iterates converge they may overshoot the true minimum, leading to orbits. In other words, the algorithm gets stuck at the sampling resolution. Two methods for resolving this issue are decreasing the descent constant (in our case α) and resampling the function at a higher resolution. Once orbits are detected α is decreased by 10%. If the iterations still do not converge after decreasing α ten times, the tangent plane is approximated from $\cup_{m^{(k)} \in R} N(m^{(k)})$ where R denotes the points in the orbit of $\{m^{(k)}\}$. This tangent plane is then resampled at a high density. These samples are then pushed back to $\mathcal{M}$ using multivariate interpolation and augmented onto the training set. Note that the algorithm described in this chapter estimates the geometric median of a set of point $x_1, ..., x_n$, but more data, $x_{n+1}, ..., x_m$ may be easily used to better understand the geometry of $\mathcal{M}$ (see algorithm 2).

4.3.5 An Algorithm for Estimating the Geometric Median on Data Manifolds

Algorithm 2 summarizes the estimator for the geometric median on data manifolds developed in this chapter.

4.4 Experiments

As a demonstration of the proposed estimation scheme we consider finding the median of a random set of points drawn from the first octant of the unit d-sphere,

$$\mathcal{M} = S_d \cap (\mathbb{R}^+)^{d+1} = \{x \in \mathbb{R}^{d+1} \mid x_i \geq 0 \ , \ \sum x_i^2 = 1\}. \tag{4.16}$$

This manifold is then embedded in $\mathbb{R}^{10}$ and n_{Tr} samples are taken from $\mathcal{M}$. Ten of these

Input :
$X = [x_1, ..., x_m]$, $x_i \in R^D$ - a set of samples from a manifold $\mathcal{M}$
I - index set for points of which median is to be computed
d - the intrinsic dimensionality of $\mathcal{M}$
Output :
m - the geometric median of $\{x_i\}_{i \in I}$

compute Laplacian Eigenmap of X, $y_i = \Phi x_i$, $i = 1, \ldots, m$
compute $\bar{y}$, the median of $\{y_i \mid i \in I\}$ in $\mathbb{R}^d$, with (4.3)
set $m^{(0)} = \Phi^{-1}\bar{y}$ where Φ^{-1} is interpolated from $x_i = \Phi^{-1} y_i$ with Shepard's method
while $||m^{(k+1)} - m^{(k)}|| > tol$ **do**
    approximate $\mathrm{Log}_m(\,\cdot\,)$ from the first d principal components of $N(m^{(k)})$ (using full dataset)
    approximate ν_k from (4.5) using $d(x_i, x_j) = ||y_i - y_j||$ for $i, j \in I$
    compute $m^{(k+1)} = \mathrm{Exp}_m(\alpha \nu_k)$ using Shepard's method on $x_i = \mathrm{Exp}_m y_i$, $x_i \in N(m^{(k)})$
    if orbits detected then
        set $\alpha \leftarrow 0.9\alpha$
        if decreased α 10 times then
            resample manifold and append to dataset
        end
    end
end

Algorithm 2: Estimating the geometric median on data manifolds.

samples are randomly selected and their median is estimated. To compute a ground truth median, $m^\star$, we take a dense, noise-free sampling of $\mathcal{M}$ (2,500 samples) and do an exhaustive search for the median using the shortest path on the graph. Performance is measured by comparing the proposed method with m_{triv}, the trivial estimator given by the sample with the minimum total shortest path to each point on the graph,

$$m_{triv} = \arg\min_{x_i} \sum_j d_G(x_i, x_j), \tag{4.17}$$

where $d_G(\cdot, \cdot)$ denotes the shortest path on the graph G. To evaluate an individual estimate's performance we compare the cost function at the estimator with the cost function at the true median,

$$score(m) = \frac{C_1(m)}{C_1(m^\star)}. \tag{4.18}$$

Here $C_1(m)$ is computed using the dense sampling of $\mathcal{M}$.

We compared the cost function relative to optimal for 6 estimators: (1) m_{triv} defined above, (2) the initial guess giving by pushing the median in Laplacian Eigenmap parameter space back to $\mathcal{M}$, (3) the estimator given in algorithm 2 using graph distance and RBF interpolation, (4) the same estimator using IDW interpolation, (5) the estimator in algorithm 2 using Laplacian Eigenmap coordinates as a proxy for distance and RBF interpolation, and (6) the same estimator using IDW interpolation. Figures 4.2, 4.3, and 4.4 show the score of each estimator as sample size, noise level, and dimensionality are varied respectively.

Our first observation is that the Laplacian Eigenmap proxy for manifold distance does indeed perform very similarly to the shortest path on the data graph. Also as expected, IDW seems to be a more stable interpolation scheme in this context than RBF. In figure 4.2 we see that the trivial estimator is sensitive to low sampling density whereas all of our estimators are not. Figure 4.3 shows that all of the estimators are sensitive to high levels of noise, which is not surprising. What is interesting is that, although at low noise levels the gradient descent algorithm typically increases the quality of the estimate, for very high levels of noise the initial guess actually provides a better estimate. This can be explained by the fact in the presence of high noise, the estimated tangent

plane from PCA may be quite far from the true tangent plane. On the other hand, the graph Laplacian has been effectively used for manifold denoising [20], [28]. With this in mind, it is not surprising that the Laplacian Eigenmap approach to finding an initial guess is more robust to extreme noise than the gradient descent algorithm. Future work could include incorporating a denoising step based on the graph Laplacian.

4.5 Conclusion

In this chapter an estimator for the geometric median was developed. In particular, a gradient descent method from differential geometry was applied to equation-free data analysis. This was accomplished by determining appropriate approximations for the logarithm and exponential maps and the manifold distance. Further, a suitable initial guess was presented and numerical stability issues were addressed. The proposed method was applied to synthetic datasets and shown to outperform a trivial estimate of the median. The estimator's sensitivity to sampling density, noise, and dimensionality was then examined. While high levels of noise decreased the quality of the estimator, the estimator was shown to not be sensitive to low sampling density. Further work could include studying the effects of the parameters in algorithm 2, such as the number of neighbors used when computing PCA, and studying the convergence criteria of our estimator. In the next chapter this method is applied to a real world neurological dataset to demonstrate its usefulness in equation-free data analysis.

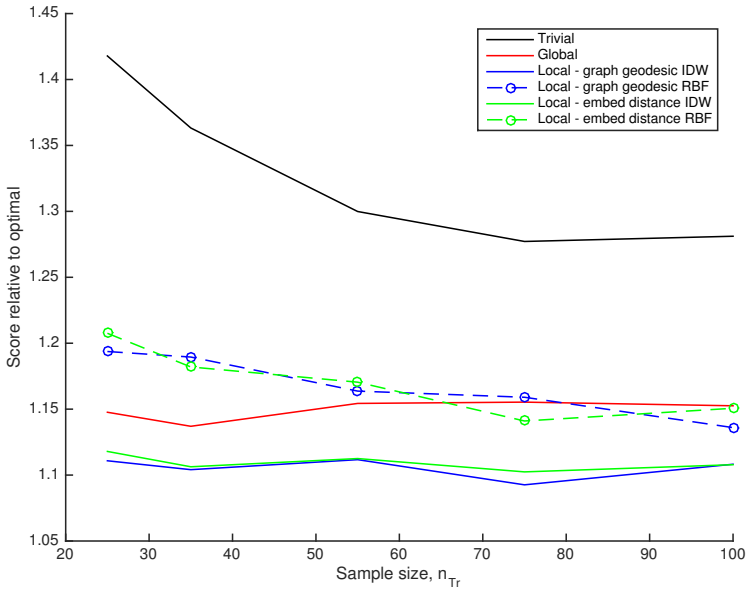

Figure 4.2: A comparison of several estimators' sensitivity to sample size. $d = 3$, noise level $= 0.05$.

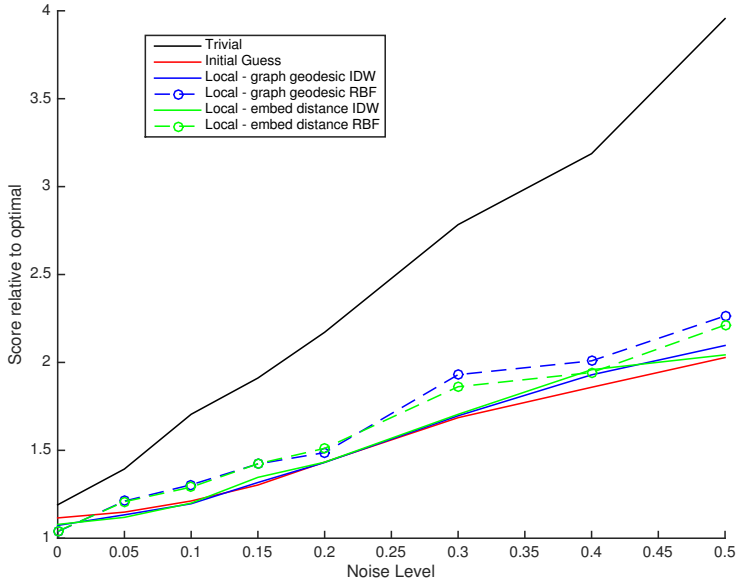

Figure 4.3: A comparison of several estimators' sensitivity to noise. $d = 3$, sample size $= 30$.

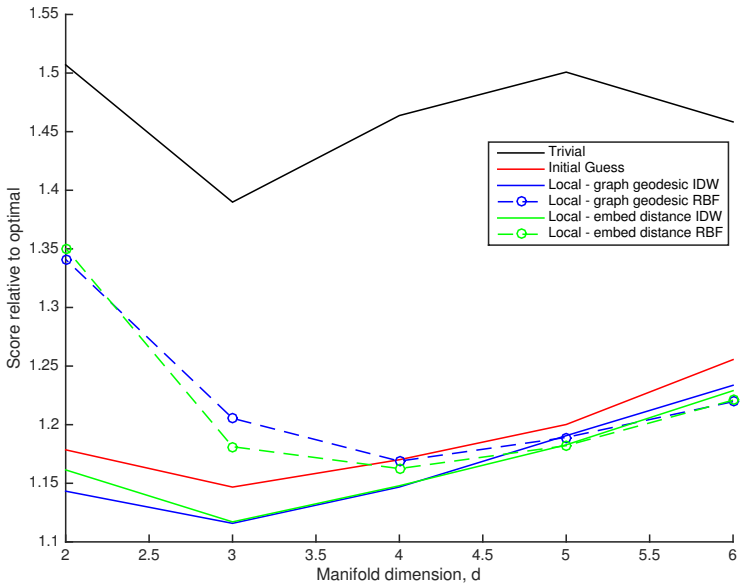

Figure 4.4: A comparison of several estimators' sensitivity to dimension. noise level = 0.05, sample size = 45.

Chapter 5

Creating a Biomarker for Epileptogenesis

5.1 Background

5.1.1 Epilepsy

Epilepsy is one of the most common neurological diseases, affecting nearly 50 million people worldwide yet the causes of most cases are unknown [1]. Some cases are the result of a traumatic event such as stroke, brain tumor, or head injury. In this case epilepsy slowly develops by a process called epileptogenesis. An active research question is how to quantify and measure the progression of epileptogenesis. This will be the focus of this chapter.

A large challenge for understanding epileptogenesis and other neurological conditions is creating an informative biomarker. The National Institute of Health defines a biomarker as a "characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention" [13]. For instance, hepatitis B serologic tests can determine the presence of certain antigens and antibodies and help understand in what stage of hepatitis B a patient is. Unfortunately, there is no know physical substance that can indicate the stage of epileptogenesis. The goal of this chapter is to develop a computational biomarker based on measurements of electrical activity in the hippocampus.

5.1.2 Experiment and Data

In this experiment, an animal model is used to collect hippocampal auditory evoked potentials (hAEP) before and after dosing of a certain epileptogenesis inducing drug. Over the course of the

experiment a lab rat progresses through four stages: pre-injection (baseline), comatose immediately following the injection (silent), recovery from the injection (latent), and repeated seizures (chronic).

Every thirty minutes over the course of the experiment, the electric potential in the animal's hippocampus is collected at 10 kHz immediately following an auditory evocation. These hippocampal auditory evoked potentials (hAEPs) are then decomposed using a wavelet decomposition, and a t-test is used to find in wavelet space the most significant predictors of the rat's condition. Finally, temporally adjacent measurements are concatenated to form a time-delay embedded signal. These concatenated responses form vectors with 12,000 elements. For more details on the experiment and preprocessing see [27]. In the following sections we estimate the local intrinsic dimensionality of these signals and develop a method to decode the progression of epileptogenesis.

5.2 Intrinsic Dimensionality

5.2.1 Visualizing Dataset

Before performing analysis on the hAEP dataset it is useful to visualize it. To this end Laplacian Eigenmaps are used to determine a low-dimensional representation of the full dataset. The result is seen in figure 5.1. We see that there indeed appears to be some low-dimensional structure to the hAEP data. Further, we notice that different regions of the data manifold do indeed correspond to specific stages.

5.2.2 Local Dimensionality Estimates

Here we apply the local dimensionality estimator from chapter 3 to the set of hAEPS. Figure 5.2 shows the mean and standard deviation of the dimensionality estimate over 100 random partitionings. $l = 2$ partitioning steps are used for reasons discussed below. We see that indeed different regions of the hAEP manifold exhibit different intrinsic dimensionality. Further, as one would expect, we see that while the rat is in the silent stage (i.e. in a coma immediately following the injection) their hAEPs arrange near a lower-dimensional region of the manifold than while the

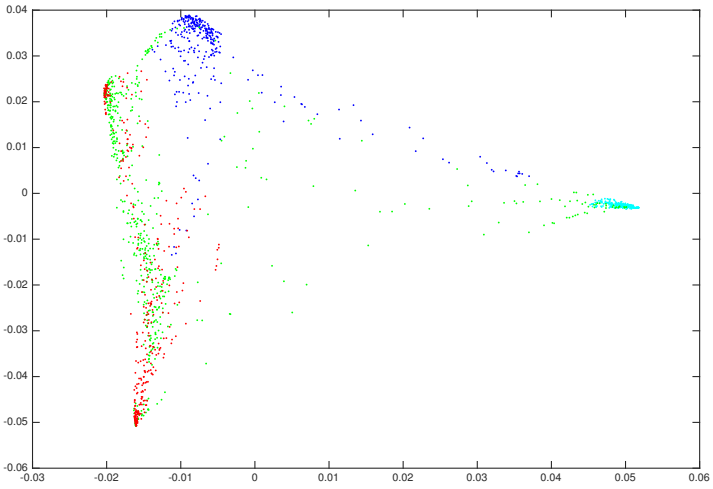

Figure 5.1: Laplacian Eigenmaps of hAEP data for each condition. Blue - Baseline, Cyan - Silent, Green - Latent, Red - Chronic.

	Standard Deviation					Mode Prevalence				
l	Baseline	Silent	Latent	Chronic	All	Baseline	Silent	Latent	Chronic	All
1	0.1051	0.2709	0.0932	0.2420	0.1612	0.9449	0.8410	0.9437	0.8434	0.9027
2	0.0000	0.2447	0.1455	0.0379	0.1071	1.0000	0.8510	0.9142	0.9782	0.9362
3	0.0460	0.1846	0.1119	0.1036	0.1088	0.9713	0.9023	0.9366	0.9439	0.9396
4	0.3590	0.2661	0.2234	0.2089	0.2551	0.7732	0.8560	0.8627	0.2089	0.6885

Table 5.1: The standard deviation and mode prevalence of the dimensionality estimate using random dimension median partitioning with l partitioning steps.

rat is awake and healthy.

If the method is giving an accurate local intrinsic dimensionality estimate we expect points to have similar dimensionality as their immediate neighbors. We thus measure the local consistency of our estimator by comparing each point to its 10 nearest neighbors. Specifically, we measure two things: (1) the standard deviation of the dimensionality estimator within each neighborhood and (2) the number of points within each neighborhood where the estimator equals the neighborhood's mode estimate. Table 5.1 shows (1) and (2) when the local dimensionality is estimated by the random dimension median approach,[1] with varying values of l. Table 5.2 shows (1) and (2) when the local dimensionality is estimated using k_D nearest neighbors. To make a fair comparison between the two methods, the dimensionality estimate from the random dimension median partitioning method is rounded to the nearest integer.

We see that the k_D-NN approach needs k_D to be around 250 in order to get similar local consistency as obtained by the random dimension median approach. For some stages the number of samples is only a little over 250 so this is almost equivalent to computing the global estimate. Accordingly, we choose to use the random dimension median method and, based on the results in Table 5.1, we select $l = 2$.

[1] The local dimension estimator is computed for each stage individually. The "All Stages" column is a weighted average of each stages score (weighted by number of data points in each stage).

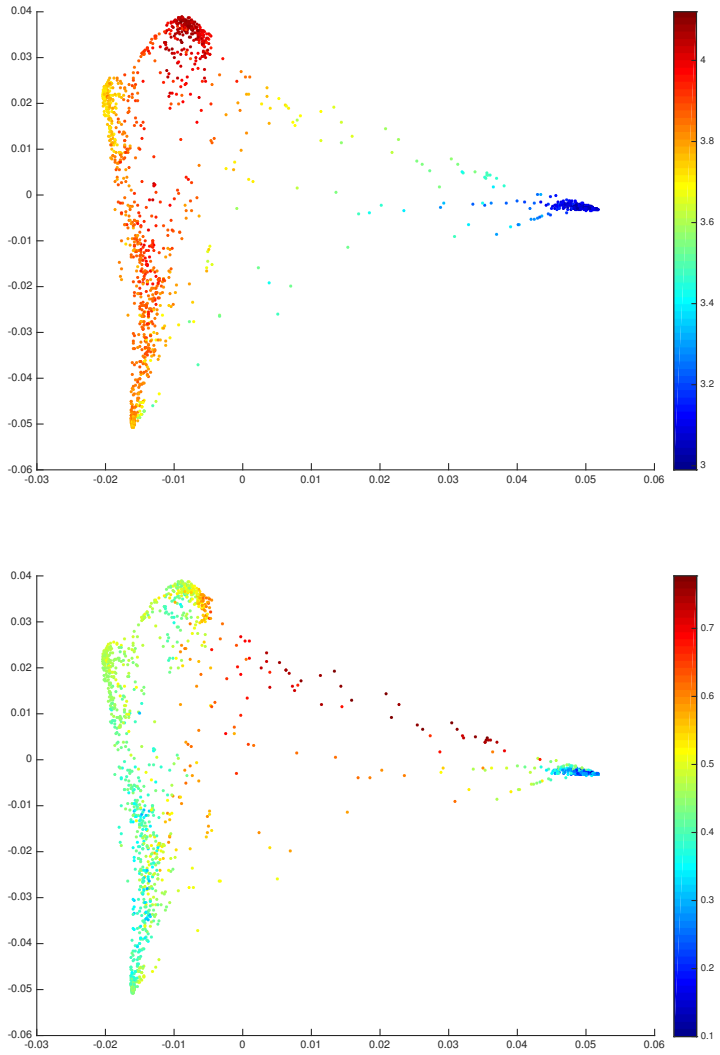

Figure 5.2: The mean (top) and standard deviation (bottom) of the dimensionality estimate for the hAEP dataset over 100 trials using $l = 2$.

	Standard Deviation					Mode Prevalence				
k_D	Baseline	Silent	Latent	Chronic	All	Baseline	Silent	Latent	Chronic	All
25	1.0319	0.7789	1.3428	1.1631	1.1403	0.5150	0.5742	0.4909	0.5147	0.5157
50	0.5751	0.3384	0.6211	0.5550	0.5478	0.5936	0.6610	0.6852	0.6948	0.6645
75	0.4824	0.1803	0.3771	0.3394	0.3562	0.6902	0.7809	0.8164	0.8123	0.7835
100	0.3591	0.0759	0.2844	0.2956	0.2668	0.7921	0.8683	0.8481	0.8425	0.8387
150	0.3143	0.0000	0.2486	0.2020	0.2086	0.8433	1.0000	0.8503	0.8838	0.8686
200	0.1566	0.0000	0.2606	0.0087	0.1352	0.9150	1.0000	0.8354	0.9967	0.9180
250	0.0000	0.0000	0.1329	0.0151	0.0551	1.0000	1.0000	0.9248	0.9945	0.9696

Table 5.2: The standard deviation and mode prevalence of the dimensionality estimate using k_D-NN.

5.2.3 Dimensionality as Disease Progresses

Figure 5.3 shows the mean and standard deviation of the local intrinsic dimensionality estimate of each rat as the disease progresses. We see that in baseline the responses are roughly four-dimensional. When the epilepsy inducing drug is injected and the rat enters the silent stage, the intrinsic dimension drops down to three. During the latent stage the dimension returns to four at different rates over the test rats. By the time the rat is experiencing chronic epileptic seizures the dimension consistently returns to roughly four.

5.3 Decoding the Progression of Epileptogenesis

We now turn our attention to developing a computational biomarker for the progression of epileptogenesis. We assume that each rat shares a common drift on the data manifold that is distorted by high-dimensional noise. We do not however assume that these drifts occur at the same rate and instead assign each rat an intrinsic time scale $\tau_i(t)$ where i denotes a specific test rat. To get a sense of the noise free trajectory we utilize the geometric median algorithm developed in chapter 4.

Unfortunately, the method developed in chapter 4 depends on extensive computations of Singular Value Decompositions in the ambient space, so instead of using the wavelet coefficients discussed previously, we parameterize the set of hAEPs using cubic spline approximation. We note

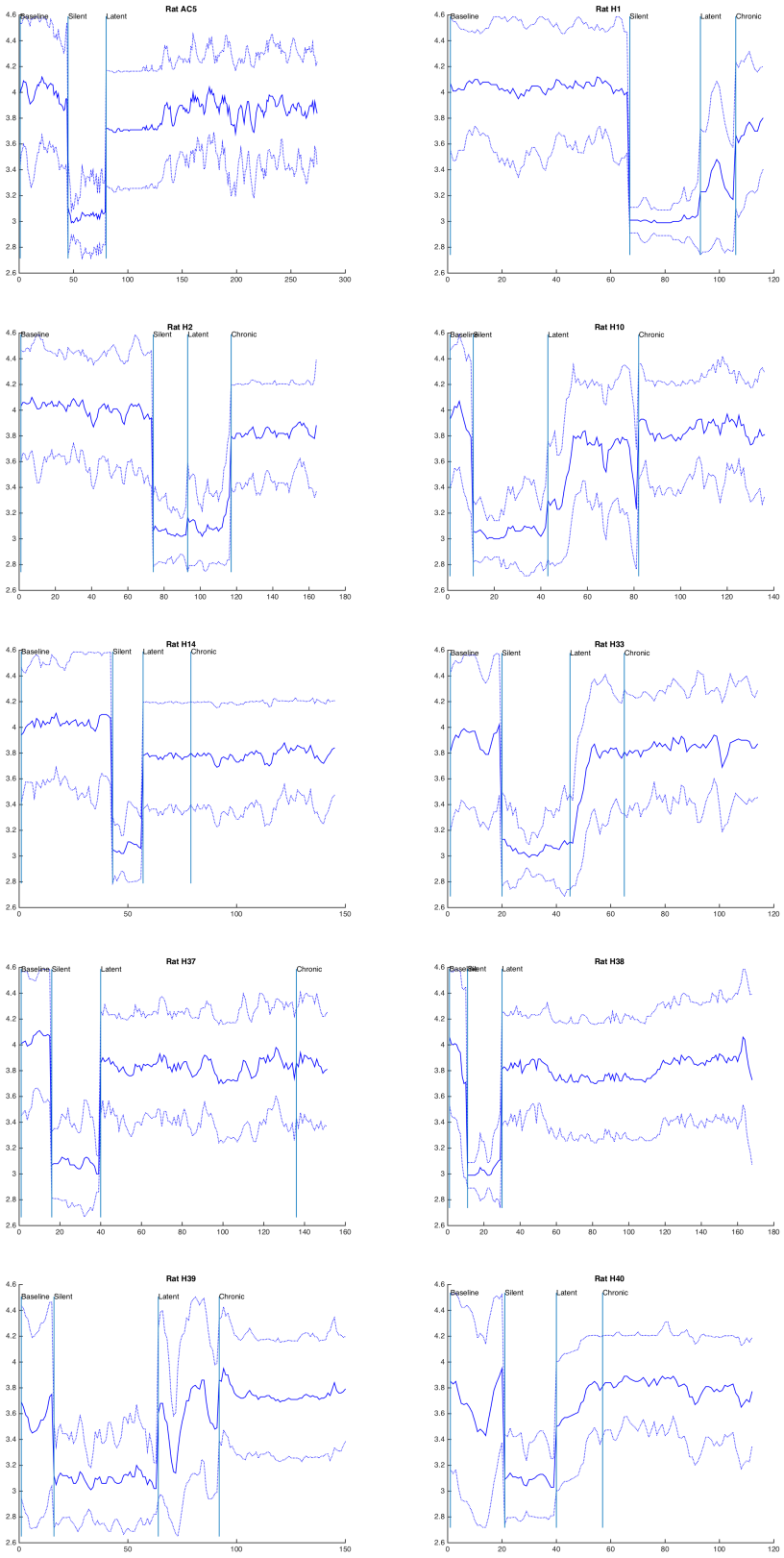

Figure 5.3: The mean and standard deviation of the local intrinsic dimensionality estimate as the disease progresses over 100 trials using $l = 2$.

that projecting the hAEPs onto the set of cubic splines may cause a decrease in dimensionality, but argue that since the strongest predictors are low frequency wavelet coefficients, this projection will preserve the structure of the hAEP dataset. Projecting the hAEP signals onto cubic spline approximations gives vectors in $\mathbb{R}^{13}$ corresponding to the set of hAEPs. In what follows let $y_i(t)$ denote the spline coefficients associated with animal i's hAEP at time t.

With this model for the hAEP dataset we seek a path $\mu(t)$ that describes the average trajectory of epileptogenesis and a set of intrinsic timescales $\{\tau_i\}$ corresponding to each animal. In particular, we seek to minimize the residual of $y_i(t)$ on $\mu(\tau_i(t))$ in L^1 norm, i.e. solve the minimization problem

$$\min_{\mu,\tau_i} \sum_{i=1}^{n_{rats}} \int_0^1 d(y_i(t), \mu(\tau_i(t))dt. \tag{5.1}$$

For simplicity we rescale the true timescale so that $t \in [0,1]$ and force the intrinsic time to remain on this scale, $\tau_i : [0,1] \to [0,1]$. These timescales τ_i serve as computational biomarkers as they provide a notion of how far epileptogenesis has progressed for each animal.

To solve this optimization problem, we propose the two step optimization scheme shown in algorithm 3.

initialize $\tau_i(t) = t$, $i = 1, \ldots, n_{rats}$
while stopping criteria not met do
  for $s \in [0,1]$ **do**
    set $\mu^{(k+1)}(s) = M(\{y_i(t) \mid \tau_i(t) \in (s - w/2, s + w/2)\})$
  end
  for $i = 1, ..., n_{rats}$ **do**
    set $\tau_i(t) = \arg\min_{\tau(t)} \int d(y_i(t), \mu(\tau(t)))dt$
  end
end

Algorithm 3: A two step optimization scheme to minimize the functional in (5.1).

Here $M(S)$ is the geometric median of the set S, w is a parameter that declares how large of a window to use for temporal averaging, and the stopping criteria is based on how much the residual decreases each iteration. At each iteration, the method developed in chapter 4 is used to estimate $M(S)$ and the dynamic programming approach in algorithm 4 is used to find the minimizing set of

τ_i. One should note that by forcing $1 \leq k \leq j$ in this algorithm we force τ_i to be non-decreasing. This is because the animals included in this experiment all developed epilepsy eventually, so we would like the intrinsic clock to be monotonically non-decreasing.

for $i, j = 1, ..., n$ **do**
| set $C(i,j) = d(y(t_i), \mu(\tau_j))$
end
for $j = 1, n$ **do**
| set $J(1,j) = C(1,j)$
end
for $i = 1, ..., n$ **do**
| **for** $j = 2, ..., n$ **do**
| | set $J(i,j) = C(i,j) + \min_{1 \leq k \leq j}(J(i-1,k))$ and $U(i,j) = \arg\min_{1 \leq k \leq j}(J(i-1,k))$
| **end**
end
Recursively define τ with $\tau(n) = \arg\min_j J(n,j)$ and $\tau(i-1) = U(i, \tau(i))$

Algorithm 4: A dynamic programming algorithm to solve for $\tau_i(t)$.

5.3.1 Biomarker Results

Figure 5.4 shows the value of $\tau_i(t)$ for each animal during the course of the experiment. We see that in most animals τ_i does indeed correspond to the stage of epileptogenesis but there is not a clear threshold to determine when a animal will begin having spontaneous seizures. Nonetheless, the proposed biomarker does seem to track the progression of epileptogenesis and provides insights on the variability in progression between different rats. For instance, two rats seem to exhibit characteristics of hippocampal damage very early, two seem to progress slowly, and the rest seem to progress at roughly the same rate. Information like this could be beneficial for quantifying the effectiveness of pharmaceutical interventions.

5.4 Conclusion

In this chapter we applied the methods developed in the previous two chapters to high-dimensional observations of a neurological process. Tracking the progression of epileptogenesis is a nontrivial problem and traditional statistics do not provide very high predictability. As we have seen in this chapter, looking at the data from a geometric perspective is beneficial in several ways. First,

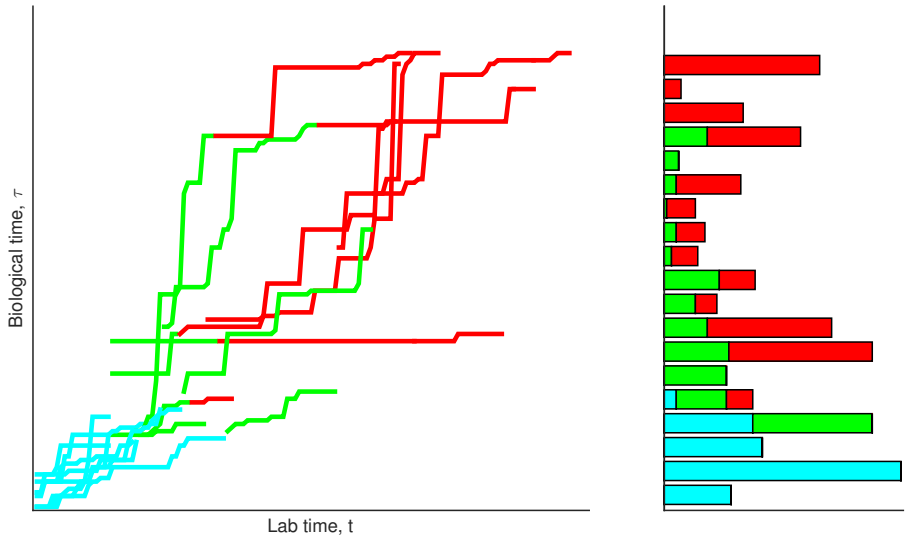

Figure 5.4: Recovered $\tau_i(t)$ for each rat colored by condition (left) and histogram of τ values by condition (right). Cyan - Silent, Green - Latent, Red - Chronic.

we saw that the responses do indeed lie near a remarkably low-dimensional manifold embedded in the ambient Euclidean space. Further, the dimensionality of this manifold was shown to vary across different regions of the ambient space. In particular, when the animal is in a comatose, its hippocampal activity was shown to exhibit lower dimensionality. This is interesting as it quantifies the intuitive notion that while the animal is unconscious there are fewer degrees of freedom in its neurological activity.

Next we showed how the intrinsic dimensionality of the responses can be leveraged to track the progression of an underlying neurological process. By using a relatively simple model that utilized no equations unique to neurology, we were able to construct a computational biomarker that reflects the progression of epileptogenesis over the course of the experiment. To our knowledge, this is the first proposed method to solve for an underlying biological timescale based solely on a set of high dimensional observations. This computational biomarker could lead to new insights on the variability of the underlying processes that occur during epileptogenesis and serve as a useful tool for experimental pharmacology.

Chapter 6

Conclusion

In this thesis we have demonstrated that ideas from differential geometry can be meaningfully applied to problems in equation-free data analysis. Understanding data from a geometric perspective has proven to be an effective way to leverage the underlying structure in high-dimensional datasets. In particular, we have seen that the intrinsic dimensionality of a dataset can be recovered, and that approximations for the tangent plane and manifold distance can be used to construct a more meaningful notion of centrality.

Much of the existing work on manifold-valued data analysis has assumed some degree of prior knowledge of the data, be it closed form expressions for the data's supporting manifold or at least its dimensionality. The first topic of this thesis addressed the problem of estimating a dataset's intrinsic dimensionality based only on sample values. Existing estimators for global dimensionality were presented and extended to the problem of local dimensionality estimation. The performance of this estimator was first demonstrated on a synthetic dataset and later shown to provide locally consistent estimates on real world data.

Then, in the second portion of this thesis, a novel extension of a previous algorithm to equation-free data analysis was developed. The geometric median, a meaningful notion of centrality for high-dimensional data with low-dimensional structure, was formalized and an estimator for the geometric median in equation-free models was proposed. This estimator was then applied to a synthetic dataset, demonstrating that by leveraging the underlying structure of a dataset one can indeed provide a more meaningful notion of centrality for manifold-valued data.

The previous two section were then applied to a real world high-dimensional data problem. In particular, the dimensionality estimator was used to understand how the geometric structure of a certain neurological measurement changed over the course of an experiment. We saw that immediately following a traumatic shock, an animal model's hippocampal activity exhibited fewer degrees of freedom than when the animal was awake. As the animal recovered from this shock, we saw that its neurological activity returned to normal. The rate at which this return to normal occurred may help understand the variability between different instances of the experiment.

Finally, the proposed estimator for the geometric median was used to create a computational biomarker for the progression of epileptogenesis. Since each animal may develop epilepsy at a different rate, we used a joint optimization scheme to fit a model where each animal has its own unique biological timescale. The geometric median estimator and a dynamic programming algorithm were used to find a trajectory and a set of animal dependent biological timescales that best fit the observations. These biological timescales then served as a computation biomarker for the progression of epileptogenesis. While there was not a clear threshold for when the animal will begin exhibiting epileptic symptoms, this computation biomarker may provide insights on how the neurological processes behind epileptogenesis vary across experiments.

The two estimators developed in this thesis are highly generalized and could be applied to a wide range of problems in data science. In any situation where a high-dimensional dataset is to be parameterized, having a suitable estimate of the intrinsic dimensionality will allow the data scientist to know how many parameters should be included. Further, many real datasets exhibit different dimensionality depending on the configuration of an underlying process. Overlooking this potential variability in intrinsic dimensionality could cause the data scientist to underestimate or overestimate the appropriate number of parameters to use.

Using an appropriate notion of centrality for data with low intrinsic dimensionality is also an important consideration in high-dimensional data analysis. In many problems high-dimensional observations are assumed to have a nonlinear, low-dimensional structure. In this case the mean in the ambient Euclidean space will typically have a different structure than the original data. By

forcing the average to remain on the low-dimensional structure, the data scientist will be able to construct a more meaningful average.

Bibliography

[1] Epilepsy. `http://www.who.int/mediacentre/factsheets/fs999/en/`. Accessed: 2017-03-25.

[2] Bijan Afsari, Roberto Tron, and René Vidal. On the convergence of gradient descent for finding the riemannian center of mass. SIAM Journal on Control and Optimization, 51(3):2230–2260, 2013.

[3] Marc Arnaudon, Frédéric Barbaresco, and Le Yang. Medians and means in riemannian geometry: existence, uniqueness and computation. In Matrix Information Geometry, pages 169–197. Springer, 2013.

[4] Marc Arnaudon, Clément Dombry, Anthony Phan, and Le Yang. Stochastic algorithms for computing means of probability measures. Stochastic Processes and their Applications, 122(4):1437–1455, 2012.

[5] Marc Arnaudon and Laurent Miclo. A stochastic algorithm finding generalized means on compact manifolds. Stochastic Processes and their Applications, 124(10):3463–3479, 2014.

[6] Marc Arnaudon, Laurent Miclo, et al. A stochastic algorithm finding p-means on the circle. Bernoulli, 22(4):2237–2300, 2016.

[7] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In NIPS, volume 14, pages 585–591, 2001.

[8] Mira Bernstein, Vin De Silva, John C Langford, and Joshua B Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Technical report, Department of Psychology, Stanford University, 2000.

[9] RN Bhattacharya, L Ellingson, X Liu, V Patrangenaru, and M Crane. Extrinsic analysis on manifolds is computationally faster than intrinsic analysis with applications to quality control by machine vision. Applied Stochastic Models in Business and Industry, 28(3):222–235, 2012.

[10] Hervé Cardot, Peggy Cénac, and Mohamed Chaouch. Stochastic approximation for multivariate and functional median. In Proceedings of COMPSTAT'2010, pages 421–428. Springer, 2010.

[11] Hervé Cardot, Peggy Cénac, Pierre-André Zitt, et al. Efficient and fast estimation of the geometric median in hilbert spaces with an averaged stochastic gradient algorithm. Bernoulli, 19(1):18–43, 2013.

[12] Kevin M Carter, Raviv Raich, and Alfred O Hero III. On local intrinsic dimension estimation and its applications. Signal Processing, IEEE Transactions on, 58(2):650–663, 2010.

[13] WA Colburn, Victor G DeGruttola, David L DeMets, Gregory J Downing, Daniel F Hoth, John A Oates, Carl C Peck, Robert T Schooley, Bert A Spilker, Janet Woodcock, et al. Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework. biomarkers definitions working group. Clinical Pharmacol & Therapeutics, 69:89–95, 2001.

[14] Gregory E Fasshauer. Meshfree approximation methods with MATLAB, volume 6. World Scientific, 2007.

[15] P Thomas Fletcher, Suresh Venkatasubramanian, and Sarang Joshi. The geometric median on riemannian manifolds with application to robust atlas estimation. NeuroImage, 45(1):S143–S152, 2009.

[16] Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. Computers, IEEE Transactions on, 100(2):176–183, 1971.

[17] Ronaldo Malheiros Gregório and Paulo Roberto Oliveira. A proximal technique for computing the karcher mean of symmetric positive definite matrices. Optimization Online, 2013.

[18] David Groisser. Newton's method, zeroes of vector fields, and the riemannian center of mass. Advances in Applied Mathematics, 33(1):95–135, 2004.

[19] Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in r d. In Proceedings of the 22nd international conference on Machine learning, pages 289–296. ACM, 2005.

[20] Matthias Hein and Markus Maier. Manifold denoising. In NIPS, volume 19, pages 561–568, 2006.

[21] Daniel N Kaslovsky and François G Meyer. Non-asymptotic analysis of tangent space perturbation. Information and Inference, 3(2):134–187, 2014.

[22] Harold W Kulin and Robert E Kuenne. An efficient algorithm for the numerical solution of the generalized weber problem in spatial economics. Journal of Regional Science, 4(2):21–33, 1962.

[23] Huiling Le. Locating fréchet means with application to shape spaces. Advances in Applied Probability, 33(02):324–338, 2001.

[24] Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. In Advances in Neural Information Processing Systems 17, volume 48109, page 1092, 2004.

[25] Bo Li, De-Shuang Huang, and Chao Wang. Improving the robustness of isomap by de-noising. In Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, pages 266–270. IEEE, 2008.

[26] Anna V Little, Yoon-Mo Jung, and Mauro Maggioni. Multiscale estimation of intrinsic dimensionality of data sets. In AAAI Fall Symposium: Manifold Learning and Its Applications, 2009.

[27] François G Meyer, Alexander M Benison, Zachariah Smith, and Daniel S Barth. Decoding epileptogenesis in a reduced state space. In Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on, pages 152–157. IEEE, 2016.

[28] François G Meyer and Xilin Shen. Perturbation of the eigenvectors of the graph laplacian: Application to image denoising. Applied and Computational Harmonic Analysis, 36(2):326–334, 2014.

[29] Nathan D Monnig, Bengt Fornberg, and Francois G Meyer. Inverting nonlinear dimensionality reduction with scale-free radial basis function interpolation. Applied and Computational Harmonic Analysis, 37(1):162–170, 2014.

[30] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems, 2:849–856, 2002.

[31] Lawrence M Ostresh Jr. On the convergence of a class of iterative methods for solving the weber location problem. Operations Research, 26(4):597–609, 1978.

[32] Daniele Panozzo, Ilya Baran, Olga Diamanti, and Olga Sorkine-Hornung. Weighted averages on surfaces. ACM Transactions on Graphics (TOG), 32(4):60, 2013.

[33] Joel W Robbin and Dietmar A Salamon. Introduction to differential geometry. ETH, Lecture Notes, preliminary version, January, 2011.

[34] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500):2323–2326, 2000.

[35] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In International Conference on Artificial Neural Networks, pages 583–588. Springer, 1997.

[36] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In Proceedings of the 1968 23rd ACM national conference, pages 517–524. ACM, 1968.

[37] Amit Singer. From graph to manifold laplacian: The convergence rate. Applied and Computational Harmonic Analysis, 21(1):128–134, 2006.

[38] Anuj Srivastava and Eric Klassen. Monte carlo extrinsic estimators of manifold-valued parameters. IEEE Transactions on Signal Processing, 50(2):299–308, 2002.

[39] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.

[40] Ulrike Von Luxburg. A tutorial on spectral clustering. Statistics and computing, 17(4):395–416, 2007.

[41] Endre Weiszfeld. On the point for which the sum of the distances to n given points is minimum. Tohoku Mathematical Journal, First Series, 43:355–386, 1937.

Appendix A

Notation

Symbol	Definition (equation)
$\mathbb{R}^D$	ambient Euclidean space
$\mathbb{R}^d$	embedding (parameterization) space
$\mathcal{M}$	a smooth manifold embedded in the observation space, $\mathcal{M} \subset \mathbb{R}^D$
Φ	a coordinate chart of a manifold $\mathcal{M}$, $\Phi : N_p \cap \mathcal{M} \to \Omega \subset \mathbb{R}^d$
$\hat{\Phi}$	an approximate global parameterization of a manifold $\mathcal{M}$, $\hat{\Phi} : \mathcal{M} \to \mathbb{R}^d$
Log_x	the logarithm map at $x \in \mathcal{M}$, $\mathrm{Log}_x : N(x) \cap \mathcal{M} \to T_x\mathcal{M}$ (4.6)
Exp_x	the exponential map at $x \in \mathcal{M}$, $\mathrm{Exp}_x : T_x\mathcal{M} \to N(x) \cap \mathcal{M}$
$T_p\mathcal{M}$	the tangent plane of $\mathcal{M}$ at a point $p \in \mathcal{M}$ (2.6)
$d(x, y)$	the manifold distance between $x, y \in \mathcal{M}$, $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ (2.10)
S_{n-1}	the unit sphere in $\mathbb{R}^n$, $S_{n-1} = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$
$B_r(z)$	a ball of radius r centered at z, $B_z(r) = \{x \mid \|x - z\| < r\}$
$X = [x_1 \dots x_n] \in \mathbb{R}^{D \times N}$	the data matrix with columns representing data points
$N_r(x_i)$	A neighborhood of size r centered at x_i, $N_r(x_i) = \{x_j \mid x_j \in B_r(x_i)\}$
$\sigma_k^{i,r}$	the k^{th} singular value of the data matrix corresponding to $N_r(x_i)$
$C_n(s)$	sample correlation sum (3.1)
$U_{n,h}(k)$	kernelized correlation sum (3.3)
$k_h(z)$	scaled kernel (3.4)
$U_{n,h}(X, Y)$	two sample kernelized correlation sum (3.8)
$C_p(m)$	the geometric p-mean cost function (4.1)

Table A.1: Notation.

Symbol	Definition (equation)
$m^{(k)}$	the k^{th} iteration of the Weiszfeld algorithm (4.3), (4.5)
$N_k(x)$	the k nearest neighbors of x (4.8)
$G = (V, E, W)$	a weighted graph with vertex set V, edge set E, and edge weights W
$d_G(v_i, v_j)$	the shortest path on G between vertices v_i and v_j
L	the combinatorial graph Laplacian, $L = D - W$
$\hat{L}$	the normalized graph Laplacian, $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ (4.15)
$M(S)$	the estimated geometric median of the set S
m_{triv}	a trivial estimator for comparison (4.17)
$m^\star$	the ground truth median
μ	the residual minimizing trajectory of epileptogenesis, $\mu : [0,1] \to \mathcal{M}$
τ_i	the residual minimizing biological clock of test animal i, $\tau_i : [0,1] \to [0,1]$

Table A.2: Notation (cont).