THE UNIVERSITY OF CHICAGO

PROBLEMS OF LEARNING ON MANIFOLDS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF MATHEMATICS

BY
MIKHAIL BELKIN

CHICAGO, ILLINOIS
AUGUST 2003

To my parents and grandparents.

ABSTRACT

This thesis discusses the general problem of learning a function on a manifold given by data points. The space of functions on a Riemannian manifold has a family of smoothness functionals and a canonical basis associated to the Laplace-Beltrami operator. Moreover, the Laplace-Beltrami operator can be reconstructed with certain convergence guarantees when the manifold is only known through the sampled data points. This allows the techniques of regularization and Fourier analysis to be applied to functions defined on data. A convergence result is proved for the case when data is sampled from a compact submanifold of $\mathbb{R}^k$. Several applications are considered.

ACKNOWLEDGMENTS

I would like to thank my adviser Partha Niyogi for his help and thoughtful advice. I am very grateful to John Goldsmith for motivating me to think about ideas which eventually lead to this thesis and to Jack Cowan for his support.

I would like to thank my fellow students Matt Cushman, Ben Blander and Kaj Gartz for conversations. I am especially grateful to Joshua Maher for many discussions and insightful suggestions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

An amazing property of human intelligence is the ability to learn well from relatively small amounts of data. By the age of three a child is already a highly proficient speaker, an achievement which is still very far from being replicated by computer models. Understanding the process of learning is no doubt one of the most intriguing problems in science. The goal of theoretical machine learning is to develop this understanding by using mathematical models. Ideally, that would lead to efficient computer algorithms to simulate natural intelligence and learning.

One of the main difficulties in machine learning is how to cope with the seemingly very high-dimensional space of stimuli. For example, even the space of small 100×100 gray scale images is already of dimension 10000. However, a human can distinguish a face from a car in a matter of milliseconds. Thus, one is lead to suspect that the true intrinsic dimensionality of images may be much lower. Another example is speech - while typical representations of speech signals are based on windowed Fourier transform and are high-dimensional, the speech signal itself is generated by the vocal tract, which has limited degrees of freedom.

However, most methods of machine learning operate on the ambient space, rather than trying to reconstruct the smaller space of the "true" data. With the development of freely available electronic data sets, plentiful data has become easily accessible for research. One might hope that it is possible to exploit some of that data to "chart" the space of data itself.

This thesis starts with the assumption that the data lie on or around a low-dimensional manifold in a (potentially) very high-dimensional space. This submanifold is unknown except for finitely many points sampled from some probability distribution. We then show that many problems of machine learning can be naturally

approached in this context. Finally some theoretical guarantees including a proof of convergence are provided.

The main problems of machine learning traditionally include classification, data representation and clustering. In the following sections we briefly describe each of these and outline methods motivating by the manifold assumption.

1.1 Classification/Regression

Given a space X and a probability distribution τ on some space $X \times \mathbb{R}$, the regression problem is to reconstruct the function $\bar{f} : X \to \mathbb{R}$, $\bar{f}(\mathbf{x}) = \mathrm{E}_{\tau_{\mathbf{x}}}(y)$, where $\tau_{\mathbf{x}}$ is the marginal distribution on $\mathbb{R}$. In the simplest case the probability distribution on $X \times \mathbb{R}$ is just a distribution on X and a fixed function $f : X \to \mathbb{R}$. We are typically given a finite number of samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$. The reconstructed function should not only describe the examples well, but also should generalize well. That is, given a new data point, the probability of making a large error should be small.

Classification can be thought as a special case of regression, where the goal is to reconstruct a function from X into a finite set. Often that finite set only contains two elements, which can be conveniently represented by 1 and -1. That is of course equivalent to partitioning the space X into two classes.

In general these problems and most other problems of machine learning are not (and cannot be) well-posed. One needs assumptions on the space of the possible functions f to obtain reasonable results. The simplest possible assumption is to assume that f is a linear function on $X = \mathbb{R}^k$.

Historically, one of the first learning algorithms for classification was the Perceptron introduced by Rosenblatt in 1957. The Perceptron is a linear learning algorithm. Rosenblatt also showed convergence. If a linear solution for the classification problem exists, then the Perceptron will find it, given enough data. Another simple linear classification/regression algorithm is least squares, where a hyperplane is chosen to minimize the squared error term. It can be easily seen that the least squares algorithm is not guaranteed to produce the correct result even given an infinite amount of data.

A more general approach to regression/classification is an interpretation of the Occam's razor principle. One wants a balance between how well a model (i.e. the regressor function) explains the data and how complicated that model is. Of course there is always a model which fits the data exactly, however such models are not likely to generalize, that is give correct predictions for unseen data.

There are many ways to define complexity for functions. An important notion of complexity, the so-called VC-dimension, was introduced in 1972 by Vapnik and Chervonenkis. The definition is purely combinatorial. The VC-dimension of a class of functions is the maximum cardinality, such that any subset of some set with that number of points can be separated from its complement using functions from the class. The Empirical Risk Minimization framework, proposed by Vapnik and Chervonenkis suggests that the best generalization is provided by classes with low VC-dimension.

Another important framework is the Minimum Description Length, introduced by Rissanen, which is a computable version of Kolmogorov complexity, where the best model is one given by the shortest model in a appropriate language.

Recently much work in machine learning has been dedicated to the kernel methods (directly related to splines in statistics, e.g. Wahba, 1990). The approach is based on properties of Reproducing Kernel Hilbert spaces. It is known in statistics as splines and in machine learning as kernel methods. Let $\mathcal{H}$ be a Hilbert space of functions on some space X. Then $\mathcal{H}$ is a RKHS if the evaluation functionals $E_{\mathbf{x}} : f \to f(\mathbf{x})$ are continuous in the topology of $\mathcal{H}$. In the regression setting one often wishes to optimize the following functional (a form of Tikhonov regularization)

$$\frac{1}{n}\sum_i (f(\mathbf{x}_i) - y_i)^2 + \lambda \| f \|^2_{\mathcal{H}}$$

The norm in the Reproducing Kernel Hilbert Space can be thought of as a measure of complexity of a function f. It turns out that such spaces can be associated to positive definite kernel functions $K(\mathbf{x}, \mathbf{y}) : X \to \mathbb{R}$. Remarkably the solution to the regularization problem can be represented in the form $h(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x})$. That reduces the variational problem in a Hilbert space to solving a system of n linear equations for the coefficients α_i. This is a fairly simple problem from the

computational point of view and leads to number of efficient algorithms for regression and classification. For an overview of the mathematical theory of kernel methods and the related issues see Cucker and Smale, 2002.

The data space X can be quite general. In particular, it is natural to consider the case when X is a manifold or a submanifold in some high-dimensional Euclidean space with the induced metric. However, as is often the case in machine learning, the exact nature of that submanifold is not known. This thesis suggests a method for reconstructing this submanifold from the data itself by constructing the adjacency graph for the data and proposes methods for regression using that reconstructed structure. More precisely, a graph is constructed by connecting adjacent data points. The eigenfunctions of the graph Laplacian then provide a natural basis for functions on the graph. Convergence of graph Laplacian to the manifold Laplacian is demonstrated. It turns out that the heat equation on the manifold suggests the choice of Gaussian weights for the graph. This follows from the fact that the heat kernel on a manifold is approximately Gaussian for small values of t.

1.2 Clustering

Clustering is an important problem in machine learning. Together with data representation it can be classified as a problem of unsupervised learning. The goal is to separate objects, e.g. points in $\mathbb{R}^k$, into "clusters" based on some notion of proximity. Such problems of grouping are usually very easy for humans, while it is often difficult to come up with satisfactory rigorous mathematical definition of what a good clustering might be. There are many methods for clustering that produce varied results when applied to real-world data.

The method most relevant in the context of this thesis is spectral clustering. Suppose we are given a graph G and the goal it to cut this graph in two parts S and $G - S$, so that the weight of the edges between the parts are minimized. One defines the Cheeger constant as

$$h_G = \min_{S \subset G} \frac{\mathrm{cut}(S, G - S)}{\min(\mathrm{vol}(S), \mathrm{vol}(G - S))}$$

Here $\mathrm{vol}(S)$ is the sum of the degrees for all vertices of G and $\mathrm{cut}(S, G - S)$ is the weight of all edges connecting S and its complement. It is clear that the split corresponding to the Cheeger constant provides in a sense optimal "clustering" of the graph. However finding such partitions is in general an NP-hard problem.

Remarkably, it turns out that this quantity can be estimated through the first non-trivial eigenvalue of the graph Laplacian. More precisely if L is the graph Laplacian (normalized Laplacian), then the following Cheeger inequalities hold:

$$\frac{\lambda_1}{2} \leq h_G < \sqrt{2\lambda_1}$$

Here λ_1 is the first nonzero eigenvalue of L. Not only that but the corresponding eigenvector provides the split (e.g., one can split the graph into vertices with positive and negative entries). Of course, computing this eigenvector is computationally far simpler than solving the combinatorial problem.

It is interesting to note the parallel to clustering on a manifold. The Cheeger constant for a manifold $\mathcal{M}^n$ is defined as

$$H_G = \min_{S \subset \mathcal{M}} \frac{\mathrm{vol}^{n-1}(\partial S)}{\min(\mathrm{vol}^n(S), \mathrm{vol}^n(\mathcal{M} - S))}$$

The Cheeger constant for $\mathcal{M}$ is produced by the solution to the isoperimetric problem, i.e. the problem of finding a submanifold with the smallest ratio of boundary area to volume. Cheeger's original insight (Cheeger, 1970) is essentially that the first eigenfunction of the Laplacian on a manifold provides a good clustering.

1.3 Data Representation/Dimensionality Reduction

In many practical problems the data is given by points in a very high-dimensional space. For example, gray scale images naturally reside in space whose dimensionality is the number of pixels. Thus any information retrieval algorithm for images has to deal with spaces of dimension potentially exceeding 10000 or even greater, which can be unfeasible. Thus it is useful to be able to reduce the dimensionality of the space without losing too much information. In many applications it is also

useful to provide interesting visualizations for the data. An archetypical data representation/dimensionality reduction method is the Principal Components Analysis (Karhunen-Loeve transform). Given data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, Principal Components Analysis attempts to find a linear space of a given dimension k that comes as close to the data as possible. The solution is given by the space of eigenfunctions of $X^t X$, where X is the $k \times n$ matrix of data points (assuming the data has mean 0. It turns out that if the points are generated by a multivariate Gaussian distribution then Principal Components Analysis is equivalent to decorrelating the data and removing directions with the smallest variance. Thus PCA can be though of as a method for denoising the data, which is the way it is used in many engineering applications. A closely related method is Multidimensional Scaling, which attempts to preserve the distances between points.

Recently several algorithms have been propose that attempt to take advantage of the potential manifold structure in the data (Roweis and Saul, 2001, Tenenbaum, et al, 2001). Tenenbaum's Isomap attempts to preserve the distances between neighbors and can be shown to converge asymptotically if the manifold is flat. Roweis and Saul's algorithm is less theoretically transparent. We propose an algorithm for data representation with certain optimality properties and show that the algorithm Roweis and Saul is closely related to calculating the eigenfunctions of the Laplacian.

1.4 Contribution of This Thesis

The main contribution of this thesis is a unified framework for approaching problems of machine learning on manifolds known only through sampled data points. Given data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we construct the adjacency G graph by putting edges between data points, which are close in some appropriate sense. Eigenfunctions of the graph Laplacian then provide a natural basis for functions on G. Algorithms for partially labeled classification, clustering and data representations are introduced. The role of the Laplace-Beltrami operator in the heat equation and the connection to the heat kernels is used to derive the appropriate weights for edges of G and to show asymptotic convergence under the assumption that the data is drawn from a submanifold of $\mathbb{R}^k$.

Chapter 2 discusses algorithms for data representation and optimality properties of the graph Laplacian. The connection to the heat equation is used to derive the form of the weights. We discuss the relation to spectral clustering and also provide a theoretical analysis of the popular recent Locally Linear Embedding algorithm (Roweis and Saul, 2001).

Chapter 3 deals with the problem of classification and regression and in particular partially labeled classification. In many practical settings one has an abundance of data, while the actual labeling has to be done manually and is quite expensive and time-consuming. We provide a theoretically principled non-parametric method for approaching the problem. Some applications and promising practical results on several datasets are discussed.

Chapter 4 discusses convergence issues. An asymptotic convergence result is proved for the case when the data is sampled from a probability distribution on a compact submanifold of $\mathbb{R}^k$ and the amount of data tends to infinity.

1.5 Collaborations and Prior Publications

Chapters 2 and 3 of this thesis are based on joint publications with Partha Niyogi (Belkin, Niyogi, 2002, 2003).

CHAPTER 2
DATA REPRESENTATION

In many areas of artificial intelligence, information retrieval and data mining, one is often confronted with intrinsically low dimensional data lying in a very high dimensional space. Consider, for example, gray scale images of an object taken under fixed lighting conditions with a moving camera. Each such image would typically be represented by a brightness value at each pixel. If there were n^2 pixels in all (corresponding to an $n \times n$ image), then each image yields a data point in $\mathbb{R}^{n^2}$. However, the intrinsic dimensionality of the space of all images of the same object is the number of degrees of freedom of the camera. In this case, the space under consideration has the natural structure of a low dimensional manifold embedded in $\mathbb{R}^{n^2}$.

Recently, there has been some renewed interest (Tenenbaum et al, 2000; Roweis and Saul, 2000) in the problem of developing low dimensional representations when data arises from sampling a probability distribution on a manifold. This thesis presents a geometrically motivated algorithm and an accompanying framework of analysis for this problem.

The general problem of dimensionality reduction has a long history. Classical approaches include Principal Components Analysis and Multidimensional Scaling. Various methods that generate nonlinear maps have also been considered. Most of them, such as self-organizing maps and other neural network based approaches, e.g., see Haykin (1999), set up a nonlinear optimization problem whose solution is typically obtained by gradient descent that is only guaranteed to produce a local optimum — global optima are difficult to attain by efficient means. Note however, that the recent approach of generalizing the PCA through kernel based techniques (Schoelkopf et al, 1998) does not have this shortcoming. Most of these methods do not explicitly consider the structure of the manifold on which the data may possibly reside.

In this thesis, we explore an approach that builds a graph incorporating neighborhood information of the data set. Using the notion of the Laplacian of the graph, we then compute a low dimensional representation of the data set that optimally preserves local neighborhood information in a certain sense. The representation map generated by the algorithm may be viewed as a discrete approximation to a continuous map that naturally arises from the geometry of the manifold.

It is worthwhile to highlight several aspects of the algorithm and the framework of analysis presented here.

1. The core algorithm is very simple. It has a few local computations and one sparse eigenvalue problem. The solution reflects the intrinsic geometric structure of the manifold. It does, however, require a search for neighboring points in a high dimensional space. We note that there are several efficient approximate techniques for finding nearest neighbors, e.g. see Indyk (2000).

2. The justification for the algorithm comes from the role of the Laplace-Beltrami operator in providing an optimal embedding for the manifold. The manifold is approximated by the adjacency graph computed from the data points. The Laplace-Beltrami operator is approximated by the weighted Laplacian of the adjacency graph with weights chosen appropriately. The key role of the Laplace Beltrami operator in the heat equation enables us to use the heat kernel to choose the weight decay function in a principled manner. Thus, the embedding maps for the data approximate the eigenmaps of the Laplace-Beltrami operator which are maps intrinsically defined on the entire manifold.

3. The framework of analysis presented here makes explicit use of these connections to interpret dimensionality reduction algorithms in a geometric fashion. In addition to the algorithms presented in this thesis, we are also able to reinterpret the recently proposed Locally Linear Embedding (LLE) algorithm of Roweis and Saul (2000) within this framework.

   The graph Laplacian has been widely used for different clustering and partition problems (e.g., Shi and Malik, 2000, Simon, 1991, Ng et al 2002). On the other

hand while the connections between the Laplace Beltrami operator and the graph Laplacian are well known to geometers and specialists in spectral graph theory (see Chung, 1997; Chung, Grigoryan and Yau, 1997) so far we are not aware of any application to dimensionality reduction or data representation.

We note, however, recent work on using diffusion kernels on graphs and other discrete structures, Kondor and Lafferty (2002).

4. The locality preserving character of the Laplacian Eigenmap algorithm makes it relatively insensitive to outliers and noise. It is also not prone to "short circuiting" as only the local distances are used.

   We show that, in fact, by trying to preserve local information in the embedding, the algorithm implicitly emphasizes the natural clusters in the data. Close connections to spectral clustering algorithms developed in learning and computer vision (in particular, the approach of Shi and Malik, 1997) then become very clear. In this sense, dimensionality reduction and clustering are two sides of the same coin and we explore this connection in some detail. In contrast, global methods like that in Tenenbaum et al, 2000, do not show any tendency to cluster as an attempt is made to preserve all pairwise geodesic distances between points.

   However not all data sets necessarily have meaningful clusters. Other methods such as PCA or Isomap might be more appropriate in that case. We will demonstrate, however, that at least in one example of such a data set (the "swiss roll"), our method produces reasonable results.

5. Since much of the discussion of Seung and Lee, 2000, Roweis and Saul, 2000, and Tenenbaum et al, 2000 is motivated by the role that non-linear dimensionality reduction may possibly play in human perception and learning, it is worthwhile to consider the implication of the previous remark in this context. The biological perceptual apparatus is confronted with high dimensional stimuli from which it must recover low dimensional structure. If the approach to recovering such low-dimensional structure is inherently local (for example, as in the algorithm

proposed here), then a natural clustering will emerge and may serve as the basis for the emergence of categories in biological perception.

6. Since our approach is based on the intrinsic geometric structure of the manifold, it exhibits stability with respect to the embedding. As long as the embedding is isometric, the representation will not change. In the example with the moving camera, different resolutions of the camera (i.e., different choices of n in the $n \times n$ image grid) should lead to embeddings of the same underlying manifold into spaces of very different dimension. Our algorithm will produce similar representations independently of the resolution.

2.1 The Problem of Dimensionality Reduction

The generic problem of dimensionality reduction is the following. Given a set $\mathbf{x}_1, \ldots, \mathbf{x}_k$ of k points in $\mathbb{R}^l$, find a set of points $\mathbf{y}_1, \ldots, \mathbf{y}_k$ in $\mathbb{R}^m$ ($m \ll l$) such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$.

In this thesis, we consider the special case where $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathcal{M}$ and $\mathcal{M}$ is a manifold embedded in $\mathbb{R}^l$.

We now consider an algorithm to construct representative $\mathbf{y}_i$'s for this special case. The sense in which such a representation is optimal will become clear later.

2.2 The Algorithm

Given k points $\mathbf{x}_1, \ldots, \mathbf{x}_k$ in $\mathbb{R}^l$, we construct a weighted graph with k nodes, one for each point, and a set of edges connecting neighboring points. The embedding map is now provided by computing the eigenvectors of the graph Laplacian. The algorithmic procedure is formally stated below.

1. Step 1 [*Constructing the Adjacency Graph*]. We put an edge between nodes i and j if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close". There are two variations:

   (a) ϵ-neighborhoods. [*parameter* $\epsilon \in \mathbb{R}$] Nodes i and j are connected by an edge if $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon$ where the norm is the usual Euclidean norm in $\mathbb{R}^l$.

Advantages: geometrically motivated, the relationship is naturally symmetric.

Disadvantages: often leads to graphs with several connected components, difficult to choose ϵ.

(b) n nearest neighbors. [*parameter* $n \in \mathbb{N}$] Nodes i and j are connected by an edge if i is among n nearest neighbors of j or j is among n nearest neighbors of i. Note that this relation is symmetric.

Advantages: easier to choose, does not tend to lead to disconnected graphs.

Disadvantages: less geometrically intuitive.

2. Step 2.[1] [*Choosing the weights*]. Here, as well, we have two variations for weighting the edges:

(a) Heat kernel. [*parameter* $t \in \mathbb{R}$]. If nodes i and j are connected, put

$$W_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}$$

otherwise put $W_{ij} = 0$. The justification for this choice of weights will be provided later.

(b) Simple-minded. [*No parameters* ($t = \infty$]. $W_{ij} = 1$ if vertices i and j are connected by an edge and $W_{ij} = 0$ if vertices i and j are not connected by an edge.

A simplification which avoids the necessity of choosing t.

3. Step 3. [Eigenmaps] Assume the graph G, constructed above, is connected, otherwise proceed with Step 3 for each connected component.

Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$L\mathbf{f} = \lambda D\mathbf{f} \tag{2.1}$$

1. In a computer implementation of the algorithm steps one and two are executed simultaneously.

where D is a diagonal weight matrix, its entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. $L = D - W$ is the Laplacian matrix. The Laplacian is a symmetric, positive semidefinite matrix which can be thought of as an operator on functions defined on the vertices of G.

Let $\mathbf{f}_0, \ldots, \mathbf{f}_{k-1}$ be the solutions of equation 2.1, ordered according to their eigenvalues,

$$L\mathbf{f}_0 = \lambda_0 D\mathbf{f}_0$$

$$L\mathbf{f}_1 = \lambda_1 D\mathbf{f}_1$$

$$\ldots$$

$$L\mathbf{f}_{k-1} = \lambda_{k-1} D\mathbf{f}_{k-1}$$

$$0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{k-1}$$

We leave out the eigenvector $\mathbf{f}_0$ corresponding to eigenvalue 0 and use the next m eigenvectors for embedding in m-dimensional Euclidean space.

$$\mathbf{x}_i \rightarrow (\mathbf{f}_1(i), \ldots, \mathbf{f}_m(i))$$

2.3 Justification

2.3.1 Optimal Embeddings

Let us first show that the embedding provided by the Laplacian Eigenmap algorithm preserves local information optimally in a certain sense.

The following section is based on the standard spectral graph theory. See Chung (1997) for a comprehensive reference.

Recall that given a data set we construct a weighted graph $G = (V, E)$ with edges connecting nearby points to each other. For the purposes of this discussion, assume the graph is connected. Consider the problem of mapping the weighted graph G to a line so that connected points stay as close together as possible. Let $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ be such a map. A reasonable criterion for choosing a "good" map is

to minimize the following objective function

$$\sum_{ij}(y_i - y_j)^2 W_{ij}$$

under appropriate constraints. The objective function with our choice of weights W_{ij} incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close" then y_i and y_j are close as well.

It turns out that for any $\mathbf{y}$, we have

$$\frac{1}{2}\sum_{i,j}(y_i - y_j)^2 W_{ij} = \mathbf{y}^T L \mathbf{y} \tag{2.2}$$

where as before, $L = D - W$. To see this, notice that W_{ij} is symmetric and $D_{ii} = \sum_j W_{ij}$. Thus

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} = \sum_{i,j}(y_i^2 + y_j^2 - 2y_i y_j) W_{ij} =$$

$$\sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_{i,j} y_i y_j W_{ij} = 2\mathbf{y}^T L \mathbf{y}$$

Note that this calculation also shows that L is positive semidefinite.

Therefore, the minimization problem reduces to finding

$$\underset{\substack{\mathbf{y} \\ \mathbf{y}^T D \mathbf{y} = 1}}{\operatorname{argmin}} \ \mathbf{y}^T L \mathbf{y}$$

The constraint $\mathbf{y}^T D \mathbf{y} = 1$ removes an arbitrary scaling factor in the embedding. Matrix D provides a natural measure on the vertices of the graph. The bigger the value D_{ii} (corresponding to the ith vertex) is, the more "important" is that vertex. It follows from equation 2.2 that L is a positive semidefinite matrix and the vector $\mathbf{y}$ that minimizes the objective function is given by the minimum eigenvalue solution

to the generalized eigenvalue problem

$$L\mathbf{y} = \lambda D\mathbf{y}$$

Let $\mathbf{1}$ be the constant function taking 1 at each vertex. It is easy to see that $\mathbf{1}$ is an eigenvector with eigenvalue 0. If the graph is connected, $\mathbf{1}$ is the only eigenvector for $\lambda = 0$. To eliminate this trivial solution which collapses all vertices of G onto the real number 1, we put an additional constraint of orthogonality and look for

$$\begin{array}{l} \text{argmin } \mathbf{y}^T L\mathbf{y} \\ \mathbf{y}^T D\mathbf{y}=1 \\ \mathbf{y}^T D\mathbf{1}=0 \end{array}$$

Thus, the solution is now given by the eigenvector with the smallest non-zero eigenvalue. The condition $\mathbf{y}^T D\mathbf{1} = 0$ can be interpreted as removing a translation invariance in $\mathbf{y}$.

Now consider the more general problem of embedding the graph into m-dimensional Euclidean space. The embedding is given by the $k \times m$ matrix $Y = [\mathbf{y}_1\mathbf{y}_2 \ldots \mathbf{y}_m]$ where the ith row provides the embedding coordinates of the ith vertex. Similarly we need to minimize

$$\sum_{i,j} \|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|^2 W_{ij} = \text{tr}(Y^T LY)$$

where $\mathbf{y}^{(i)} = [\mathbf{y}_1(i), \ldots, \mathbf{y}_m(i)]^T$ is the m-dimensional representation of the ith vertex. This reduces to finding

$$\begin{array}{l} \text{argmin } \text{tr}(Y^T LY) \\ Y^T DY=I \end{array}$$

For the one-dimensional embedding problem, the constraint prevents collapse onto a point. For the m-dimensional embedding problem, the constraint presented above prevents collapse onto a subspace of dimension less than $m - 1$ (m if, as in one-dimensional case, we require orthogonality to the constant vector). Standard methods show that the solution is provided by the matrix of eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue problem $L\mathbf{y} = \lambda D\mathbf{y}$.

2.3.2 The Laplace-Beltrami Operator

The Laplacian of a graph is analogous to the Laplace-Beltrami operator on manifolds. In this section we provide a justification for why the eigenfunctions of the Laplace-Beltrami operator have properties desirable for embedding.

Let $\mathcal{M}$ be a smooth, compact, m-dimensional Riemannian manifold. If the manifold is embedded in $\mathbb{R}^l$ the Riemannian structure (metric tensor) on the manifold is induced by the standard Riemannian structure on $\mathbb{R}^l$.

As we did with the graph, we are looking here for a map from the manifold to the real line such that points close together on the manifold get mapped close together on the line. Let f be such a map. Assume that $f : \mathcal{M} \to \mathbb{R}$ is twice differentiable.

Consider two neighboring points $\mathbf{x}, \mathbf{z} \in \mathcal{M}$. They are mapped to $f(\mathbf{x})$ and $f(\mathbf{z})$ respectively. We first show that

$$|f(\mathbf{z}) - f(\mathbf{x})| \leq \text{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{z}) \|\nabla f(\mathbf{x})\| + o(\text{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{z})) \tag{2.3}$$

The gradient $\nabla f(x)$ is a vector in the tangent space $T\mathcal{M}_x$, such that given another vector $\mathbf{v} \in T\mathcal{M}_x$, $df(\mathbf{v}) = \langle \nabla f(x), \mathbf{v} \rangle_{\mathcal{M}}$.

Let $l = \text{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{z})$. Let $c(t)$ be the geodesic curve parameterized by length connecting $\mathbf{x} = c(0)$ and $\mathbf{z} = c(l)$. Then

$$f(\mathbf{z}) = f(\mathbf{x}) + \int_0^l df(c'(t))dt = f(\mathbf{x}) + \int_0^l \langle \nabla f(c(t)), c'(t) \rangle dt$$

Now by Schwartz Inequality,

$$\langle \nabla f(c(t)), c'(t) \rangle \leq \|\nabla f(c(t))\| \, \|c'(t)\| = \|\nabla f(c(t))\|$$

Since $c(t)$ is parameterized by length, we have $\|c'(t)\| = 1$. We also have $\|\nabla f(c(t))\| = \|\nabla f(\mathbf{x})\| + O(t)$ (by Taylor's approximation). Finally, by integrating we have

$$|f(\mathbf{z}) - f(\mathbf{x})| \leq l \, \|\nabla f(\mathbf{x})\| + o(l)$$

where both O and o are used in the infinitesimal sense.

If $\mathcal{M}$ is isometrically embedded in $\mathbb{R}^l$ then $\text{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{z}) = \|\mathbf{x}-\mathbf{z}\|_{\mathbb{R}^l} + o\left(\|\mathbf{x}-\mathbf{z}\|_{\mathbb{R}^l}\right)$ and

$$|f(\mathbf{z}) - f(\mathbf{x})| \leq \|\nabla f(\mathbf{x})\| \, \|\mathbf{z} - \mathbf{x}\| + o\left(\|\mathbf{z} - \mathbf{x}\|\right)$$

Thus we see that if $\|\nabla f\|$ provides us with an estimate of how far apart f maps nearby points.

We therefore look for a map that best preserves locality on average by trying to find

$$\underset{\|f\|_{L^2(\mathcal{M})}=1}{\text{argmin}} \int_{\mathcal{M}} \|\nabla f(x)\|^2 \tag{2.4}$$

where the integral is taken with respect to the standard measure on a Riemannian manifold. Note that minimizing $\int\limits_{\mathcal{M}} \|\nabla f(x)\|^2$ corresponds to minimizing $L\mathbf{f} = \frac{1}{2}\sum_{i,j}(f_i - f_j)^2 W_{ij}$ on a graph. Here $\mathbf{f}$ is a function on vertices and f_i is the value of $\mathbf{f}$ on the ith node of the graph.

It turns out that minimizing the objective function of eq. 2.4 reduces to finding eigenfunctions of the Laplace Beltrami operator $\mathcal{L}$. Recall that

$$\mathcal{L}f \stackrel{def}{=} -\operatorname{div} \nabla(f)$$

where div is the divergence of the vector field. It follows from the Stokes' theorem that $-$div and ∇ are formally adjoint operators, i.e. if f is a function and $\mathbf{X}$ is a vector field then $\int_{\mathcal{M}} \langle \mathbf{X}, \nabla f \rangle = -\int_{\mathcal{M}} \operatorname{div}(\mathbf{X}) f$. Thus

$$\int_{\mathcal{M}} \|\nabla f\|^2 = \int_{\mathcal{M}} \mathcal{L}(f) f$$

We see that $\mathcal{L}$ is positive semidefinite. The f that minimizes $\int_{\mathcal{M}} \|\nabla f\|^2$ has to be an eigenfunction of $\mathcal{L}$. The spectrum of $\mathcal{L}$ on a compact manifold $\mathcal{M}$ is known to be discrete (e.g., Rosenberg, 1997). Let the eigenvalues (in increasing order) be $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots$ and let f_i be the eigenfunction corresponding to eigenvalue λ_i. It is easily seen that f_0 is the constant function that maps the entire manifold

to a single point. To avoid this eventuality, we require (just as in the graph setting) that the embedding map f be orthogonal to f_0. It immediately follows that f_1 is the optimal embedding map. Following the arguments of the previous section, we see that

$$\mathbf{x} \to (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$$

provides the optimal m-dimensional embedding.

2.3.3 Heat Kernels and the Choice of Weight Matrix

The Laplace-Beltrami operator on differentiable functions on a manifold $\mathcal{M}$ is intimately related to the heat flow. Let $f : \mathcal{M} \to \mathbb{R}$ be the initial heat distribution, $u(x,t)$ be the heat distribution at time t ($u(x,0) = f(x)$). The heat equation is the partial differential equation $(\frac{\partial}{\partial t} + \mathcal{L})u = 0$. The solution is given by $u(x,t) = \int_{\mathcal{M}} H_t(x,y)f(y)$, where H_t is the heat kernel — the Green's function for this partial differential equation. Therefore,

$$\mathcal{L}f(x) = -\mathcal{L}u(x,0) = -\left(\frac{\partial}{\partial t}\left[\int_{\mathcal{M}} H_t(x,y)f(y)\right]\right)_{t=0} \tag{2.5}$$

It turns out that in an appropriate coordinate system (exponential, which to the first order coincides with the local coordinate system given by a tangent plane in $\mathbb{R}^l$) H_t is approximately the Gaussian.

$$H_t(x,y) = (4\pi t)^{-\frac{m}{2}} e^{-\frac{\|x-y\|^2}{4t}} (\phi(x,y) + O(t))$$

where $\phi(x,y)$ is a smooth function with $\phi(x,x) = 1$. Therefore when x and y are close and t is small

$$H_t(x,y) \approx (4\pi t)^{-\frac{m}{2}} e^{-\frac{\|x-y\|^2}{4t}}$$

See Rosenberg (1997) for more details.

Notice that as t tends to 0, the heat kernel $H_t(x,y)$ becomes increasingly localized and tends to Dirac's δ-function, i.e., $\lim_{t\to 0} \int_{\mathcal{M}} H_t(x,y)f(y) = f(x)$. Therefore, for small

t from the definition of the derivative we have

$$\mathcal{L}f(x) \approx \frac{1}{t}\left[f(x) - (4\pi t)^{-\frac{m}{2}} \int_{\mathcal{M}} e^{-\frac{\|x-y\|^2}{4t}} f(y)dy\right]$$

If $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are data points on $\mathcal{M}$, the last expression can be approximated by

$$\mathcal{L}f(\mathbf{x}_i) \approx \frac{1}{t}\left[f(\mathbf{x}_i) - \frac{1}{k}(4\pi t)^{-\frac{m}{2}} \sum_{\substack{\mathbf{x}_j \\ 0<\|\mathbf{x}_j-\mathbf{x}_i\|<\epsilon}} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}} f(\mathbf{x}_j)\right]$$

The coefficient $\frac{1}{t}$ is global and will not affect the eigenvectors of the discrete Laplacian. Since the inherent dimensionality of $\mathcal{M}$ may be unknown, we put $\alpha = \frac{1}{k}(4\pi t)^{-\frac{m}{2}}$. It is interesting to note that since the Laplacian of the constant function is zero, it immediately follows that $\frac{1}{\alpha} = \sum_{\substack{\mathbf{x}_j \\ 0<\|\mathbf{x}_j-\mathbf{x}_i\|<\epsilon}} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}}$ and

$$\alpha = \left(\sum_{\substack{\mathbf{x}_j \\ 0<\|\mathbf{x}_j-\mathbf{x}_i\|<\epsilon}} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}}\right)^{-1}$$

This observation leads to several possible approximation schemes for the manifold Laplacian. In order to ensure that the approximation matrix is positive semidefinite, we compute the graph Laplacian with the following weights:

$$W_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}} & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

2.4 Connections to Spectral Clustering

The approach to dimensionality reduction considered in this thesis utilizes maps provided by the eigenvectors of the graph Laplacian and eigenfunctions of the Laplace-

Beltrami operator on the manifold. Interestingly, this solution may also be interpreted in the framework of clustering and has very close ties to spectrally based clustering techniques such as those used for image segmentation (Shi, Malik, 1997), load balancing (Hendrickson, Leland, 1993), and circuit design (Hadley et al, 1992). A closely related algorithm for clustering has been recently proposed in Ng, et al, 2002. The approach considered there uses a graph which is globally connected with exponentially decaying weights. The decay parameter then becomes very important. In many high dimensional problems the minimum and the maximum distances between points are fairly close in which case the weight matrix will be essentially non-sparse for any rate of decay.

Here we briefly outline the ideas of spectral clustering It is often of interest to cluster a set of n objects into a finite number of clusters. Thus, given a set of n objects (visual, perceptual, linguistic or otherwise), one may introduce a matrix of pair wise similarities between the n objects. It is then possible to formulate a general graph-theoretic framework for clustering as follows. Let $G = (V, E)$ be a weighted graph, W is the matrix of weights, where the vertices are numbered arbitrarily. The weight W_{ij} associated with the edge e_{ij} is the similarity between v_i and v_j. We assume that the matrix of pairwise similarities is symmetric and the corresponding undirected graph is connected.[2]

Let us consider clustering the objects into two classes. Therefore, we wish to divide V into two disjoint subsets A, B, $A \cup B = V$, so that the "flow" between A and B is minimized. The "flow" is a measure of similarity between the two clusters and the simplest definition of the "flow" or "cut" between A and B is the total weight of the edges that have to be removed to make A and B disjoint.

$$\mathrm{cut}(A, B) = \sum_{u \in A, v \in B} W(u, v)$$

Trying to minimize the $\mathrm{cut}(A, B)$ will favor cutting off weakly connected outliers

2. If the graph is not connected, there are many algorithms for finding its connected components.

which tends to lead to poor partitioning quality. To avoid that problem a measure on the set of vertices is introduced. The weight of a vertex is its "importance" relative to other vertices.

$$\mathrm{vol}(A) = \sum_{u \in A, v \in V} W(u, v)$$

where $W(u, v)$ is the weight on the edge between u and v.

Shi and Malik (1997), define the normalized cut

$$\mathrm{Ncut}(A, B) = \mathrm{cut}(A, B) \left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)} \right)$$

The problem, as formulated by Shi and Malik (1997), is to minimize Ncut over all partitions of the vertex set V. A similar and, perhaps, more geometrically motivated quantity is the Cheeger constant

$$h_G = \min_{A \subset V} \frac{\mathrm{cut}(A, \bar{A})}{\min(\mathrm{vol}(A), \mathrm{vol}((\bar{A})))}$$

where $\bar{A}$ is the complement of A in V.[3]

It turns out the the combinatorial optimization problem as stated is NP-hard.[4] However, if we allow relaxation of the indicator functions to real values, the problem reduces to minimizing the Laplacian of the graph, which can be easily computed in polynomial time with arbitrary precision.

Recall that

$$\mathbf{x}^T L \mathbf{x} = \sum_{i,j} (x_i - x_j)^2 w_{ij}$$

Let, as above, A, B be disjoint subsets of V, $A \cup B = V$, and $a = \mathrm{vol}(A)$, $b = \mathrm{vol}(B)$.

---

3. Cheeger's original observation (Cheeger, 1970), was essentially that the first non-constant eigenfunction of the Laplace-Beltrami operator on a manifold provides a good "clustering" of the manifold. He then used a geometric invariant of the manifold, the Cheeger's constant (as it is referred to now) to give a bound for an analytic invariant λ_1.

4. A proof due to Papadimitrou can be found in Shi and Malik (1997).

Put

$$x_i = \begin{cases} \frac{1}{\text{vol}(A)}, \text{if } V_i \in A \\ -\frac{1}{\text{vol}(B)}, \text{if } V_i \in B \end{cases}$$

We have

$$\mathbf{x}^T L\mathbf{x} = \sum_{i,j}(x_i - x_j)^2 w_{ij} = \sum_{V_i \in A, V_j \in B} (\frac{1}{a} + \frac{1}{b})^2 cut(A,B)$$

Also

$$\mathbf{x}^T D\mathbf{x} = \sum_i x_i^2 d_{ii} = \sum_{V_i \in A} \frac{1}{a^2} d_{ii} + \sum_{V_i \in B} \frac{1}{b^2} d_{ii} = \frac{1}{a^2} vol(A) + \frac{1}{b^2} vol(B) = \frac{1}{a} + \frac{1}{b}$$

Thus

$$\frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T D\mathbf{x}} = \text{cut}(A,B)(\frac{1}{a} + \frac{1}{b}) = \text{Ncut}(A,B)$$

Notice that $\mathbf{x}^T D\mathbf{1} = \mathbf{0}$, where $\mathbf{1}$ is a column vector of ones.

The relaxed problem is to minimize $\frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T D\mathbf{x}}$ under the condition that $\mathbf{x}^T D\mathbf{1} = \mathbf{0}$. Put $\mathbf{y} = D^{1/2}\mathbf{x}$. D is invertible, assuming G has no isolated vertices. Then

$$\frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T D\mathbf{x}} = \frac{\mathbf{y}^T D^{-1/2} L D^{-1/2}\mathbf{y}}{\mathbf{y}^T\mathbf{y}}$$

where $\mathbf{x} \perp D^{1/2}\mathbf{1}$.

The matrix $\tilde{L} = D^{-1/2} L D^{-1/2}$ is the so-called normalized graph Laplacian. $\tilde{L}$ is symmetric positive semidefinite. Notice that $D^{1/2}\mathbf{1}$ is an eigenvector for $\tilde{L}$ with eigenvalue 0, which is the smallest eigenvalue of $\tilde{L}$. Thus $\min_{\mathbf{y} \perp D^{1/2}\mathbf{1}} \frac{\mathbf{y}^T \tilde{L}\mathbf{y}}{\mathbf{y}^T\mathbf{y}}$ is achieved when $\mathbf{y}$ is an eigenvector corresponding to the second smallest eigenvalue of $\tilde{L}$. Of course, zero can be a multiple eigenvalue which happens if and only if G has more than one connected component.

Remark: The central observation to be made here is that the process of dimensionality reduction that preserves locality yields the same solution as clustering. It is worthwhile to compare the global algorithm presented in Tenenbaum et al, 2000 with the local algorithms suggested here and in Roweis and Saul, 2000. One approach to non-linear dimensionality reduction as exemplified by Tenenbaum et al attempts to

faithfully approximate all geodesic distances on the manifold. This may be viewed as a global strategy. In contrast, the local approach presented here (as well as that presented in Roweis and Saul, 2000) attempts only to approximate or preserve neighborhood information. This, as we see from the preceding discussion may also be interpreted as imposing a soft clustering of the data (which may be converted to a hard clustering by a variety of heuristic techniques). In this sense the local approach to dimensionality reduction imposes a natural clustering of the data.

2.5 Analysis of Locally Linear Embedding Algorithm

We provide a brief analysis of the Locally Linear Embedding (LLE) algorithm recently proposed by Roweis and Saul, 2000, and exhibit its connection to the Laplacian.

Here is a brief description of their algorithm. As before, one is given a data set $\mathbf{x}_1, \ldots, \mathbf{x}_k$ in a high-dimensional space $\mathbb{R}^l$. The goal is to find a low-dimensional representation $\mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{R}^m$, $m \ll k$.

1. Step 1. [Discovering the Adjacency Information] For each $\mathbf{x}_i$ find the its n nearest neighbors in the dataset, $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_n}$. Alternatively $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_n}$ could be data points contained in an ϵ-ball around $\mathbf{x}_i$.

2. Step 2. [Constructing the Approximation Matrix] Let W_{ij} be such that $\sum_j W_{ij}\mathbf{x}_{i_j}$ equals to the orthogonal projection of $\mathbf{x}_i$ onto the affine linear span of $\mathbf{x}_{i_j}$'s. In other words, one chooses W_{ij} by minimizing

$$\sum_{i=1}^{l} \|\mathbf{x}_i - \sum_{j=1}^{n} W_{ij}\mathbf{x}_{i_j}\|^2$$

   under the condition that $\sum_j W_{ij} = 1$ for each i. Assume that W_{ij}'s are well-determined. If it is not, as it happens for example in the case when $n > k + 1$, the authors propose a heuristic which we will not analyze here.

3. Step 3. [Computing the Embedding] Compute the embedding by taking eigenvectors corresponding to the k lowest eigenvalues of the matrix

$$E = (I - W)^T (I - W)$$

Notice that E is a symmetric positive semidefinite matrix.

E can be thought of as an operator acting on functions defined on the data points. We will now provide an argument that under certain conditions

$$Ef \approx \frac{1}{2}\mathcal{L}^2 f$$

Eigenvectors of $\frac{1}{2}\mathcal{L}^2$, of course, coincide with the eigenvectors of $\mathcal{L}$. We develop this argument over several steps.

Step 1:

Let us fix a data point $\mathbf{x}_i$. We now show that

$$[(I - W)f]_i \approx -\frac{1}{2}\sum_j W_{ij}(\mathbf{x}_i - \mathbf{x}_{i_j})^T H(\mathbf{x}_i - \mathbf{x}_{i_j})$$

where f is a function on the manifold (and therefore defined on the data points), H is the Hessian of f at $\mathbf{x}_i$. To simplify the analysis, the neighboring points ($\mathbf{x}_{i_j}$'s) are assumed to lie on a locally linear patch on the manifold around $\mathbf{x}_i$.

Consider now a coordinate system in the tangent plane centered at $\mathbf{o} = \mathbf{x}_i$. Let $\mathbf{v}_j = \mathbf{x}_{i_j} - \mathbf{x}_i$. Since the difference of two points can be regarded as a vector with the origin at the second point, we see that $\mathbf{v}_j$'s are vectors in the tangent plane originating at $\mathbf{o}$. Let $\alpha_j = W_{ij}$. Since $\mathbf{x}_i$ belongs to the affine span of its neighbors and by construction of the matrix W, we have

$$\mathbf{o} = \mathbf{x}_i = \sum_j \alpha_j \mathbf{v}_j$$

where

$$\sum \alpha_j = 1$$

If f is a smooth function, its second-order Taylor approximation can be written as

$$f(\mathbf{v}) = f(\mathbf{o}) + \mathbf{v}^T \nabla f + \frac{1}{2}(\mathbf{v}^T H \mathbf{v}) + o(\|\mathbf{v}\|^2)$$

Here $\nabla f = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n})^T$ is the gradient and H is the Hessian, $H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$ (both evaluated at $\mathbf{o}$). Therefore

$$[(I - W)f]_i = f(\mathbf{o}) - \sum_j \alpha_j f(\mathbf{v}_j)$$

and using the Taylor approximation for $f(\mathbf{v}_j)$ we have

$$f(\mathbf{o}) - \sum_j \alpha_j f(\mathbf{v}_j) \approx f(\mathbf{o}) - \sum_j \alpha_j f(\mathbf{o}) - \sum_j \alpha_j \mathbf{v}_j^T \nabla f - \frac{1}{2} \sum_j \alpha_j \mathbf{v}_j^T H \mathbf{v}_j$$

Since $\sum \alpha_j = 1$ and $\sum \alpha_j \mathbf{v}_j = \mathbf{o}$, we see that the first three terms disappear and

$$f(\mathbf{o}) - \sum_j \alpha_j f(\mathbf{v_j}) \approx -\frac{1}{2} \sum \alpha_j \mathbf{v}_j^T H \mathbf{v^j}$$

Step 2:

Now note that if $\sqrt{\alpha_i}\ \mathbf{v_i}$ form an orthonormal basis (which, of course, is not usually the case) then

$$\sum_j W_{ij} \mathbf{v}_j^T H \mathbf{v}_j = \mathrm{tr}(H) = \mathcal{L} f$$

More generally, we observe that if $\mathbf{x}$ is a random vector, such that its distribution is uniform on every sphere centered at $\mathbf{x}_i$ (which is true, for example, for any locally uniform measure on the manifold) then the expectation $\mathrm{E}(\mathbf{v}^T H \mathbf{v})$ is proportional to $\mathrm{tr}H$.

Indeed if $\mathbf{e}_1, \ldots, \mathbf{e}_n$ form an orthonormal basis for H corresponding to the eigen-

values $\lambda_1, \ldots, \lambda_n$, then using the Spectral theorem,

$$\mathrm{E}(\mathbf{v}^T H \mathbf{v}) = \mathrm{E}(\sum \lambda_i \langle \mathbf{v}, \mathbf{e}_i \rangle^2)$$

But since $\mathrm{E}\langle \mathbf{v}, \mathbf{e}_i \rangle^2$ is independent of i, put $\mathrm{E}\langle \mathbf{v}, \mathbf{e}_i \rangle^2 = r$ and the above expression reduces to

$$\mathrm{E}(\mathbf{v}^T H \mathbf{v}) = r(\sum_i \lambda_i) = r\mathrm{tr}(H) = r\mathcal{L}f$$

Step 3:

Putting steps 1 and 2 together, we see that

$$(I - W)^T (I - W) f \approx \frac{1}{2}\mathcal{L}^2 f$$

LLE attempts to minimize $f^T (I - W)^T (I - W) f$ which reduces to finding the eigenfunctions of $(I - W)^T (I - W)$ which in turn can now be interpreted as trying to find the eigenfunctions of the iterated Laplacian $\mathcal{L}^2$. Eigenfunctions of $\mathcal{L}^2$ coincide with those of $\mathcal{L}$.

2.6 Examples

We now briefly consider several possible applications of the algorithmic framework developed in this thesis. We begin with a simple synthetic example of a "swiss roll" considered in Tenenbaum et al , 2000, and Roweis and Saul, 2000. We then consider a toy example from vision with vertical and horizontal bars in a "visual field". We conclude with some low dimensional representations constructed from naturally occurring data sets in the domains of speech and language respectively.

We do not yet know of a principled way to choose the heat kernel parameter t. However, we conduct experiments on the "swiss roll" data set to demonstrate the effect of t and number of nearest neighbors N on the low dimensional representation. It is clear that for very large values of N it is critical to choose t correctly. It seems that choosing a smaller t tends to improve the quality of the representation for bigger but

still relatively small N. For small values of N the results do not seem to significantly depend on t.

In the rest of our experiments, we use the simplest version of the algorithm, $W_{ij} \in \{0, 1\}$ or $t = \infty$ which seems to work well in practice and does not involve a choice of a real-valued parameter.

2.6.1 A Synthetic "swiss roll"

The data set of 2000 points chosen at random from the "swiss roll" is shown in fig. A.2. The "swiss roll" is a flat two dimensional submanifold of $\mathbb{R}^3$. Two dimensional representations of the "swiss roll" for different values of parameters N and t are shown in fig. A.3. Note that $t = \infty$ corresponds to the case when the weights are set to 1. Unlike Isomap, our algorithm does not attempt to isometrically embed the "swiss roll" into $\mathbb{R}^2$. However, it manages to unroll the "swiss roll" thereby preserving the locality on the manifold, although not the distances. We observe that for small values of N we obtain virtually identical representations for different t's. However, when N becomes bigger, smaller values of t seemingly lead to better representations.

It is worthwhile to point out that an isometric embedding preserving global distances such as that attempted by Isomap is theoretically possible only when the surface is flat, i.e., the curvature tensor is zero, which is the case with the "swiss roll". However a classical result due to Gauss shows that even for a 2-dimensional sphere (or any part of a sphere) no distance preserving map into the plane can exist.

2.6.2 A Toy Vision Example

Consider binary images of vertical and horizontal bars located at arbitrary points in the visual field. Each image contains exactly one horizontal or vertical bar at a random location in the image plane. In principal, we may consider each image to be represented as a function

$$f : [0, 1] \times [0, 1] \to \{0, 1\}$$

where $f(\mathbf{x}) = 0$ means the point $\mathbf{x} \in [0,1] \times [0,1]$ is white and $f(\mathbf{x}) = 1$ means the point is black. Let $v(x, y)$ be the image of a vertical bar. Then all images of vertical bars may be obtained from $v(x, y)$ by the following transformation:

$$v_t(x, y) = v(x - t_1, y - t_2)$$

The space of all images of vertical bars is a two dimensional manifold. Similarly, the space of all horizontal bars is a two dimensional manifold as well. Each of these manifolds is embedded in the space of functions ($L^2([0,1] \times [0,1])$). Notice that while these manifolds do not intersect, they come quite close to each other. In practice it is usually impossible to tell whether the intersection of two classes is empty or not.

To discretize the problem, we consider a 40×40 grid for each image. Thus each image may be represented as a 1600 dimensional binary vector. We choose 1000 images (500 containing vertical bars and 500 containing horizontal bars) at random. The parameter N is chosen to be 14 and $t = \infty$.

In fig. A.4 the left panel shows a horizontal and vertical bar to give the reader a sense of the scale of the image. The middle panel is a two dimensional representation of the set of all images using the Laplacian eigenmaps. Notice that while the local graph is connected, the two dimensional representation shows two well defined components. The right panel shows the result of a principal components analysis using the first two principal directions to represent the data.

2.6.3 A Linguistic Example

An experiment was conducted with the 300 most frequent words in the Brown corpus – a collection of texts containing about a million words (not distinct) available in electronic format. Each word is represented as a vector in a 600 dimensional space using information about the frequency of its left and right neighbors (computed from the corpus). More precisely, let the 300 words be w_1 through w_{300}. Then the representation of w_i is a 600 dimensional vector $\mathbf{v}_i$ (say) where the first three hundred dimensions of $\mathbf{v}_i$ characterize left neighbor relations and the next three hundred characterize right neighbor relations. Thus $\mathbf{v}_i(j)$ – the jth component ($j \leq 300$) of v_i

is the number of times the sequence $w_j w_i$ occurs in the corpus (referred to as the bigram count). Similarly, $\mathbf{v}_i(j + 300)$ is the the count of the number of times the sequence $w_i w_j$ occurs in the corpus.

Thus there are 300 vectors in $\mathbb{R}^{600}$. Of course, we do not claim that there is a natural low dimensional manifold structure on these vectors. Nevertheless, it is useful for practical applications to construct low dimensional representations of this data. For example, the well known LSI (Latent Semantic Indexing) approach uses Principal Components Analysis to represent the documents in a vector space model for purposes of search and information retrieval. Applying the Laplacian eigenmap with $N = 14; t = \infty$ to the data yields a low dimensional representation shown in figs. A.5 and A.6 respectively. Note that words belonging to similar syntactic categories seem to cluster together highlighting further the connections between clustering and dimensionality reduction as discussed in this thesis.

2.6.4 Speech

We turn finally to an example from human speech. It has long been recognized that while the speech signal is high dimensional, the distinctive phonetic dimensions are few. An important open question in the field is to develop a low dimensional representation of the speech signal that is correlated with phonetic content.

In this example, we consider the low dimensional representations that arise by applying the Laplacian eigenmap algorithm to a sentence of speech sampled at 16kHz. A short time Fourier transform (with a 30 millisecond window) was computed from the speech signal at 5 millisecond intervals. This yielded a vector of Fourier coefficients for every 30ms chunk of the speech signal. There were 685 such vectors in all As a standard practice in speech recognition, the data was represented by the logarithm of these Fourier coefficients. Each vector contained 256 logs of Fourier coefficients. As before we choose $N = 14; t = \infty$. Furthermore, each vector was labeled according to the identity of the phonetic segment it belonged to. These labels are not utilized by the Laplacian Eigenmap algorithm which finds a low dimensional representation for the data. Shown in fig. A.7, are the speech data points plotted in the two di-

mensional Laplacian representation. The two "spokes" correspond predominantly to fricatives and closures respectively. The central portion corresponds mostly to periodic sounds like vowels, nasals, and semivowels. A natural clustering into the broad classes is obtained and fig. reffig:phonemesome shows three different regions of the representation space. Note the phonetic homogeneity of the data points that lie in each of these regions. Points mapped to the same region in the representation space share similar phonetic features though points with the same label may originate from different occurrences of the same phoneme.

2.7 Conclusions

In this chapter we introduced a coherent framework for dimensionality reduction for the case where data resides on a low dimensional manifold embedded in a higher dimensional space. A number of questions remain to be answered.

1. Our approach utilizes the properties of Laplace-Beltrami operator to construct invariant embedding maps for the manifold. While such maps have some demonstrable locality preserving properties they do not in general provide an isometric embedding. The celebrated Nash's embedding theorem (Nash, 1954) guarantees that an n-dimensional manifold admits an isometric C^1 embedding into a $2n+1$ dimensional Euclidean space[5]. However it remains unclear whether such an embedding is easily computable by a discrete algorithm. Furthermore, there are usually many possible isometric embeddings of a given manifold. For example, any knot in $\mathbb{R}^3$ is an isometric embedding of a circle. However when the embedded manifold is isometric to a domain in $\mathbb{R}^k$ the canonical embedding is given by the exponential map. In that case Isomap provides an embedding and guarantees convergence (Bernstein et al, 2000). In general it is not clear how to discriminate between "good" and "bad" isometric embeddings. It would there-

5. The C^1 condition is essential. If the embedding has to be infinitely differentiable, the required dimension is much higher (Nash, 1956).

fore be interesting to formulate more precisely what properties of an embedding make it desirable for pattern recognition and data representation problems.

2. We have not given any consideration to other geometric invariants of the manifold that may be potentially estimated from data. For example, it is unclear how to reliably estimate even such a simple invariant as the intrinsic dimensionality of the manifold.

3. There are further issues pertaining to our framework that need to be sorted out. First, we have implicitly assumed a uniform probability distribution on the manifold according to which the data points have been sampled. Second, it remains unclear how the algorithm behaves when the manifold in question has a boundary. Third, appropriate choices for N (or ϵ) and t and their effect on the behavior of the embeddings need to better understood. Fourth, the convergence of the finite sample estimates of the embedding maps need to be addressed.

4. Finally, and most intriguingly, while the notion of manifold structure in natural data is a very appealing one, we do not really know how often and in which particular empirical contexts, the manifold properties are crucial to account for the phenomena at hand. Vastly more systematic studies of the specific problems in different application domains need to be conducted to shed light on this question.

CHAPTER 3
SEMI-SUPERVISED LEARNING

3.1 Introduction

In many practical applications of data classification and data mining, one finds a wealth of easily available unlabeled examples, while collecting labeled examples can be costly and time-consuming. Classical examples include object recognition in images, speech recognition, classifying news articles by topic and so on. In recent times, genetics has also provided enormous amounts of readily accessible data. However, classification of this data involves experimentation and can be very resource intensive.

Consequently it is of interest to develop algorithms that are able to utilize both labeled and unlabeled data for classification and other purposes. Although the area of partially labeled classification is fairly new a considerable amount of work has been done in that field since the early 90's (Blum, Chawla, 2001, Castelli, Cover, 1995, Nigam et al, 2000, Szummer, Jaakkola, 2002).

This chapter addresses the problem of classifying a partially labeled set by developing the ideas proposed in the previous chapter for data representation. In particular, we exploit the intrinsic structure of the data to improve classification with unlabeled examples under the assumption that the data resides on a low-dimensional manifold within a high-dimensional representation space. In some cases it seems to be a reasonable assumption that the data lies on or close to a manifold. For example a handwritten digit **0** can be fairly accurately represented as an ellipse, which is completely determined by the coordinates of its foci and the sum of the distances from the foci to any point. Thus the space of ellipses is a five-dimensional manifold. An actual handwritten **0** would require more parameters, but perhaps not more than 15 or 20. On the other hand the dimensionality of the ambient representation space is the number of pixels which is typically far higher.

For other types of data the question of the manifold structure seems significantly more involved. For example, in text categorization documents are typically represented by vectors whose elements are (sometimes weighted) counts of words/terms appearing in the document. It is far from clear why the space of documents should be a manifold. However there is no doubt that it has a complicated intrinsic structure and occupies only a tiny portion of the representation space, which is typically very high-dimensional, with dimensionality higher than 1000. We show that even lacking convincing evidence for manifold structure, we can still use our methods with good results. It is also important to note that while objects are typically represented by vectors in $\mathbb{R}^n$, the natural distance is often different from the distance induced by the ambient space $\mathbb{R}^n$.

While there has been recent work on using manifold structure for data representation (Roweis, Saul, 2000, Tenenbaum, et al, 2000), the only other application to machine learning, that we are aware of, was in Szummer, Jaakkola, 2002, where the authors use a random walk on the adjacency graph for partially labeled classification.

3.2 Why Manifold Structure is Useful for Partially Supervised Learning

Consider first a two-class classification problem with classes C_1, C_2 and the space $\mathcal{X}$, whose elements are to be classified. A probabilistic model for that problem would include two main ingredients, a probability density $p(x)$ on $\mathcal{X}$, and the class densities $\{p(C_1 \,|\, x \in \mathcal{X})\}$, $\{p(C_2 \,|\, x \in \mathcal{X})\}$. The unlabeled data alone does not necessarily tell us much about the conditional distributions as we cannot identify the classes without labels. However, we can improve our estimate of the probability density $p(x)$ using the unlabeled data.

The simplest example is two disjoint classes on the real line. In that case the Bayes risk is zero, and given sufficiently many unlabeled points, the structure can be recovered completely with just one labeled example. In general, the unlabeled data provides us with information about the probability distribution $p(x)$, while labeled points tell us about the conditional distributions.

We consider a version of this problem where $p(x)$ puts all its measure on a compact (low-dimensional) manifold in $\mathbb{R}^n$. Therefore, as we shall see shortly, the unlabeled examples can be used to estimate the manifold and the labeled examples then specify a classifier defined on that manifold.

To provide a motivation for using a manifold structure, consider a simple synthetic example shown in Figure A.1. The two classes consist of two parts of the curve shown in the first panel (row 1). We are given a few labeled points and a 500 unlabeled points shown in panels 2 and 3 respectively. The goal is to establish the identity of the point labeled with a question mark. There are several observations that may be made in the context of this example.

1. By observing the picture in panel 2 (row 1) we see that we cannot confidently classify **?** by using the labeled examples alone. On the other hand, the problem seems much more feasible given the unlabeled data shown in panel 3.

2. Since there is an underlying manifold, it seems clear at the outset that the (geodesic) distances along the curve are more meaningful than Euclidean distances in the plane. Many points which happen to be close in the plane are on the opposite sides of the curve. Therefore rather than building classifiers defined on the plane ($\mathbb{R}^2$) it seems preferable to have classifiers defined on the curve itself.

3. Even though the data suggests an underlying manifold, the problem is still not quite trivial since the two different parts of the curve come confusingly close to each other. There are many possible potential representations of the manifold and the one provided by the curve itself is unsatisfactory. Ideally, we would like to have a representation of the data which captures the fact that it is a closed curve. More specifically, we would like an embedding of the curve where the coordinates vary as slowly as possible when one traverses the curve. Such an ideal representation is shown in the panel 4 (first panel of the second row). Note that both represent the same underlying manifold structure but with different coordinate functions. It turns out (panel 6) that by taking a two-dimensional

representation of the data with Laplacian Eigenmaps, we get very close to the desired embedding. Panel 5 shows the locations of labeled points in the new representation space. We see that "?" now falls squarely in the middle of "+" signs and can easily be identified as a "+".

This artificial example illustrates that recovering the manifold and developing classifiers on the manifold itself might give us an advantage in classification problems. To recover the manifold, all we need is unlabeled data. The labeled data is then used to develop a classifier defined on this manifold. These are the intuitions we formalize in the rest of the chapter.

3.3 Representing Data as a Manifold

We hope we provided at least some justification for using the manifold structure for classification problems. Of course, this structure cannot be utilized unless we have a reasonable model for the manifold. The model used here is that of a weighted graph whose vertices are data points. Two data points are connected with an edge if and only if the points are adjacent, which typically means that either the distance between them is less than some ϵ or that one of them is in the set of n nearest neighbors of the other.

To each edge we can associate a distance between the corresponding points. The "geodesic distance" between two vertices is the length of the shortest path between them on the adjacency graph. Notice that the geodesic distance can be very different from the distance in the ambient space. It can be shown that if the points are sampled from a probability distribution supported on the whole manifold the geodesic distance on the graph will converge to the actual geodesic distance on the manifold as the number of points tends to infinity (see Tenenbaum, et al, 2000).

Once we set up an approximation to the manifold, we need a method to exploit the structure of the model to build a classifier. One possible simple approach would be to use the "geodesic nearest neighbors". The geodesic nearest neighbor of an unlabeled point u is a labeled point l such that "geodesic distance" along the edges of

the adjacency graph, between the points u and l is the shortest. Then as with usual nearest neighbors the label of l is assigned to u.

However, while simple and well-motivated, this method is potentially unstable. Even a relatively small amount of noise or a few outliers can change the results dramatically. A related more sophisticated method based on a random walk on the adjacency graph is proposed in Szummer, Jaakkola, 2002. We also note the approach taken in Blum, Chawla, 2001, which uses mincuts of certain graphs for partially labeled classifications.

3.4 Our Approach

Our approach is based on the Laplace-Beltrami operator on the manifold. A Riemannian manifold, i.e. a manifold with a notion of local distance, has a natural operator Δ on differentiable functions, which is known as the Laplace-Beltrami operator, or the Laplacian[1].

In the case of $\mathbb{R}^n$ the Laplace-Beltrami operator is simply $\Delta = -\sum_i \frac{\partial^2}{\partial x_i^2}$. Note that we adopt the geometric convention of writing it with the '-' sign.

Δ is a positive-semidefinite self-adjoint (with respect to the $\mathcal{L}^2$ inner product) operator on twice differentiable functions. Remarkably, it turns out when $\mathcal{M}$ is a compact manifold, Δ has a discrete spectrum and eigenfunctions of Δ provide an orthogonal basis for the Hilbert space $\mathcal{L}^2(\mathcal{M})$. Note that Δ is only defined on a subspace in $\mathcal{L}^2(\mathcal{M})$.

Therefore any function $f \in \mathcal{L}^2(\mathcal{M})$ can be written as

$$f(\mathbf{x}) = \sum_{i=0}^{\infty} a_i e_i(\mathbf{x})$$

where e_i are eigenfunctions, $\Delta e_i = \lambda_i e_i$.

1. There is an extensive literature on the connection between the geometric properties of the manifold and the Laplace-Beltrami operator. See Rosenberg, 1997, for an introduction to the subject.

Now assuming that the data lies on a manifold $\mathcal{M}$, we consider the simplest model, where the class membership is represented by a square integrable function $m : \mathcal{M} \to \{-1, 1\}$. Equivalently, we can say that the classes are represented by measurable sets S_1, S_2 with null intersection. Alternatively, if S_1 and S_2 do intersect, we can put $m(\mathbf{x}) = 1 - 2\ \mathrm{Prob}(\mathbf{x} \in S_1)$. The only condition we need is that $m(\mathbf{x})$ is a measurable function.

The classification problem can be interpreted as a problem of interpolating a function on a manifold. Since a function can be written in terms of the eigenfunctions of the Laplacian, we adjust the coefficients of the Laplacian to provide the optimal fit to the data (i.e the labeled points), just as we might approximate a signal with a Fourier series[2] $m(\mathbf{x}) \approx \sum_{0}^{N} a_i e_i(\mathbf{x})$.

It turns out that not only the eigenfunctions of the Laplacian are a natural basis to consider, but that they also satisfy a certain optimality condition. In a sense, which we will make precise later, they provide a maximally smooth approximation, similar to the way splines are constructed.

3.5 Description of the Algorithm

Given k points $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^l$, we assume that the first $s < k$ points have labels c_i, where $c_i \in \{-1, 1\}$ and the rest are unlabeled. The goal is to label the unlabeled points. We also introduce a straightforward extension of the algorithm when there are more than two classes.

Step 1 [*Constructing the Adjacency Graph with n nearest neighbors*]. Nodes i and j corresponding to the points $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected by an edge if i is among n nearest neighbors of j or j is among n nearest neighbors of i. The distance can be the standard Euclidean distance in $\mathbb{R}^l$ or some other distance, such as angle, which is natural for text classification problems. We take the weights to be one. For the discussion about the choice of weights, and the connection to

2. In fact when $\mathcal{M}$ is a circle, we do get the Fourier series.

the heat kernel see the second chapter. Thus $w_{ij} = 1$ if points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close and $w_{ij} = 0$ otherwise.

Step 2. [Eigenfunctions] Compute p eigenvectors corresponding to the smallest eigenvalues for the eigenvector problem:

$$L\mathbf{e} = \lambda\mathbf{e}$$

Matrix $L = W - D$ is the graph Laplacian for the adjacency graph. Here W is the adjacency matrix defined above and D is diagonal matrix of the same size as W, with row sums of W as entries, $D_{ii} = \sum_j W_{ji}$. Laplacian is a symmetric, positive semidefinite matrix which can be thought of as an operator on functions defined on vertices of the graph. The eigenfunctions can be interpreted as a generalization of the low frequency Fourier harmonics on the manifold defined by the data points.

$$\mathbf{E} = \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1k} \\ e_{21} & e_{22} & \dots & e_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{p1} & e_{p2} & \dots & e_{pk} \end{pmatrix}$$

Step 3. [*Building the classifier*] To approximate the class we minimize the error function

$$\mathrm{Err}(\mathbf{a}) = \sum_{i=1}^{s} \left(c_i - \sum_{j=1}^{p} a_j e_{ji} \right)^2$$

where p is the number of eigenfunctions we wish to employ and the sum is taken over all labeled points and the minimization is considered over the space of coefficients $\mathbf{a} = (a_1, \dots, a_p)^T$. The solution is given by

$$\mathbf{a} = \left(\mathbf{E}_{\mathrm{lab}}^T \mathbf{E}_{\mathrm{lab}} \right)^{-1} \mathbf{E}_{\mathrm{lab}}^T \mathbf{c}$$

where $\mathbf{c} = (c_1, \ldots, c_s)$ and

$$\mathbf{E}_{\text{lab}} = \begin{pmatrix} e_{11} & e_{12} & \ldots & e_{1s} \\ e_{21} & e_{22} & \ldots & e_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ e_{p1} & e_{p2} & \ldots & e_{ps} \end{pmatrix}$$

is the matrix of values of eigenfunctions on the labeled points. For the case of several classes, we build a one-against-all classifier for each individual class.

Step 4. [*Classifying unlabeled points*] If $\mathbf{x}_i$, $i > s$ is an unlabeled point we put

$$c_i = \begin{cases} 1, & \text{if } \sum_{j=1}^{p} e_{ij} a_j \geq 0 \\ -1, & \text{if } \sum_{j=1}^{p} e_{ij} a_j < 0 \end{cases}$$

This, of course, is just applying a linear classifier constructed in Step 3. If there are several classes, one-against-all classifiers compete using $\sum_{j=1}^{p} e_{ij} a_j$ as a confidence measure.

3.6 Theoretical Interpretation

Here we give a brief discussion of the theoretical underpinnings of the algorithm. Let $\mathcal{M} \subset \mathbb{R}^k$ be an n-dimensional compact Riemannian manifold isometrically embedded in $\mathbb{R}^k$ for some k[3]. Intuitively $\mathcal{M}$ can be thought of as a n-dimensional "surface" in $\mathbb{R}^k$. Riemannian structure on $\mathcal{M}$ induces a volume form that allows us to integrate functions defined on $\mathcal{M}$. The square integrable functions form a Hilbert space $\mathcal{L}^2(\mathcal{M})$. If by $C^\infty(\mathcal{M})$ we denote the space of infinitely differentiable functions on $\mathcal{M}$ then we have the Laplace-Beltrami operator as a second-order differential operator $\Delta_{\mathcal{M}} : C^\infty(\mathcal{M}) \to C^\infty(\mathcal{M})$.[4].

3. The assumption that the manifold is isometrically embedded in $\mathbb{R}^k$ is not necessary, but will simplify the discussion.

4. Strictly speaking, the functions do not have to be infinitely differentiable, but we prefer not to worry about the exact differentiability conditions.

There are two important properties of the Laplace-Beltrami operator that are relevant to our discussion here.

3.6.1 The Laplacian provides a basis on $\mathcal{L}^2(\mathcal{M})$

It can be shown (e.g., see Rosenberg, 1997) that Δ is a self-adjoint positive semidefinite operator and that its eigenfunctions form a basis for the Hilbert space $\mathcal{L}^2(\mathcal{M})$. The spectrum of Δ is discrete (provided $\mathcal{M}$ is compact) , with the smallest eigenvalue 0 corresponding to the constant eigenfunction. Therefore any $f \in \mathcal{L}^2(\mathcal{M})$ can be written as

$$f(\mathbf{x}) = \sum_{i=0}^{\infty} a_i e_i(\mathbf{x})$$

where e_i are eigenfunctions, $\Delta e_i = \lambda_i e_i$.

The simplest nontrivial example is a circle S^1.

$$\Delta_{S^1} f(\phi) = -\frac{d^2 f(\phi)}{d\phi^2}$$

Therefore the eigenfunctions are given by

$$-\frac{d^2 e(\phi)}{d\phi^2} = e(\phi)$$

where $f(\phi)$ is a π-periodic function. It is easy to see that all eigenfunctions of Δ are of the form $e(\phi) = \sin(n\phi)$ or $e(\phi) = \cos(n\phi)$ with eigenvalues $\{1^2, 2^2, \ldots\}$. Therefore, as a corollary of these far more general results, we see that the Fourier series for a π-periodic $\mathcal{L}^2$ function f converges to f in $\mathcal{L}^2$[5].

$$f(\phi) = \sum_{n=0}^{\infty} a_n \sin(n\phi) + b_n \cos(n\phi)$$

Thus we see that the eigenfunctions of the Laplace-Beltrami operator provide a natural basis for representing functions on $\mathcal{M}$. However Δ provides more than just a

5. Stronger conditions are needed for pointwise convergence.

basis, it also yields a measure of smoothness for functions on the manifold.

3.6.2 The Laplacian as a smoothness functional

A simple measure of the degree of smoothness (following the theory of splines, for example, Wahba, 1990) for a function f on a unit circle S^1 is the "smoothness functional"

$$\mathcal{S}(f) = \int_{S^1} |f(\phi)'|^2 d\phi$$

. If $\mathcal{S}(f)$ is close to zero, we think of f as being "smooth".

Naturally, constant functions are the most "smooth". Integration by parts yields

$$\mathcal{S}(f) = \int_{S^1} f'(\phi) df = \int_{S^1} f \Delta f d\phi = \langle \Delta f, f \rangle_{\mathcal{L}^2(S^1)}$$

In general, if $f : \mathcal{M} \to \mathbb{R}$, then

$$\mathcal{S}(f) \stackrel{\text{def}}{=} \int_{\mathcal{M}} |\nabla f|^2 d\mu = \int_{\mathcal{M}} f \Delta f d\mu = \langle \Delta f, f \rangle_{\mathcal{L}^2(\mathcal{M})}$$

where ∇f is the gradient vector field of f. If the manifold is $\mathbb{R}^n$ then $\nabla f = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \frac{\partial}{\partial x_i}$. In general, for an n-manifold, the expression in a local coordinate chart involves the coefficients of the metric tensor.

Therefore the smoothness of a unit norm eigenfunction e_i of Δ is controlled by the corresponding eigenvalue λ_i since

$$S(e_i) = \langle \Delta e_i, e_i \rangle_{\mathcal{L}^2(\mathcal{M})} = \lambda_i$$

For an arbitrary $f = \sum_i \alpha_i e_i$, we can write $\mathcal{S}(f)$ as

$$\mathcal{S}(f) = \langle \Delta f, f \rangle = \langle \sum_i \alpha_i \Delta e_i, \sum_i \alpha_i e_i \rangle = \sum_i \lambda_i \alpha_i^2$$

The linear subspace, where the smoothness functional is finite is a Reproducing Kernel Hilbert Space (e.g., see Wahba, 1990).

It is not hard to see that $\lambda_1 = 0$ is the smallest eigenvalue for which the eigenfunction is the constant function $e_1 = \frac{1}{\mu(\mathcal{M})}$. It can also be shown that if $\mathcal{M}$ is compact and connected there are no other eigenfunctions with eigenvalue 0.

Therefore approximating a function $f(x) \approx \sum_1^p a_i e_i(x)$ in terms of the first p eigenfunctions of Δ is a way of controlling the smoothness of the approximation. The optimal approximation is obtained by minimizing the $\mathcal{L}^2$ norm of the error:

$$\mathbf{a} = \underset{\mathbf{a}=(a_1,\ldots,a_p)}{\operatorname{argmin}} \int_{\mathcal{M}} \left(f(\mathbf{x}) - \sum_i^p a_i e_i(\mathbf{x}) \right)^2 d\mu$$

This approximation is given by a projection in $\mathcal{L}^2$ onto the span of the first p eigenfunctions

$$a_i = \int_{\mathcal{M}} e_i(\mathbf{x}) f(\mathbf{x}) d\mu = \langle e_i, f \rangle_{\mathcal{L}^2(\mathcal{M})}$$

In practice we only know the values of f at a finite number of points $\mathbf{x}_1, \cdots, \mathbf{x}_n$ and therefore have to solve a discrete version of this problem

$$\bar{\mathbf{a}} = \underset{\bar{\mathbf{a}}=(\bar{a}_1,\ldots,\bar{a}_p)}{\operatorname{argmin}} \sum_{i=1}^n \left(f(\mathbf{x}_i) - \sum_{j=1}^p \bar{a}_j e_j(\mathbf{x}_i) \right)^2$$

The solution to this standard least squares problem is given by

$$\bar{\mathbf{a}}^T = (\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}\mathbf{y}^T$$

where $\mathbf{E}_{ij} = e_i(\mathbf{x}_j)$ and $\mathbf{y} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$.

3.7 Regularization on Graphs

We consider the problem of predicting the labels on vertices of a partially labeled graph. Consider a weighted graph $G = (V, E)$ where $V = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is the vertex

set and E is the edge set. Associated with each edge $e_{ij} \in E$ is a weight W_{ij}. If there is no edge present between $\mathbf{x}_i$ and $\mathbf{x}_j$, $W_{ij} = 0$. Imagine a situation where a subset of these vertices are labeled with values $y_i \in \mathbb{R}$. We wish to predict the values of the rest of the vertices. In doing so, we would like to exploit the structure of the graph. In particular, in our approach we will assume that the weights are indications of the affinity of nodes with respect to each other and consequently are related to the potential similarity of the y values, these nodes are likely to have.

To approximate a function on a graph G, with the weight matrix W_{ij} we need a notion of a "good" function. One way to think about such a function is that is that it does not make too many "jumps". We formalize that notion by taking

$$\sum_{i \sim j} W_{ij}(f_i - f_j)^2$$

It can be seen that $\sum\limits_{i \sim j} W_{ij}(f_i - f_j)^2 = \mathbf{f}^T L \mathbf{f}$, where L is the Laplacian $L = D - W$, $D = \text{diag}(\sum_i W_{1i}, \ldots, \sum_i W_{ni})$.

Other smoothness matrices, such as L^n, $\exp(-tL)$ are also possible.

3.7.1 Algorithms for Regression on Graphs

Let $G = (V, E)$ be a graph be a graph with n vertices and the weight matrix W_{ij}. We assume that G is connected. We assume that the vertices of the graph are numbered. We would like to regress a function $f : V \to \mathbb{R}$. f is defined on vertices of G, however we have only partial information, say for the first k vertices. That is $f(\mathbf{x}_i) = y_i$, $1 \leq i \leq k$. Potentially the labels can be noisy.

We precondition the data by mean subtracting first. That is we take

$$\tilde{\mathbf{y}} = (y_1 - \bar{y}, \ldots, y_k - \bar{y})$$

where $\bar{y} = \frac{1}{k} \sum y_i$. That leads to improved stability of the algorithms as will be seen in the theoretical discussion. From now on we assume that the data has mean 0.

Algorithm 1: Tikhonov regularization (parameter $\gamma \in \mathbb{R}$). The objective is to minimize the square loss function plus the smoothness penalty.

$$f = \underset{\mathbf{f}=(f_1,\ldots,f_n)}{\operatorname{argmin}} \frac{1}{k}\sum_i (f_i - y_i)^2 + \gamma \mathbf{f}^t S \mathbf{f}^t$$

S here is a smoothness matrix, e.g. $S = L$ or $S = L^t$.

The solution is given in the form

$$\mathbf{f} = (k\gamma S + I_k)^{-1}\mathbf{y}$$

where $\tilde{\mathbf{y}}$ is the n-vector $\mathbf{y} = (y_1, y_2, \ldots, y_k, 0, \ldots, 0)$ and I_k is a matrix with k ones on the diagonal and zero off the diagonal, $I_k = \mathrm{d}iag(1, 1, \ldots, 1, 0, \ldots, 0)$.

Algorithm 2: Interpolated Regularization (no parameters).

Here we assume that the values $y_1, \ldots, y_k$ have no noise. Thus the optimization problem is to find a function of maximum smoothness satisfying $f(\mathbf{x}_i) = y_i$, $1 \le i \le k$:

$$\mathbf{f} = \underset{\mathbf{f}=(y_1,\ldots,y_k,f_{k+1},\ldots,f_n)}{\operatorname{argmin}} \mathbf{f}^t S \mathbf{f}$$

As before S is a smoothness matrix, typically L or L^2. We partition S as

$$S = \begin{pmatrix} S_1 & S_2 \\ S_2^T & S_3 \end{pmatrix}$$

where S_1 is a $k \times k$ matrix, S_2 is $k \times n-k$ and S_3 is $(n-k) \times (n-k)$. Let $\tilde{f}$ be the values of f, where the function is unknown, $\tilde{\mathbf{f}} = (f_{k+1}, \ldots, f_n)$.

Straightforward linear algebra yields the solution:

$$\tilde{\mathbf{f}} = S_3^{-1} S_2^T (y_1, \ldots, y_k)^T$$

The regression formula is very simple and has no free parameters. However the quality of results depends on whether S_3 is well conditioned.

3.7.2 Connection with the Graph Laplacian

As we are approximating a manifold with a graph, we need a suitable measure of smoothness for functions defined on the graph.

It turns out that many of the concepts in the previous section have parallels in graph theory (e.g., see Chung, et al). Let $G = (V, E)$ be a weighted graph on n vertices. We assume that the vertices are numbered and use the notation $i \sim j$ for adjacent vertices i and j. The graph Laplacian of G is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{W}$ is the weight matrix and $\mathbf{D}$ is a diagonal matrix, $D_{ii} = \sum_j W_{ji}$. [6] $\mathbf{L}$ can be thought of as an operator on functions defined on vertices of the graph. It is not hard to see that $\mathbf{L}$ is a self-adjoint positive semidefinite operator. By the (finite dimensional) spectral theorem any function on G can be decomposed as a sum of eigenfunctions of $\mathbf{L}$.

If we think of G as a model for the manifold $\mathcal{M}$ it is reasonable to assume that a function on G is smooth if it does not change too much between nearby points. If $\mathbf{f} = (f_1, \ldots, f_n)$ is a function on G, then we can formalize that intuition by defining the smoothness functional

$$\mathcal{S}_G(\mathbf{f}) = \sum_{i \sim j} w_{ij}(f_i - f_j)^2$$

It is not hard to show that

$$\mathcal{S}_G(\mathbf{f}) = \mathbf{f}\,\mathbf{L}\,\mathbf{f}^T = \langle \mathbf{f}, \mathbf{L}\mathbf{f} \rangle_G = \sum_{i=1}^{n} \lambda_i \langle \mathbf{f}, \mathbf{e}_i \rangle_G$$

which is the discrete analogue of the integration by parts from the previous section. The inner product here is the usual Euclidean inner product on the vector space with coordinates indexed by the vertices of G, $\mathbf{e}_i$ are normalized eigenvectors of $\mathbf{L}$, $\mathbf{L}\mathbf{e}_i = \lambda_i \mathbf{e}_i$, $\|\mathbf{e}_i\| = 1$. All eigenvalues are non-negative and the eigenfunctions

6. The alternative definition is the so-called "normalized Laplacian" $\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{\frac{1}{2}}$ which has many nice properties. That is the definition in Chung, 1997.

corresponding to the smaller eigenvalues can be thought as "more smooth". The smallest eigenvalue $\lambda_1 = 0$ corresponds to the constant eigenvector $\mathbf{e}_1$.

The idea behind popular spectral partitioning and clustering techniques is essentially the following observation. If the graph $G = (V, E)$ has a partition in two parts $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ with only a few edges connecting V_1 and V_2, then it is possible to construct a "smooth" non-constant function $\mathbf{p}$ on G by setting

$$p_i = \begin{cases} -\frac{1}{\mathrm{vol}(V_1)}, & \text{if vertex } i \in V_1 \\ \frac{1}{\mathrm{vol}(V_2)}, & \text{if vertex } i \in V_2 \end{cases}$$

Here $\mathrm{vol}(V_l) = \sum_{\substack{i,j \in V_l \\ i \sim j}} w_{ij}$, $l \in \{1, 2\}$. It is easy to check that $\mathbf{p}$ is orthogonal to $\mathbf{e}_1$ and that $\|\mathbf{p}\| = 1$. It can be seen that if the "cut" between V_1 and V_2 (i.e. the total weight of connecting edges) is small, then $\mathcal{S}(\mathbf{p})$ is small as well. Thus good partitions correspond to smooth functions on a graph, orthogonal to the constant function. Therefore one can think of $\mathbf{e_2}$ as an approximation for $\mathbf{p}$. While finding an optimal $\mathbf{p}$ is a hard combinatorial problem, calculating $\mathbf{e}_2$ is straightforward. The graph can then be partitioned into the part where $\mathbf{e}_2$ is greater than 0 and its complement. λ_2 is known as the algebraic connectivity of the graph G, or its Fiedler number.

The corresponding observation for the continuous case was in fact made even earlier by Cheeger (Cheeger, 1970) who went in the opposite direction. Cheeger used a geometric invariant of the manifold, to find a lower bound for the smallest nonzero eigenvalue of Δ.

3.8 Experimental Results

The experiments with unlabeled and unlabeled data may be conducted in two different ways.

1. Labeling a partially labeled data set: Given a set L of labeled examples and a set U of unlabeled data, classify the unlabeled set with maximal accuracy. This setting is often referred to as "transductive inference".

2. Labeling a held out test set using a training set consisting of labeled and unlabeled examples.

Note that ultimately (1) and (2) are equivalent in the following sense. First, (2) implies (1) trivially as we can always use the developed classifier to classify the unlabeled examples. But also (1) implies (2). If we have an algorithm for solving (1), then we can solve (2), i.e., classify a new point x by simply adding x to the unlabeled set and running the algorithm with this revised unlabeled set $U \cup \{x\}$.

In the following sections, we concentrate on experiments conducted in the first setting. We can, of course, use the method (1) for solving problems in the second setting as well. However, following such protocol literally turns out to be computationally too expensive as a large eigenvalue problem has to be solved for each new test point. Instead we propose a simple heuristic and provide some encouraging experimental results for this case.

3.8.1 Handwritten Digit Recognition

As an application of our techniques we consider the problem of optical character recognition. We use the popular MNIST dataset which contains 28x28 grayscale images of handwritten digits.[7] We use the 60000 image training set for our experiments. For all experiments we use 8 nearest neighbors to compute the adjacency matrix. Note that the adjacency matrices are very sparse which makes solving eigenvector problems for matrices as big as 60000 by 60000 possible.

All 60000 images are provided with labels in the original dataset. For a particular trial, we fix the number of labeled examples we wish to use. A random subset of the 60000 images is used with labels to form L. The rest of the images are used without labels to form U. The classification results (for U) are averaged over 20 different random draws for L. The results are presented in Table 3.1. Each row

7. We use the first 100 principal components of the set of all images to represent each image as a 100 dimensional vector. This was done to accelerate finding the nearest neighbors, but turned out to have a pleasant side effect of improving the baseline classification accuracy, possibly by denoising the data.

corresponds to a different number of labeled points (size of L). We compute the error rates when the number of eigenvectors is smaller than the number of labeled points as no generalization can be expected to take place otherwise.

The rightmost columns show baseline performances obtained using the best k-nearest neighbors classifier (k was taken to be 1, 3 or 5) to classify the unlabeled set U using the labeled set L. We choose the nearest neighbors as a baseline, since the Laplacian based algorithm presented in this chapter makes use only of the nearest neighbor information to classify the unlabeled data. In addition, nearest neighbors is known to be a good general purpose classifier. k-NN and its variations are often used in practical applications.

Each row represents a different choice for the number of labeled examples used. The columns show performance for different choices of the number of eigenvectors of the graph Laplacian retained by the algorithm.

For a fixed number of eigenvectors, performance improves with the number of labeled points but saturates after a while. The saturation point is empirically seen to be when the number of labeled points is roughly ten times the number of eigenvectors.

For a fixed number of labeled points, error rate decreases with the number of eigenvectors and then begins to increase again. Presumably, if too many eigenvectors are retained, the algorithm starts to overfit. This turning point happens when the number of eigenvectors is somewhere between 10% and 50% of the number of labeled examples. The 20% percent ratio seems to work well in a variety of experiments with different data sets and this is what we recommend for comparison with the base line.

The improvements over the base line are striking, sometimes exceeding 70% depending on the number of labeled and unlabeled examples. With only 100 labeled examples (and 59900 unlabeled examples), the Laplacian classifier does nearly as well as the nearest neighbor classifier with 5000 labeled examples. Similarly, with 500/59500 labeled/unlabeled examples, it does slightly better than the nearest neighbor base line using 20000 labeled examples.

Shown in fig. A.9 is a summary plot of classification accuracy on the unlabeled set comparing the nearest neighbors baseline with our algorithm that retains the number

Labeled	Number of Eigenvectors								best
points	5	10	20	50	100	200	500	1000	k-NN
20	53.7	35.8							53.4
50	48.3	24.7	12.9						37.6
100	48.6	22.0	6.4	14.4					28.1
500	49.1	22.7	5.6	3.6	3.5	7.0			15.1
1000	51.0	24.1	5.5	3.4	3.2	3.4	8.1		10.8
5000	47.5	25	5.6	3.4	3.1	2.9	2.7	2.7	6.0
20000	47.7	24.8	5.4	3.3	3.1	2.9	2.7	2.4	3.6
50000	47.3	24.7	5.5	3.4	3.1	3.0	2.7	2.4	2.3

Table 3.1: Percentage error rates for different numbers of labeled points for the 60000 point MNIST dataset. The error rate is calculated on the unlabeled part of the dataset, each number is an average over 20 random splits.The rightmost two columns contain the nearest neighbor base line.

of eigenvectors by following the 20% rule.[8] The results for the total 60000 point data set, and 10000 and 1000 subsets are compared. We see that adding unlabeled data consistently improves classification accuracy. We notice that when almost all of the data is labeled, the performance of our classifier is close to that of k-NN. It is not particularly surprising as our method uses the nearest neighbor information. Curiously, it is also the case when there are very few labeled points (20 labeled points, or just 2 per class on average). Both observations seem to be applicable across the datasets. Not surprisingly for the same number of labeled points, using fewer unlabeled points results in a higher error rate. However, yet again, when the number of labeled examples is very small (20 and 50 labeled examples, i.e., an average of 2 and 5 examples per class), the number of unlabeled points does not seem to make much difference. We conjecture this might be due to the small number of eigenvectors used, which is not sufficient to capture the behavior of the class membership functions.

8. For 60000 points we were unable to compute more than 1000 eigenvectors due to the memory limitations. Therefore the actual number of eigenvectors never exceeds 1000. We suspect that computing more eigenvectors would improve performance even further.

3.8.2 Text Classification

The second application we consider is text classification using the popular 20 Newsgroups data set. This data set contains approximately 1000 postings from each of 20 different newsgroups. Given an article, the problem is to determine to which newsgroup it was posted. The problem is fairly difficult as many of the newsgroups deal with similar subjects. For example, five newsgroups discuss computer-related subjects, two discuss religion and three deal with politics. About 4% of the articles are cross-posted. Unlike the handwritten digit recognition, where human classification error rate is very low, there is no reason to believe that this would be an easy test for humans. There is also no obvious reason why this data should have manifold structure.

We tokenize the articles using the Rainbow software package written by Andrew McCallum. We use a standard "stop-list" of 500 most common words to be excluded and also exclude headers, which among other things contain the correct identification of the newsgroup. No further preprocessing is done. Each document is then represented by the counts of the most frequent 6000 words normalized to sum to 1. Documents with 0 total count are removed, thus leaving us with 19935 vectors in a 6000-dimensional space.

The distance is taken to be the angle between the representation vectors. More sophisticated schemes, such as TF-IDF representations, increasing the weights of dimensions corresponding to more relevant words and treating cross-posted articles properly would be likely to improve the baseline accuracy.

We follow the same procedure as with the MNIST digit data above. A random subset of a fixed size is taken with labels to form L. The rest of the dataset is considered to be U. We average the results over 20 random splits[9]. As with the digits, we take the number of nearest neighbors for the algorithm to be 8.

The results are summarized in the table 3.2

9. In the case of 2000 eigenvectors we take just 10 random splits since the computations are rather time-consuming.

Labeled	Number of Eigenvectors									best
points	5	10	20	50	100	200	500	1000	2000	k-NN
50	83.4	77.3	72.1							75.6
100	81.7	74.3	66.6	60.2						69.6
500	83.1	75.8	65.5	46.4	40.1	42.4				54.9
1000	84.6	77.6	67.1	47.0	37.7	36.0	42.3			48.4
5000	85.2	79.7	72.9	49.3	36.7	32.3	28.5	28.1	30.4	34.4
10000	83.8	79.8	73.8	49.8	36.9	31.9	27.9	25.9	25.1	27.7
18000	82.9	79.8	73.8	50.1	36.9	31.9	27.5	25.5	23.1	23.1

Table 3.2: Percentage error rates for various numbers of labeled points and eigenvectors. The total number of points is 19935. The error is calculated on the unlabeled part of the dataset.

We observe that the general patterns of the data are very similar to those for the MNIST data set. For a fixed number of labeled points, performance is optimal when the number of eigenvectors retained is somewhere between 20% and 50% of the number of labeled points. We see that when the ratio of labeled to unlabeled examples is either very small or very large, the performance of our algorithm is close to that of the nearest neighbor baseline, with the most significant improvements occurring in the midrange, seemingly when the number of labeled points is between 5% and 30% of the unlabeled examples.

While the decreases in the error rate from the baseline are not quite as good as with MNIST data set, they are still significant, reaching up to 30%.

In fig. A.10 we summarize the results by taking 19935, 2000 and 600 total points respectively and calculating the error rate for different numbers of labeled points. The number of eigenvectors used is alway 20% of the number of labeled points. We see that having more unlabeled points improves the classification error in most cases although when there are very few labeled points, the differences are small.

3.8.3 Phoneme Classification

Here we consider the problem of phoneme classification. More specifically we are trying to distinguish between three vowels "aa" (as in "dark"), "iy" (as in "beat"),"eh" (as in "bet"). The data is taken from the TIMIT data set. The data is presegmented

Labeled	Number of Eigenvectors								best
points	5	10	20	50	100	200	500	1000	k-NN
20	28.5	23.7							28.7
50	24.9	15.0	19.9						21.2
100	22.7	13.0	13.3	18.8					18.2
500	22.7	12.3	11.6	10.3	10.7	13.4			12.7
1000	22.4	12.2	11.3	9.9	9.7	10.3	14.5		11.5
5000	21.8	12.2	11.3	9.6	9.2	9.1	8.9	9.3	9.7
10000	22.3	12.2	11.1	9.4	9.2	8.9	8.4	8.5	9.0

Table 3.3: Percentage error rates for various numbers of labeled points and eigenvectors. The total number of points is 13168. The error is calculated on the unlabeled part of the dataset.

into phonemes. Each vowel is represented by the average of the logarithm of the Fourier spectrum of each frame in the middle third of the phoneme.

We follow the same procedure as before, the number of nearest neighbors is taken to be 10. The total number of phonemes considered is 13168. The results are shown in table 3.3 and fig. A.11. The results parallel those for the rest of our experiments with one interesting exception: no significant difference is seen between the results for just 2000 total points and the whole dataset. In fact the corresponding graphs are almost identical. However going from 600 points to 2000 points yields in a significant performance improvement. It seems that for this particular data set unlabeled the structure is learned with relatively few unlabeled points.

3.8.4 Improving Existing Classifiers With Unlabeled Data

In this chapter we considered the problem of classifying the unlabeled data. While theoretically a classifier can be constructed by simply adding each new point to the unlabeled data and reclassifying, this method is far too slow to be of much practical use. A somewhat more practical suggestion would be to accumulate unlabeled data first and then classify it in the "batch" mode.

However another intriguing possibility is to classify the unlabeled data, and then to use these labels to train a different classifier with the hope that the error rate on the unlabeled data would be small.

Figure A.12 provides an illustration of this technique on the MNIST data set. Using a certain number of labeled points on the 60000 point training set, we use our algorithm to classify the remainder of the dataset. We then use the obtained fully labeled training set (with some incorrect labels, of course) to classify the held out 10000 point test set (which we do not use in other experiments) using a 3-NN classifier. We see that the performance is only slightly worse than the Laplacian baseline error rate, which is calculated on the unlabeled portion of the training set[10]. By thus labeling the unlabeled data set and treating it as a fully labeled training set, we obtained significant improvements over the baseline best k-NN ($k = 1, 3, 5$) classifier.

3.9 Conclusions and Further Directions

We have shown that methods motivated by the geometry of manifolds can yield significant benefits for partially labeled classification. We believe that this is just a first step towards more systematically exploiting the geometric structure of the data as many crucial questions still remain to be answered.

1. In this chapter we have not discussed questions relating to the convergence of our algorithm. It seems that under certain conditions convergence can be demonstrated rigorously, however the precise connection between the parameters of the manifold such as curvature and the nature of convergence are still unclear. We note that the heat equation seems to play a crucial role in this context.

2. It would be very interesting to explore different bases for functions on the manifold. There is no reason to believe that the Laplacian is the only or the most natural choice. Note that there are a number of different bases for function approximation and regression in $\mathbb{R}^k$.

10. If the test set is included with the training set and is labeled in the "batch mode", the error rate drops down to the base line.

3. While the idea that natural data lie on manifolds has recently attracted considerable attention, there still seems to be no convincing proof that such manifold structures are actually present. While the results in this chapter provide some indirect evidence for this, it would be extremely interesting to develop methods to look for such structures. Even the simplest questions such as practical methods for estimating the dimensionality seem to be unresolved.

CHAPTER 4
CONVERGENCES ISSUES

This chapter provides a proof for some convergence properties of the Laplacian. We will show the the Laplacian of the adjacency graph for data sampled from a uniform distribution on a compact submanifold $\mathcal{M}$ of $\mathbb{R}^N$ tend to the Laplace-Beltrami operator on $\mathcal{M}$. Of course, this statement will need to be clarified and qualified as will be shown below. Thus this chapter provides a theoretical foundation for the algorithms described in the previous two chapters. While in practice it is unlikely that the data actually resides on a manifold, it seems quite possible that it might be concentrated around a manifold, with perhaps some noise. We do not, however, tackle the problem of noise or more general probability distributions here.

Recall that the Laplacian in $\mathbb{R}^n$ is the second degree differential operator

$$\Delta f(\mathbf{x}) = -\sum_{1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

We follow the geometric convention here by placing the minus sign in front of the sum.

The Laplace-Beltrami operator plays a fundamental role in partial differential equations. It appears in two of the main equations of mathematical physics - the wave equation and the heat equation. We will use the connection to the heat equation throughout this chapter. It is interesting to note one reason for its ubiquity:

Theorem 4.0.1. *Any partial differential operator T on $\mathbb{R}^n$ invariant under translation and rotation can be written as*

$$T = \sum_{j} a_j \Delta^j$$

where a_j are constants.

4.1 Computations in $\mathbb{R}^n$

This section establishes some technical results in $\mathbb{R}^n$, which we will need for the general case later. Throughout this chapter we will assume that all function are sufficiently differentiable and that all integrals exist.

Recall that a function $u(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ satisfies the heat equation in $\mathbb{R}^n$ if

$$\frac{\partial}{\partial t} u(\mathbf{x}, t) + \Delta_{\mathbf{x}} u(x, t) = 0$$

Thus $u(\mathbf{x}, t)$ can be viewed as the amount of heat at point $\mathbf{x}$ and time t resulting from the heat diffusion starting with the initial distribution $f(\mathbf{x}) = u(\mathbf{x}, 0)$.

A basic result in partial differential equations shows how to solve this equation:

Theorem 4.1.1 (Existence of the heat kernel in $\mathbb{R}^n$). *Given the initial heat distribution $f(\mathbf{x})$, where f is a continuous and bounded function, there exists a solution to the heat equation given as*

$$u(\mathbf{x}, t) = f * H_t(\mathbf{x}, \mathbf{y}) = \int_{\mathbb{R}^n} f(\mathbf{y}) H_t(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

where

$$H_t(\mathbf{x}, \mathbf{y}) = (4\pi t)^{-\frac{n}{2}} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}}$$

Moreover, $\lim_{t \to 0} u(\mathbf{x}, t) = f(\mathbf{x})$.

It immediately follows that for a sufficiently nice f

Corollary 4.1.2.

$$\Delta f(\mathbf{x}) = -\frac{\partial}{\partial t} \left((4\pi t)^{-\frac{n}{2}} \int_{\mathbb{R}^n} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}} f(\mathbf{y}) d\mathbf{y} \right) \Bigg|_0$$

Since the Gaussian becomes very "thin" for small values of t, it is easy to see that we can restrict the domain of integration to any set containing $\mathbf{x}$ in its interior:

Lemma 4.1.3.

$$\Delta f(\mathbf{x}) = -\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\int_B e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}} f(\mathbf{y})d\mathbf{y}\right)\Bigg|_0$$

where B is any open set containing $\mathbf{x}$.

Proof. Put $g = \chi_B f$, where χ_B is the indicator function for the set B. It is clear that $\Delta f(\mathbf{x}) = \Delta g(\mathbf{x})$. By the previous corollary

$$\Delta g(\mathbf{x}) = -\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\int_{\mathbb{R}^n} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}} g(\mathbf{y})d\mathbf{y}\right)\Bigg|_0$$

$$= -\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\int_B e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}} f(\mathbf{y})d\mathbf{y}\right)\Bigg|_0$$

□

For the sake of simplicity we will from now on assume that the Laplacian is always computed at the origin $\mathbf{x} = 0$.

Lemma 4.1.4. *Let $\phi : \mathbb{R}^n \to \mathbb{R}^n$ be a differentiable function such that $\phi(\mathbf{x}) = \mathbf{x} + O(\mathbf{x}^3)$, i.e. the Taylor expansion for each coordinate of ϕ does not have any terms of degree 2, $[\phi(\mathbf{x})]_i = x_i + O(x_i^3)$ at the origin. Then for any open set B containing the origin the following two expressions hold (the first one is true even if ϕ has terms of degree 2.*

$$f(0) = \lim_{t\to 0}(4\pi t)^{-\frac{n}{2}}\int_{B\in R^n} e^{-\frac{\phi(\mathbf{y})^2}{4t}} f(\mathbf{y})\,d\mathbf{y} \tag{4.1}$$

$$\Delta f(0) = -\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\int_{B\in R^n} e^{-\frac{\phi(\mathbf{y})^2}{4t}} f(\mathbf{y})\,d\mathbf{y}\right)\Bigg|_0 + Cf(0) \tag{4.2}$$

C here is a constant depending only on ϕ.

Proof. We will concentrate on proving formula (4.2), formula (4.1) is a simple corollary of the computation below. As follows from the previous lemmas, the set B can be chosen to be the whole space R^n. For simplicity we will show the formula when

$n = 1$. The case of arbitrary n is no different but requires rather cumbersome notation. We can write $f(y) = a_0 + a_1 y + a_2 y^2 + \ldots$ and $\phi(y) = y + b_0 y^3 + \ldots$. Put $y = \sqrt{t}x$. Changing the variable, we get:

$$\frac{1}{\sqrt{t}} \int_{\mathbb{R}} e^{-\frac{\phi(y)^2}{4t}} f(y) dy = \frac{1}{\sqrt{t}} \int_{\mathbb{R}} e^{-\frac{ty^2 + t^2 b_0 y^4 + \ldots}{4t}} f(\sqrt{t}y)\sqrt{t} dy =$$

$$= \int_{\mathbb{R}} e^{-\frac{y^2 + t b_0 y^4 + o(t)}{4}} f(\sqrt{t}y) dy$$

Note that $e^{-\frac{y^2 + t b_0 y^4 + o(t)}{4}} = e^{-\frac{y^2}{4}} e^{-\frac{t b_0 y^4 + o(t)}{4}} = e^{-\frac{y^2}{4}}(1 - t\frac{b_0}{4} y^4 + o(t))$.

Thus the previous integral can be written as

$$\int_{\mathbb{R}} e^{-\frac{x^2}{4}} \left(1 - t\frac{b_0}{4} x^4 + o(t)\right) f(\sqrt{t}x) dx$$

$$= \int_{\mathbb{R}} e^{-\frac{x^2}{4}} \left(1 - t\frac{b_0}{4} x^4 + o(t)\right) \left(a_0 + a_1\sqrt{t}x + a_2 t x^2 + o(t)\right) dx$$

$$= \int_{\mathbb{R}} e^{-\frac{x^2}{4}} \left(a_0 + a_1\sqrt{t}x + t(a_2 x^2 - a_0\frac{b_0}{4} x^4) + o(t)\right) dx$$

Note that the second term $a_1\sqrt{t}x$ is an odd function in x and therefore $\int_{\mathbb{R}} e^{-\frac{x^2}{4}} a_1\sqrt{t}x dx = 0$.

Thus

$$\frac{\partial}{\partial t}\left(\frac{1}{2\sqrt{t\pi}} \int_{\mathbb{R}} e^{-\frac{y^2 + y^4\phi(y)}{4t}} f(y) dy\right)\Big|_0 = \frac{1}{2\sqrt{\pi}} \int_{\mathbb{R}} e^{-\frac{x^2}{4}} a_2 x^2 dx - \frac{1}{2\sqrt{\pi}} \int_{\mathbb{R}} e^{-\frac{x^2}{4}} a_0 \frac{b_0}{4} x^4 dx$$

The first integral in the sum is exactly the Laplacian of f at 0, $\Delta f(0) = \frac{1}{2\sqrt{\pi}} \int_{\mathbb{R}} e^{-\frac{x^2}{4}} a_2 x^2 dx$. The second summand depends only on the value $a_0 = f(0)$ and the function ϕ, which completes the proof. □

4.2 Computations on a Submanifold in $\mathbb{R}^N$

In this section we consider an n-dimensional manifold $\mathcal{M}^n$ isometrically embedded in $\mathbb{R}^N$. We will be interested in computing the Laplacian-Beltrami operator for $\mathcal{M}$ at a fixed point $\mathcal{M}$. Without a loss of generality we may assume that that point is the origin of the ambient space, $0 \in \mathcal{M}$.

Lemma 4.2.1. *The geodesic distance from* 0 *as a function of* $\mathbf{y}$ *can be written as*

$$\mathrm{dist}_{\mathcal{M},0}(\mathbf{y}) = \|\mathbf{y}\| + \psi(\mathbf{y})$$

where ψ *vanishes to order* 2 *at* 0 *and* $\|\mathbf{y}\|$ *is the Euclidean norm in* $\mathbb{R}^N$.

Proof. We first prove the statement for the curve length in $\mathbb{R}^2$. Let $f(x)$ be a differentiable function. Without the loss of generality we can assume that $f(0) = 0$, $f'(0) = 0$. Therefore $f(x) = ax^2 + O(x^3)$. Now the length of the curve along the graph of $f(x)$ is given by

$$\mathrm{dist}_{\mathcal{M},0}(t) = \int_0^t \sqrt{1 + (f')^2}\, dx$$

We have $\sqrt{1 + (f')^2} = 1 + 2ax^2 + O(x^3)$. Thus

$$\int_0^t \sqrt{1 + (f')^2}\, dx = t + \frac{2}{3}at^3 + O(t^4)$$

Similarly, we can also see that segment of the line connecting the point t to the origin is equal in length to both the curve length and to t up to some terms of order 3.

In general, we can take a section of the manifold by a plane through t and the origin, such that the plane intersects the manifold at a curve. It is clear that the length of the geodesic is between the lengths of the section curve and the line segment connecting t to the origin and therefore must also be equal to distance plus order 3 terms. □

On a Riemannian manifold we have the exponential map from the tangent space at a point to the manifold $\exp_{\mathbf{x}} : T\mathcal{M}^n_{\mathbf{x}} \to \mathcal{M}$. This provides a local coordinate

chart (exponential normal coordinates) for a neighborhood of zero. Note that since $\mathcal{M}^n$ is isometrically embedded in $\mathbb{R}^N$ we can identify $T\mathcal{M}^n_0$ with the corresponding n-dimensional linear subspace of $\mathbb{R}^N$. The exponential map is one-to-one, so the inverse is locally well-defined.

Lemma 4.2.2. *If $\mathbf{y} \in T\mathcal{M}^n_0$ is a point in the tangent space*

$$\|\exp_0(\mathbf{y})\| = \|\mathbf{y}\| + O(\|\mathbf{y}\|^3)$$

Proof. Follows straightforwardly from the previous lemma and the definition of the exponential map. □

Given a function $f : \mathcal{M}^n \to \mathbb{R}$, we denote by $\tilde{f}$ its representation in the exponential coordinates $\tilde{f}(\mathbf{y}) = f(\exp(\mathbf{y}))$.

Claim 4.2.3. *In the exponential normal coordinates $\mathbf{y} = (y_1, \ldots, y_n) \in T\mathcal{M}_0$ centered at 0 the manifold Laplacian can be written as*

$$\mathcal{L}f(0) = \Delta_{R^n}\tilde{f}(0)$$

where 0 on the left-hand side is a point on the manifold and 0 on the right is the origin of the tangent space to $\mathcal{M}^n$ at 0. However they are the same point in $\mathbb{R}^N$.

Proof. Rosenberg, page 90. □

Corollary 4.2.4.

$$\mathcal{L}f(\mathbf{x}) = -\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\int_{B_\epsilon} e^{-\frac{\|\mathbf{y}\|^2}{4t}}\tilde{f}(\mathbf{y})d\mathbf{y}\right)\Bigg|_0$$

Lemma 4.2.5. *For a sufficiently small open set $B \subset \mathcal{M}^n$, $0 \in B$*

$$\int_{B\subset\mathcal{M}^n} f(\mathbf{x})d\mu_{\mathbf{x}} = \int_{\exp_0^{-1}(B)\subset T\mathcal{M}^n} \tilde{f}(\mathbf{y})\det((d\exp_0)(\mathbf{y}))d\mathbf{y}$$

Proof. Put $\mathbf{x} = \exp_0(\mathbf{y})$. We have d $\mathbf{x} = \det(d \exp_0) d\mathbf{y}$ and the formula follows the change of variable rule and the fact that exp is locally one-to-one. □

Claim 4.2.6. *In exponential coordinates*

$$\det(d \exp_0^{-1}(\mathbf{y})) = 1 - \frac{1}{6} s(0) (\sum_i y_i^2) + \textit{higher order terms}$$

where $s(0)$ is the scalar curvature of $\mathcal{M}$ at 0.

Proof. E.g., Do Carmo, 1992, Chapter 5, Section 2. □

Lemma 4.2.7.

$$f(0) = \lim_{t \to 0} (4\pi t)^{-\frac{n}{2}} \int_{B \in R^n} e^{-\frac{\phi(\mathbf{y})^2}{4t}} f(\mathbf{y})\, d\mathbf{y} \tag{4.3}$$

Moreover

$$-\frac{d}{dt} \left((4\pi t)^{-\frac{n}{2}} \int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}} f(\mathbf{x}) d\mu_{\mathbf{x}} \right) \Bigg|_0 = \mathcal{L} f(0) - \frac{1}{3} n s(0) f(0) + C f(0) \tag{4.4}$$

where s(0) is the scalar curvature of $\mathcal{M}^n$ at 0 and C is a constant. $\|\mathbf{x}\|$ denotes the norm in the ambient space $\mathbb{R}^N$.

Proof. We first note that by a standard argument along the lines of Lemma we can integrate over a sufficiently small open set B instead of the whole manifold.

Formula (4.3) follows easily from (4.1) and the argument below, therefore we proceed to prove formula (4.4).

From Lemma 4.2.5 we obtain:

$$\int_B e^{-\frac{\|x\|^2}{4t}} f(\mathbf{x}) d\mu_{\mathbf{x}} = \int_{\tilde{B} \subset T\mathcal{M}^n} e^{-\frac{\|\exp_0(\mathbf{y})\|^2}{4t}} \tilde{f}(\mathbf{y}) \det((d \exp_0)(\mathbf{y})) d\mathbf{y}$$

where $\tilde{B} = \exp_0^{-1}(B)$.

From Lemma 4.2.2 and Lemma 4.1.4 we see that

$$-\frac{d}{dt} \left((4\pi t)^{-\frac{n}{2}} \int_{\tilde{B} \subset T\mathcal{M}^n} e^{-\frac{\|\exp_0(\mathbf{y})\|^2}{4t}} \tilde{f}(\mathbf{y}) \det((d \exp_0)(\mathbf{y})) d\mathbf{y} \right) \Bigg|_0 \tag{4.5}$$

$$= \Delta(\tilde{f}\det((d\exp_0)(\mathbf{y}))) + C\tilde{f}(0)(\det(d\exp_0))(0)$$

As we have seen in the previous lemma $\det(d\exp_0^{-1}(\mathbf{y})) = 1 - \frac{1}{6}s(0)(\sum_i y_i^2) + o(\mathbf{y}^2)$. Thus $(\det(d\exp_0))(0) = 1$. We have $\Delta(\det(d\exp_0^{-1}(\mathbf{y})) = \Delta\left(1 - \frac{1}{6}s(0)(\sum_i y_i^2) + o(\mathbf{y}^2)\right)(0) = \frac{1}{3}ns(0)$. Thus it is easy to see that

$$\Delta\left(\tilde{f}(\mathbf{y})(1 - \frac{1}{6}s(0)\sum_i y_i^2 + o(\mathbf{y}^2))\right)(0) = \Delta\tilde{f}(0) - \frac{1}{3}ns(0)\tilde{f}(0)$$

Combining this equation with formula (4.5) and recalling that $\mathcal{L}f(0) = \Delta_{R^n}\tilde{f}(0)$ and $\tilde{f}(0) = f(0)$ complete the proof. □

Corollary 4.2.8. *For some constant* β

$$-\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}}\beta d\mu_{\mathbf{x}}\right)\Bigg|_0 = -\frac{1}{3}s(0)\beta + C\beta$$

Proof. Follows immediately from the previous lemma applied to the constant function β and noting that $\mathcal{L}(\beta) \equiv 0$. □

Theorem 4.2.9.

$$(4\pi)^{-\frac{n}{2}}\lim_{t\to 0} t^{-(\frac{n}{2}+1)}\left(\int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}}f(\mathbf{x})\,d\mu_{\mathbf{x}} - \int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}}f(0)\,d\mu_{\mathbf{x}}\right) = \mathcal{L}f(0)$$

Proof. Using the previous corollary with $\beta = f(0)$ and subtracting it from the formula (4.4), we see that

$$\begin{aligned}\frac{\partial}{\partial t}\left(-(4\pi t)^{-\frac{n}{2}}\textstyle\int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}}f(\mathbf{x})d\mu_{\mathbf{x}}\right)\Bigg|_0 \\ + \frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\textstyle\int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}}f(0)\,d\mu_{\mathbf{x}}\right)\Bigg|_0 = \mathcal{L}f(0)\end{aligned}$$

From the definition of derivative and (4.3) one immediately obtains:

$$\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}}\int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}}f(\mathbf{x})\,d\mu_{\mathbf{x}}\right)\Bigg|_0$$

$$= \lim_{t\to 0} \frac{1}{t}\left((4\pi t)^{-\frac{n}{2}} \int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}} f(\mathbf{x})\, d\mu_{\mathbf{x}} - f(0)\right)$$

Similarly

$$\frac{\partial}{\partial t}\left((4\pi t)^{-\frac{n}{2}} \int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}} f(0)\, d\mu_{\mathbf{x}}\right)\Bigg|_0$$

$$= \lim_{t\to 0} \frac{1}{t}\left((4\pi t)^{-\frac{n}{2}} \int_{\mathcal{M}^n} e^{-\frac{\|\mathbf{x}\|^2}{4t}} f(0)\, d\mu_{\mathbf{x}} - f(0)\right)$$

Taking the difference proves the theorem. □

Claim 4.2.10.

$$\int_{B_{\mathbf{x}}} f(\mathbf{y}) e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}} d\mu_{\mathbf{y}} = \int_{\exp_{\mathbf{x}}^{-1}(B)} f(\exp_{\mathbf{x}}(\mathbf{y})) e^{-\frac{\|\mathbf{x}-\exp(\mathbf{y})\|^2}{4t}} \det(d\exp_{\mathbf{x}}^{-1}(\mathbf{y})) d\mathbf{y}$$

Claim 4.2.11.

$$\mathcal{L}f(\mathbf{x}) - \frac{1}{3} n s(\mathbf{x}) f(\mathbf{x}) + C f(\mathbf{x}) = -\frac{\partial}{\partial t}\left[(4\pi t)^{-\frac{n}{2}} \int_{B\subset\mathcal{M}} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}} f(\mathbf{y}) d\mu\right](0)$$

Claim 4.2.12. *The integrals we are evaluating for computing the graph Laplacian are:*

$$Wf(\mathbf{x}) \approx \int_B e^{-\frac{\|\mathbf{y}-\mathbf{x}\|}{4t}} f(\mathbf{y}) d\mu$$

and

$$Df(\mathbf{x}) \approx f(\mathbf{x}) \int_B e^{-\frac{\|\mathbf{y}-\mathbf{x}\|}{4t}} d\mu$$

where the approximation can be made arbitrarily good given sufficiently many points.

Corollary 4.2.13.

$$\mathcal{L}f(\mathbf{x}) \approx \frac{1}{3} n f(\mathbf{x}) s(\mathbf{x}) + C f(\mathbf{x}) - \frac{(4\pi t)^{-\frac{n}{2}}\left[\int_B e^{-\frac{\|\mathbf{y}-\mathbf{x}\|}{4t}} f(\mathbf{y}) d\mu - f(\mathbf{x})\right]}{t}$$

also

$$0 = f(\mathbf{x})\mathcal{L}\mathbf{1} \approx \frac{1}{3}nf(\mathbf{x})s(\mathbf{x}) + Cf(\mathbf{x}) - f(\mathbf{x})\frac{(4\pi t)^{-\frac{n}{2}}\left[\int_B e^{-\frac{\|\mathbf{y}-\mathbf{x}\|}{4t}}d\mu - 1\right]}{t}$$

Therefore by subtracting these equalities we get

$$\mathcal{L} \approx \frac{(4\pi t)^{-\frac{n}{2}}[Df - Lf]}{t}$$

APPENDIX A
FIGURES

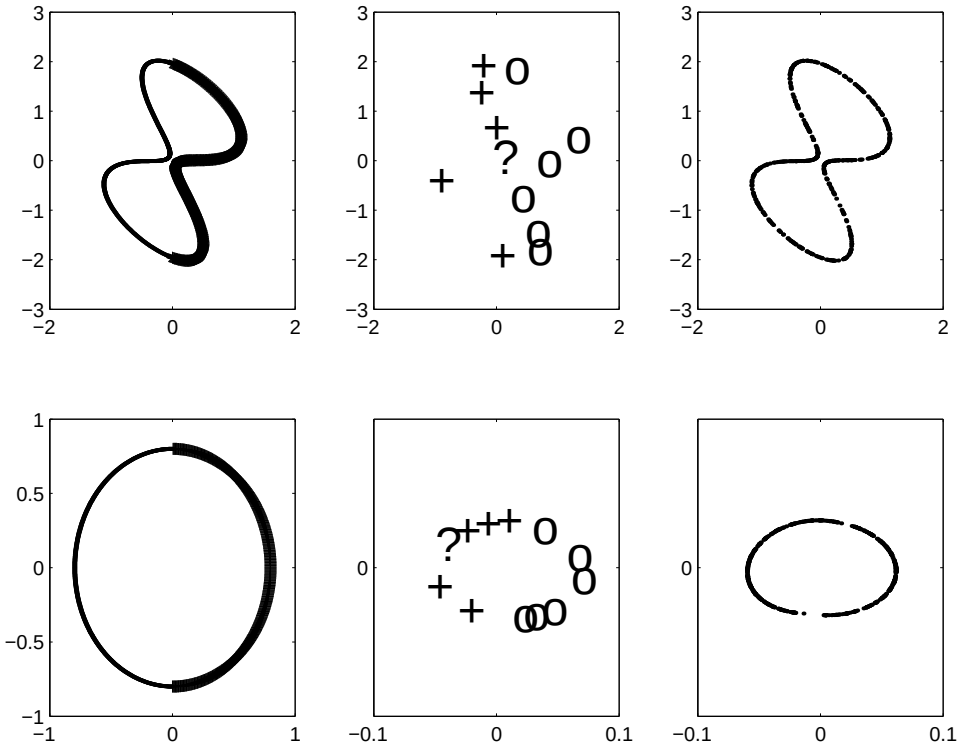

Figure A.1: Top row: Panel 1. Two classes on a plane curve. Panel 2. Labeled examples. "?" is a point to be classified. Panel 3. 500 random unlabeled examples. Bottom row: Panel 4. Ideal representation of the curve. Panel 5. Positions of labeled points and "?" after applying eigenfunctions of the Laplacian. Panel 6. Positions of all examples.

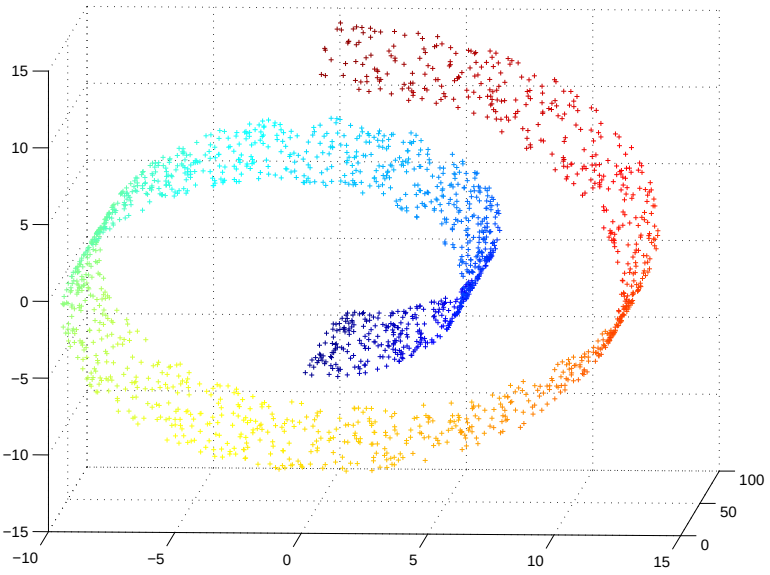

Figure A.2: 2000 random data points on the "swiss roll".

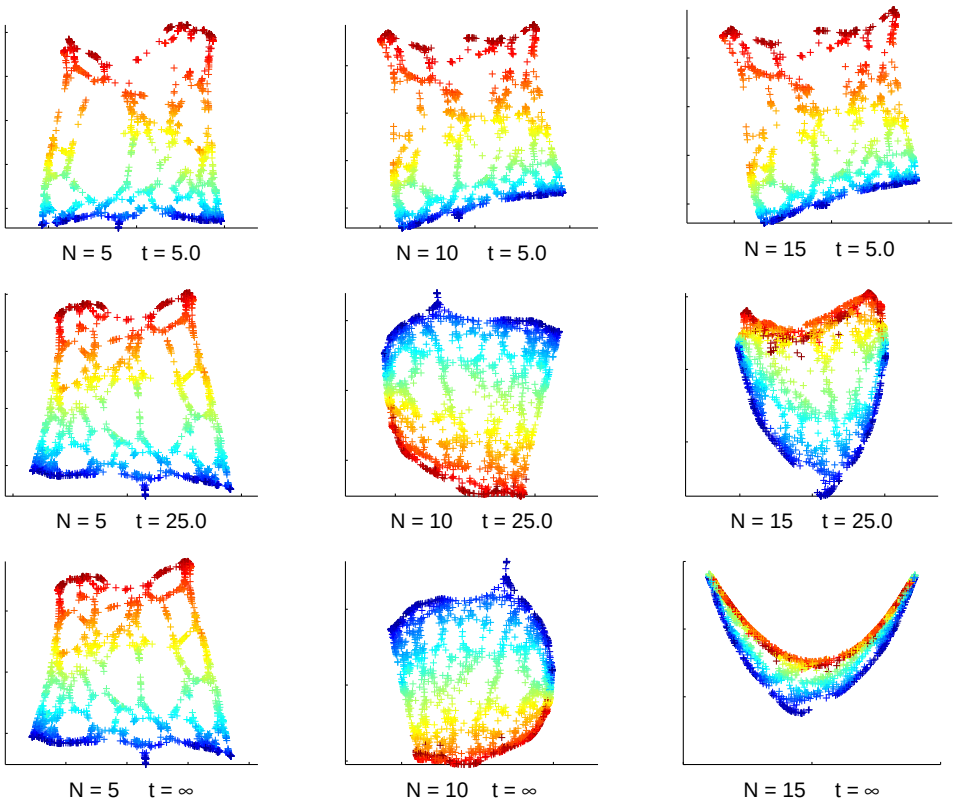

Figure A.3: Two-dimensional representations of the "swiss roll" data, for different values of the number of nearest neighbors N and the heat kernel parameter t. $t = \infty$ corresponds to the discrete weights.

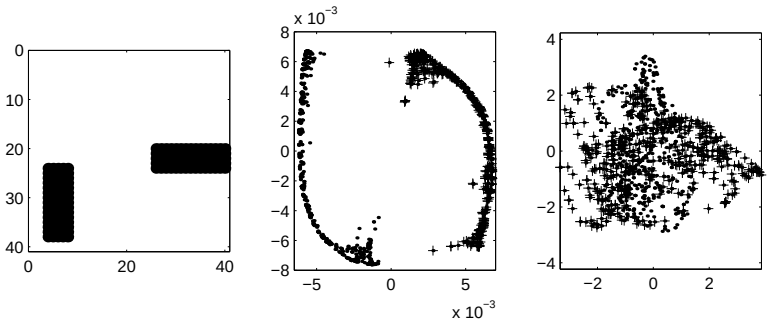

Figure A.4: The left panel shows a horizontal and a vertical bar. The middle panel is a two dimensional representation of the set of all images using the Laplacian eigenmaps. The right panel shows the result of a principal components analysis using the first two principal directions to represent the data. Dots correspond to images of vertical bars and '+' signs correspond to images of horizontal bars.

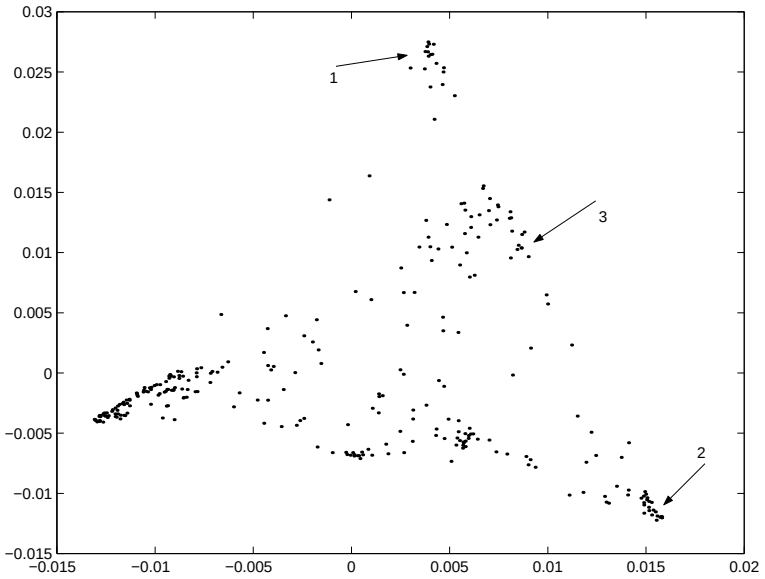

Figure A.5: 300 most frequent words of the Brown corpus represented in the spectral domain.

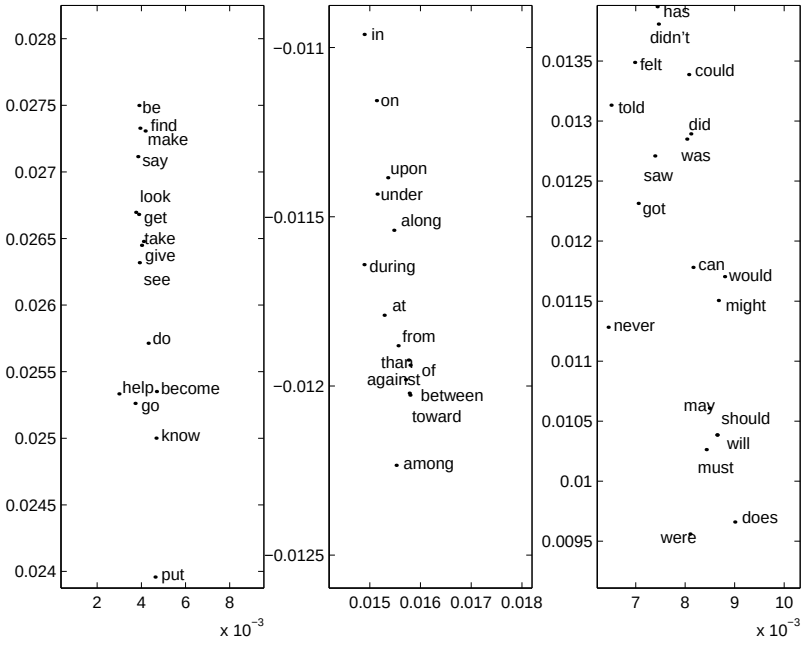

Figure A.6: Fragments labelled by arrows, from left to right. The first is exclusively infinitives of verbs, the second contains prepositions and the third mostly modal and auxiliary verbs. We see that the syntactic structure is well-preserved.

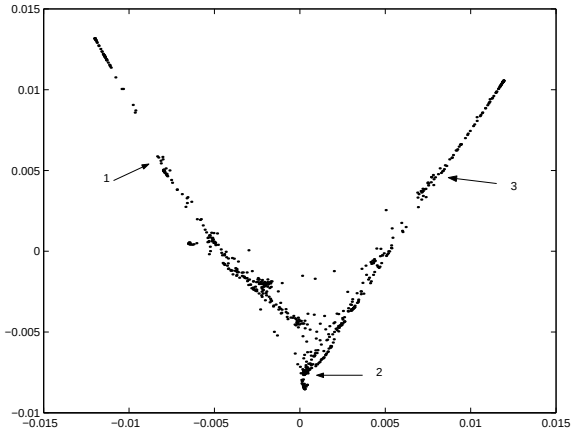

Figure A.7: 685 speech data points plotted in the two dimensional Laplacian spectral representation.

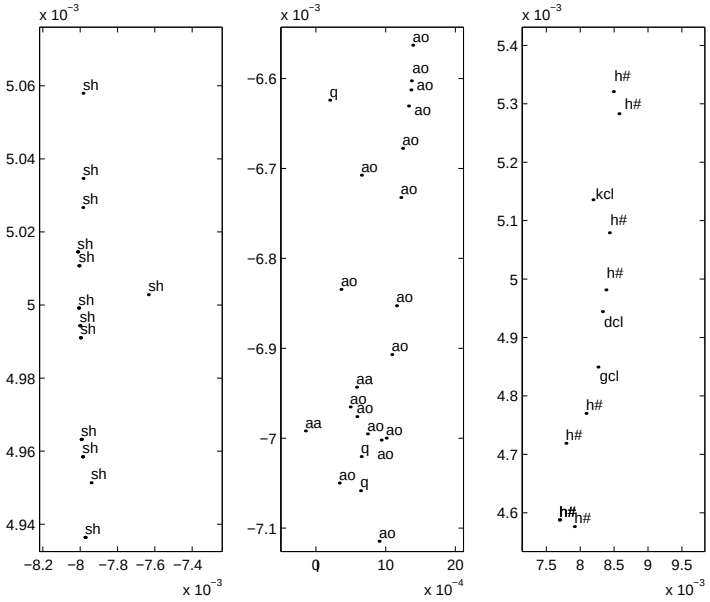

Figure A.8: A blowup of the three selected regions (1,2,3) left to right. Notice the phonetic homogeneity of the chosen regions. The data points corresponding to the same region have similar phonetic identity though they may (and do) arise from occurrences of the same phoneme at different points in the utterance. The symbol "sh" stands for the fricative in the word *she*; "aa","ao" stand for vowels in the words *dark* and *all* respectively; "kcl","dcl","gcl" stand for closures preceding the stop consonants "k","d","g" respectively. "h#" stands for silence.

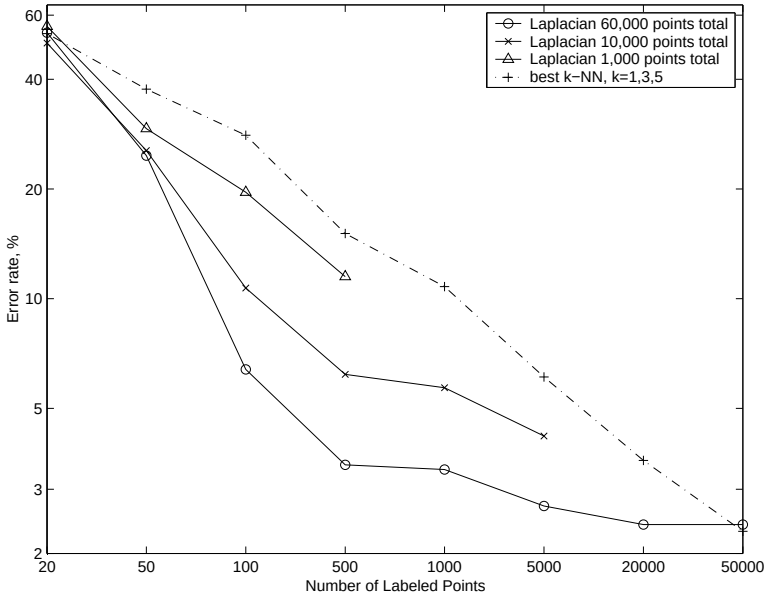

Figure A.9: MNIST data set. Percentage error rates for different numbers of labeled and unlabeled points compared to best k-NN base line.

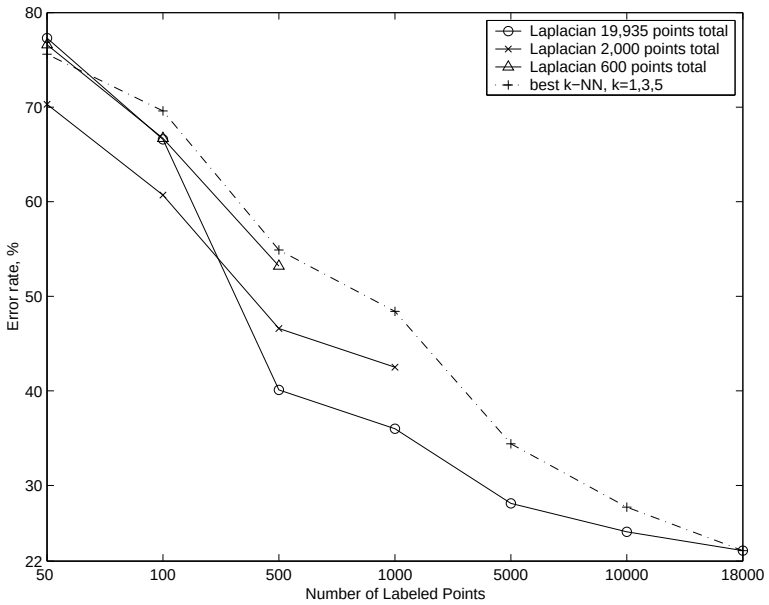

Figure A.10: 20 Newsgroups data set. Error rates for different numbers of labeled and unlabeled points compared to best k-NN baseline.

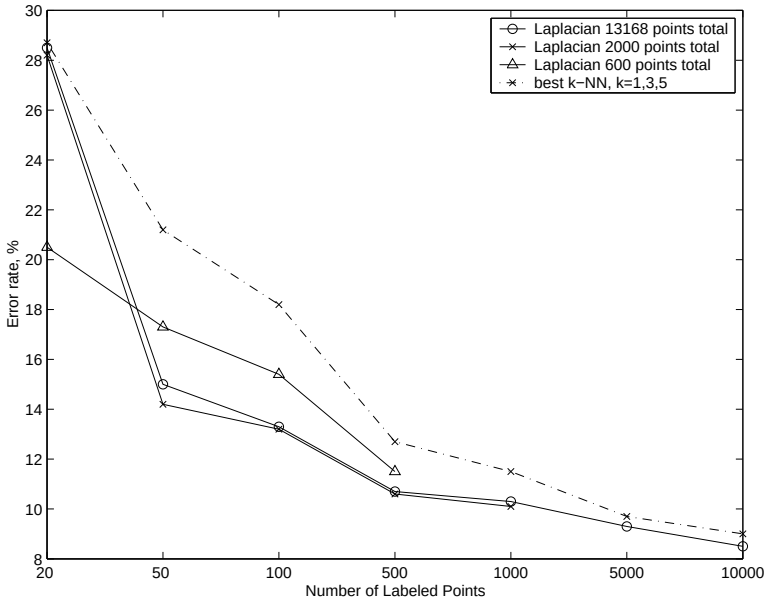

Figure A.11: TIMIT dataset. Error rates for different numbers of labeled and unlabeled points compared to best k-NN baseline.

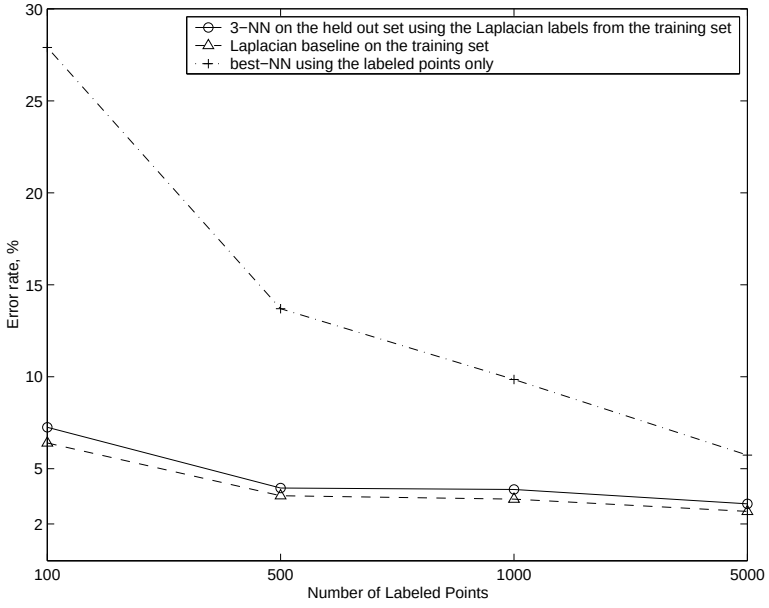

Figure A.12: Results on the held out data set. Randomly chosen labeled are used to label the rest of the 60000 point training set using the Laplacian classifier. Then a 3-NN classifier is applied to the held out 10000 point test set.

REFERENCES

[1] M. Belkin, P. Niyogi, *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*, Neural Computation, June 2003; 15 (6):1373-1396.

[2] Belkin, M., Niyogi, P. (2002). *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering*, Advances in Neural Information Processing Systems (NIPS), vol 14.

[3] Belkin, M., Niyogi, P. (2003). *Semi-supervised Learning on Manifolds*, Machine Learning Journal, Special Issue on Clustering, to appear.

[4] Bernstein, M., de Silva, V., Langford, J.C., Tenenbaum, J.B. (2000). *Graph Approximations to Geodesics on Embedded Manifolds*, Preprint, available at http://isomap.stanford.edu/BdSLT.pdf

[5] A. Blum, S. Chawla, *Learning from Labeled and Unlabeled Data using Graph Mincuts*, ICML, 2001,

[6] Bousquet, O., A. Elisseeff, *Algorithmic Stability and Generalization Performance*. Advances in Neural Information Processing Systems 13, 196-202, MIT Press (2001),

[7] V. Castelli, T. M. Cover, *On the Exponential Value of Labeled Samples,* Pattern Reconition Letters, 16 (1995),

[8] J.Cheeger, *A Lower Bound for the Smallest Eigenvalue of the Laplacian,* Problems in Analysis (R.C. Gunnings, ed), Princeton University Press, 1970,

[9] Chung, Fan R. K. (1997). *Spectral Graph Theory,* Regional Conference Series in Mathematics, number 92.

[10] Do Carmo, M. (1992). *Riemannian Geometry,* Birkhauser.

[11] Chung, Fan R. K., Grigor'yan, A., Yau, S.-T., *Higher eigenvalues and isoperimetric inequalities on Riemannian manifolds and graphs,* Communications on Analysis and Geometry, to appear,

[12] Hadley, S.W. et al, (1992). *An efficient eigenvector approach for finding netlist partitions. IEEE Transactions on Computer-Aided Design,* 11(7):885-892.

[13] Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation* Prentice Hall.

[14] Indyk, P. (2001). *Dimensionality Reduction Techniques for Proximity Problems*, 11th Symposium on Discrete Algorithms.

[15] Kondor, R.I., Lafferty, J. (2002). *Diffusion Kernels on Graphs and Other Discrete Input Spaces*, Proceedings of ICML 2002,

[16] Hendrickson, B, Leland, R. (1993). *Multidimensional Spectral Load Balancing, short version in Proc.*, 6th SIAM Conf. Parallel Proc., 953-961.

[17] C. McDiarmid, *Concentration*, chapter in Probabilistic Methods for Algorithmic Discrete Mathematics (Habib, McDiarmid,Ramirez,Reed (eds.)), Springer-Verlag, 1998,

[18] A.Y. Ng, M. Jordan, Y. Weiss, *On Spectral Clustering: Analysis and an Algorithm*, Advances in Neural Information Processing Systems (NIPS) 2002, vol 14.,

[19] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, *Text Classification from Labeled in Unlabeled Data*, Machine Learning 39(2/3), 2000

[20] Nash, J. (1954). C^1 *Isometric Imbeddings*, Annals of Mathematics, 56.

[21] Nash, J. (1956). *The Imbedding Problem for Riemannian Manifolds*, Annals of Mathematics, 63.

[22] Rosenberg, S. (1997). *The Laplacian on a Riemmannian Manifold,* Cambridge University Press.

[23] Roweis, S.T., Saul, L.K. (2000). *Nonlinear Dimensionality Reduction by Locally Linear Embedding,* Science, vol 290.

[24] Schoelkopf, B., Smola, A., Mueller, K.-R. (1998). *Nonlinear Component Analysis as a Kernel Eigenvalue Problem,* Neural Computation, Vol. 10(5).

[25] Martin Szummer, Tommi Jaakkola, *Partially labeled classification with Markov random walks*, Neural Information Processing Systems (NIPS) 2002, vol 14.,

[26] de Silva, V., Tenenbaum, J. (2002). *Unsupervised Learning of Curved Manifolds,* Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification.

[27] A. N. Tikhonov, V. Y. Arsenin, *Solutions of Ill-posed Problems*, W. H. Winston, Washington, D.C. 1977.

[28] Simon, H.D. (1991). *Partitioning of Unstructured Problems for Parallel Processing,* Computing Systems in Engineering,2,pp.135-148.

[29] Shi, J., Malik, J. (2000). *Normalized Cuts and Image Segmentation,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 22, no 8.

[30] F. Cucker, S. Smale, (2002), *On the Mathematical Foundations of Learning*, Bull. of the American Math. Society, vo; 39, no 1.

[31] Tenenbaum, J., de Silva, V., Langford, J. (2000). *A Global Geometric Framework for Nonlinear Dimensionality Reduction,* Science, Vol 290.

[32] Grace Wahba, *Spline Models for Observational Data,* Society for Industrial and Applied Mathematics, 1990.