Semi-supervised learning on manifolds

Mikhail Belkin[*] Partha Niyogi[†]

November 29, 2002

Abstract

We consider the general problem of utilizing both labeled and unlabeled data to improve classification accuracy. Under the assumption that the data lie on a submanifold in a high dimensional space, we develop an algorithmic framework to classify a partially labeled data set in a principled manner. The central idea of our approach is that classification functions are naturally defined only on the submanifold in question rather than the total ambient space. Using the Laplace Beltrami operator one produces a basis for a Hilbert space of square integrable functions on the submanifold. To recover such a basis, only unlabeled examples are required. Once such a basis is obtained, training can be performed using the labeled data set.

Our algorithm models the manifold using the adjacency graph for the data and approximates the Laplace Beltrami operator by the graph Laplacian. We provide details of the algorithm, its theoretical justification, and several practical applications for image, speech, and text classification.

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

[*]University of Chicago, Department of Mathematics, misha@math.uchicago.edu

[†]University of Chicago, Departments of Computer Science and Statistics, niyogi@cs.uchicago.edu

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 ([3, 4, 10, 14]).

In this paper we address the problem of classifying a partially labeled set by developing the ideas proposed in [1] 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 ([12, 15]), the only other application to machine learning, that we are aware of, was in [14], where the authors use a random walk on the adjacency graph for partially labeled classification.

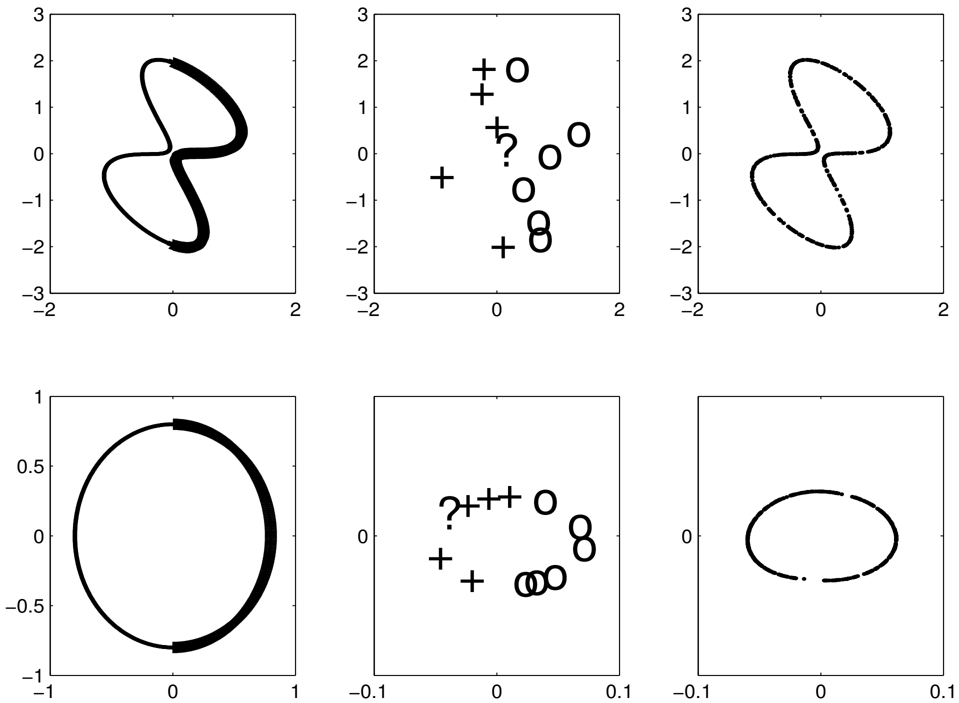

Figure 1: Top row: Panel 1. Two classes on a plane curve. Panel 2. Labeled examples. "?" is a point to be classified. Panel 3. 500 random unlabeled examples. Bottom row: Panel 4. Ideal representation of the curve. Panel 5. Positions of labeled points and "?" after applying eigenfunctions of the Laplacian.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

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 $\{p(C_1 \mid x \in \mathcal{X})\}$, $\{p(C_2 \mid x \in \mathcal{X})\}$. 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.

In this paper 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 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 [1], 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 paper.

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 neighbours 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 [15]).

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 [14]. We also note the approach taken in [3] which uses mincuts of certain graphs for partially labeled classifications.

3.1 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 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, $\Delta e_i = \lambda_i e_i$.

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 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 Lapla-

[1] There is an extensive literature on the connection between the geometric properties of the manifold and the Laplace-Beltrami operator. See[11] for an introduction to the subject.

cian 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.

4 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 the heat kernel see [1]. 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 fre-

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

quency 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 $\mathbf{c} = (c_1, \dots, c_s)$ and

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

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.

5 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].

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

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

It can be shown (e.g., [11]) 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

[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.

$\{1^2, 2^2, \ldots\}$. 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 basis, it also yields a measure of smoothness for functions on the manifold.

5.2 The Laplacian as a smoothness functional

A simple measure of the degree of smoothness (following the theory of splines, for example, [16]) 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

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

[5]Stronger conditions are needed for pointwise convergence.

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 [16]).

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:

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

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

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

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 $\mathbf{y} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$.

5.3 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 [6]). 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 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

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

[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 [6].

Here $\mathrm{vol}(V_l) = \sum\limits_{\substack{i,j \in V_l \\ i \sim j}} w_{ij}$, $l \in \{1,2\}$. 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 [5], 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.

6 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.

6.1 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 1. Each row 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 paper 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

[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.

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. 2 is a summary plot of classification accuracy on the unlabeled set comparing the nearest neighbors baseline with our algorithm that retains the number 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.

6.2 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,

[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.

Labeled points	Number of Eigenvectors								best k-NN
	5	10	20	50	100	200	500	1000	
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 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.

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

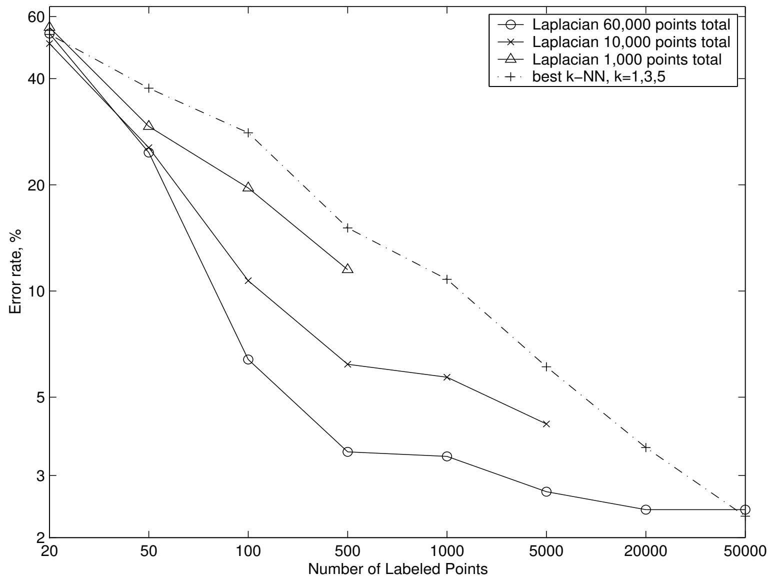

Figure 2: MNIST data set. Percentage error rates for different numbers of labeled and unlabeled points compared to best k-NN base line.

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 2

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. 3 we summarize the results by taking 19935, 2000 and 600 total points respectively and calculating the error rate for different numbers of

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

Labeled points	Number of Eigenvectors									best k-NN
	5	10	20	50	100	200	500	1000	2000	
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 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.

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.

6.3 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 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 and fig. 4. 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.

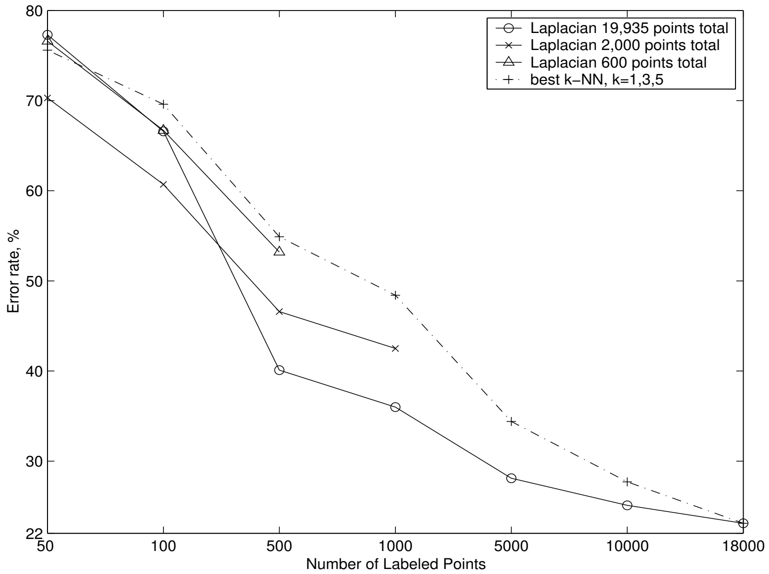

Figure 3: 20 Newsgroups data set. Error rates for different numbers of labeled and unlabeled points compared to best k-NN baseline.

6.4 Improving Existing Classifiers With Unlabeled Data

In this paper 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 5 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

Labeled points	Number of Eigenvectors								best k-NN
	5	10	20	50	100	200	500	1000	
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: 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.

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.

7 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 paper 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.

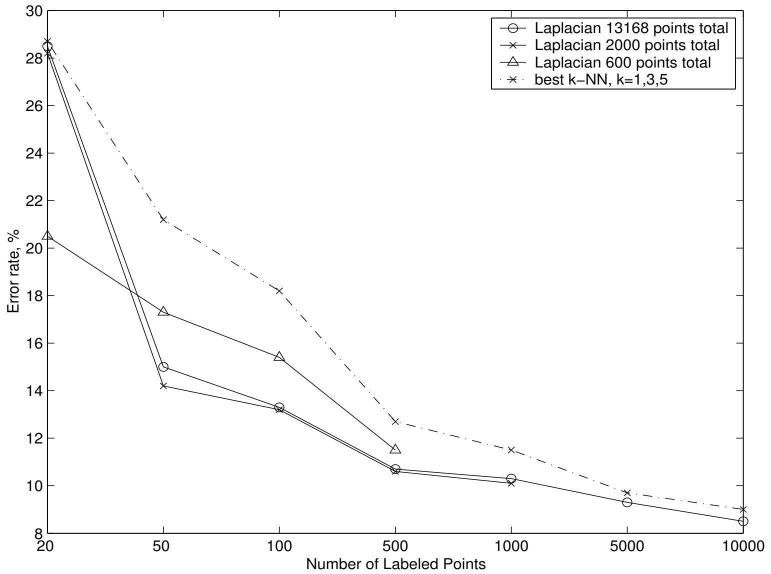

Figure 4: TIMIT dataset. Error rates for different numbers of labeled and unlabeled points compared to best k-NN baseline.

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 paper 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.

Acknowledgments:

We are grateful to Yali Amit for a number of conversations and helpful suggestions over the course of this work. We are also grateful to Dinoj Surendran for preprocessing the TIMIT Database for our phonemic experiments.

References

[1] M. Belkin, P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduc-

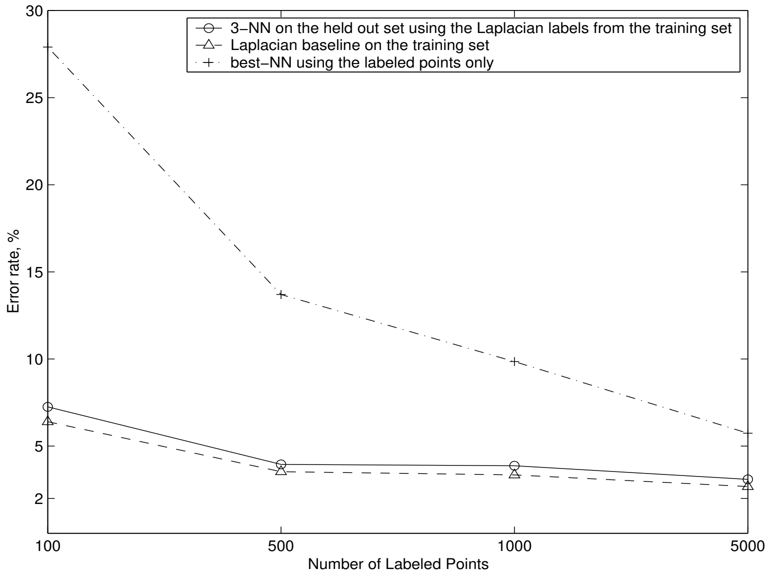

Figure 5: Results on the held out data set. Randomly chosen labeled are used to label the rest of the 60000 point training set using the Laplacian classifier. Then a 3-NN classifier is applied to the held out 10000 point test set.

tion and Data Representation, Technical Report, TR-2002-01, Department of Computer Science, The University of Chicago, 2002.

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

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

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

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

[6] Fan R. K. Chung, Spectral Graph Theory, Regional Conference Series in Mathematics, number 92, 1997

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

[8] Simon Haykin, Neural Networks, A Comprehensive Foundation Prentice Hall, 1999,

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

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

[11] S. Rosenberg, The Laplacian on a Riemmannian Manifold, Cambridge University Press, 1997,

[12] Sam T. Roweis, Lawrence K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, vol 290, 22 December 2000,

[13] Schlkopf, B., Smola, A., Mlller, K.-R., Nonlinear Component Analysis as a Kernel Eigenvalue Problem, Neural Computation, Vol. 10(5), 1998,

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

[15] Joshua B. Tenenbaum, Vin de Silva, John C. Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science, Vol 290, 22 December 2000,

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