Machine Learning, 56, 209–239, 2004

Semi-Supervised Learning on Riemannian Manifolds

MIKHAIL BELKIN misha@math.uchicago.edu
PARTHA NIYOGI niyogi@cs.uchicago.edu
Department of Computer Science, University of Chicago, 1100 E. 58th Street, Chicago, IL 60637, USA

Editors: Nina Mishra and Rajeev Motwani

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 (the Laplacian Eigenmaps) 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.

Keywords: semi-supervised learning, manifold learning, graph regularization, laplace operator, graph laplacian

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 (e.g., Blum & Chawla, 2001; Castelli & Cover, 1995; Nigam et al., 2000; Szummer & Jaakkola, 2002). In particular, there has been a lot of recent interest in semi-supervised learning and graphs, including (Zhu, Lafferty, & Ghahramani, 2003; Zhou et al., 2003; Chapelle, Weston, & Scholkopf, 2003; Joachims, 2003; Belkin, Matveeva, & Niyogi, 2003) and closely related graph kernels and especially the diffusion kernel (Kondor and Lafferty, 2002; Smola & Kondor, 2003).

In this paper we address the problem of classifying a partially labeled set by developing the ideas proposed in Belkin and Niyogi (2003) 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 lie 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 no 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, de Silva, & Langford, 2000), the only other application to classification, that we are aware of, was in Szummer and Jaakkola (2002), where the authors use a random walk on the adjacency graph for partially labeled classification.

There are two central ideas that come together in this paper. First, we utilize the geometry of the underlying space of possible patterns to construct representations, invariant maps, and ultimately learning algorithms. Natural patterns are typically embedded in a very high dimensional space. However, the intuition of researchers has always been that although these patterns are ostensibly high dimensional, there is a low dimensional structure that needs to be discovered. If this low dimensional structure is a linear subspace, then linear projections to reduce the dimensionality are sufficient. Classical techniques like Principal Components Analysis and Random Projections may be used and one can then construct classifiers in the low dimensional space. If on the other hand, the patterns lie on a low dimensional manifold embedded in the higher dimensional space, then one needs to do something different. This paper presents an approach to this situation. Second, we observe that in order to estimate the manifold all that is needed are unlabeled data (the $\mathbf{x}'s$). Once the manifold is estimated, then the Laplace-Beltrami operator may be used to provide a basis for maps intrinsically defined on this manifold and then the appropriate classifier (map) is estimated on the basis of the labeled examples. Thus we have a natural framework for learning with partially labeled examples.

Our framework is fairly general and while we focus in this paper on classification problems, it is noteworthy that one may also construct algorithms for dimensionality reduction and clustering within the same framework (see the discussion in Section 7). The rest of the paper is organized as follows. In Sections 2, 3, and 4, we gradually present the motivation for manifolds and an algorithmic framework for exploiting manifold structure. In Section 5 we

provide the differential geometric underpinnings of the basic framework. Section 6 provides experimental results on partially labeled classification on a number of different data sets and problems. In Section 7, we provide additional perspectives on issues like regularization in general, convergence theorems, dimensionality reduction and clustering. We conclude in Section 8.

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

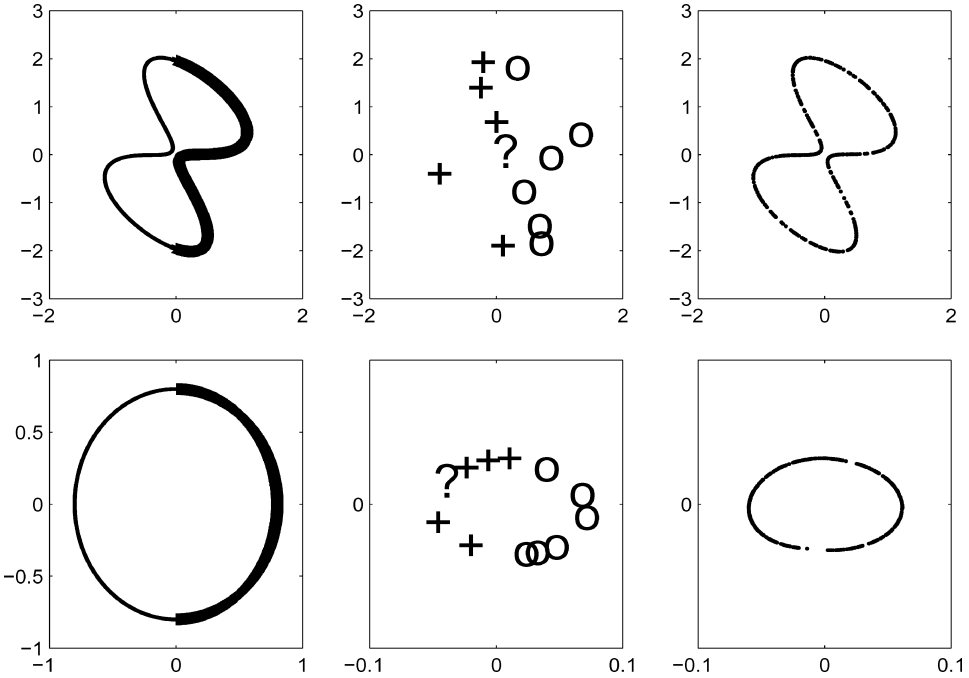

Figure 1. Top row: Panel 1. Two classes on a plane curve. Panel 2. Labeled examples. "?" is a point to be classified. Panel 3. 500 random unlabeled examples. Bottom row: Panel 4. Ideal representation of the curve. Panel 5. Positions of labeled points and "?" after applying eigenfunctions of the Laplacian.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

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 (Belkin & Niyogi, 2003), 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. Thus given a set of labeled examples $((x_i, y_i); x_i \in \mathbb{R}^k, y_i \in Y)$ and a set of unlabeled examples $(x_j \in \mathbb{R}^k)$, the normal impulse is to seek a classifier

$$f : \mathbb{R}^k \to Y$$

Since k is very large, this immediately leads to difficulties due to the "curse of dimensionality". Instead, we exploit the fact that all $x_k \in \mathcal{M}$ where $\mathcal{M}$ is a low dimensional manifold.

Therefore, we construct classifiers of the form

$$f : \mathcal{M} \to Y$$

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 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, de Silva, & Langford, 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 and Jaakkola (2002). We also note the approach taken in Blum and Chawla (2001) 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 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, $\Delta e_i = \lambda_i e_i$.

Now assuming that the data lie 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 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.

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. For our experiments we take the weights to be one. However, see Belkin and Niyogi (2003) for the discussion about the choice of weights, and its connection to the heat kernel. 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

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

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

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

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., 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 $\mathcal{L}^2$ (stronger conditions are needed for pointwise convergence):

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

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, 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

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

The linear subspace, where the smoothness functional is finite is a Reproducing Kernel Hilbert Space (e.g., see Wahba, 1990). We develop this point of view further in Section 7.

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

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 Chung, 1997). 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, $D_{ii} = \sum_j W_{ij}$.[5] $\mathbf{L}$ 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 $\mathbf{e}_1$.

6. Experimental results

The experiments with labeled 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 28×28 grayscale images of handwritten digits.[6] 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

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 column contains the nearest neighbor base line.

	Number of eigenvectors								
Labeled points	5	10	20	50	100	200	500	1000	Best 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

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 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

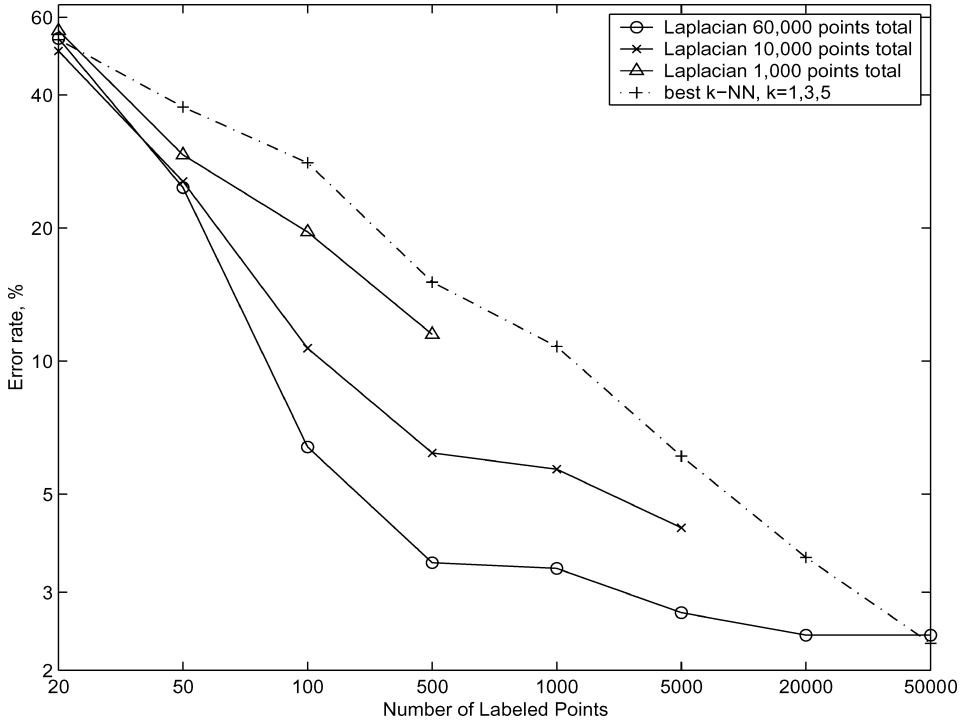

Figure 2. MNIST data set. Percentage error rates for different numbers of labeled and unlabeled points compared to best k-NN base line.

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 figure 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.[7] 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, 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.[8] 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.

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.

	Number of eigenvectors									
Labeled points	5	10	20	50	100	200	500	1000	2000	Best 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

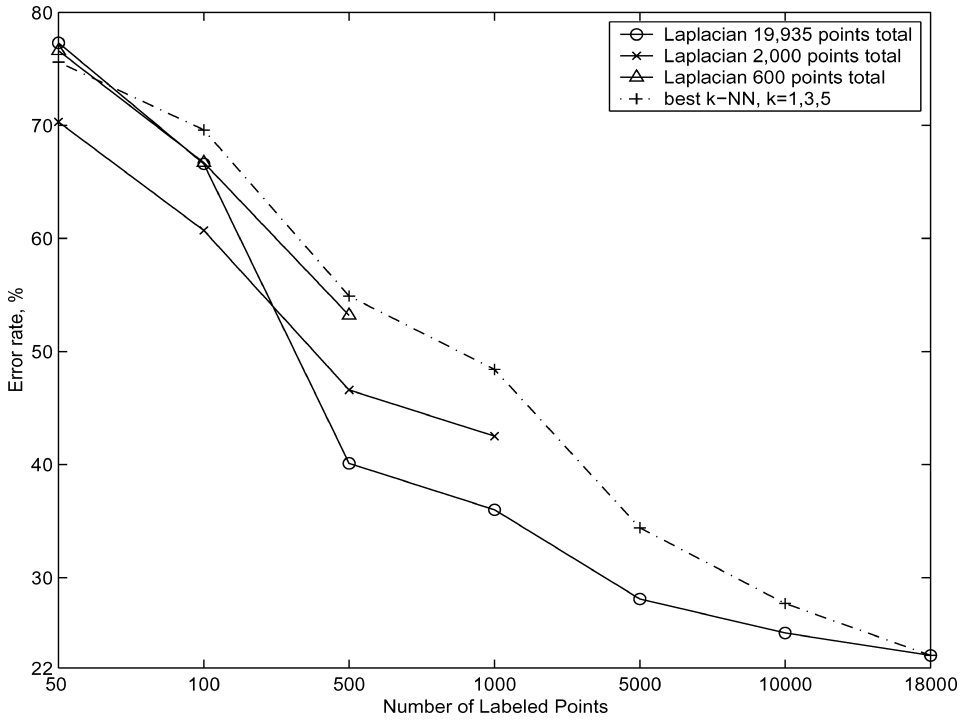

Figure 3. 20 Newsgroups data set. Error rates for different numbers of labeled and unlabeled points compared to best k-NN baseline.

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 figure 3 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 always 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

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.

	Number of eigenvectors								
Labeled points	5	10	20	50	100	200	500	1000	Best 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

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 figure 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.

6.4. Comparison with geodesic nearest neighbors

The simplest semi-supervised learning algorithm that utilizes the manifold structure of the data is arguably the geodesic nearest neighbors (GNN). This is a version of the nearest neighbor algorithm that uses geodesic distances on the data derived adjacency graph instead of Euclidean distances in the ambient space. The graph is computed in the same manner as before. The edge weights are taken to be local Euclidean distances.

In this section, we conduct some experiments to see how our method compares with GNN. Figure 5 shows the results of the comparison between the Laplacian and the best k-GNN, where $k \in \{1, 3, 5\}$. The experiments are on a 10000 point subset of the MNIST hand-written digit dataset. We see that for very small numbers of labeled points (20 total, i.e. approximately 2 per class, as we did not balance the classes) GNN performs slightly better, but once the number of labeled reaches exceeds 100, the Laplacian produces a considerably lower error rate.

It is worth noting that GNN is consistently better than standard nearest neighbors (NN) suggesting that even a naive utilization of manifold structure provides a benefit. This paper provides a more sophisticated way to use manifold structure.

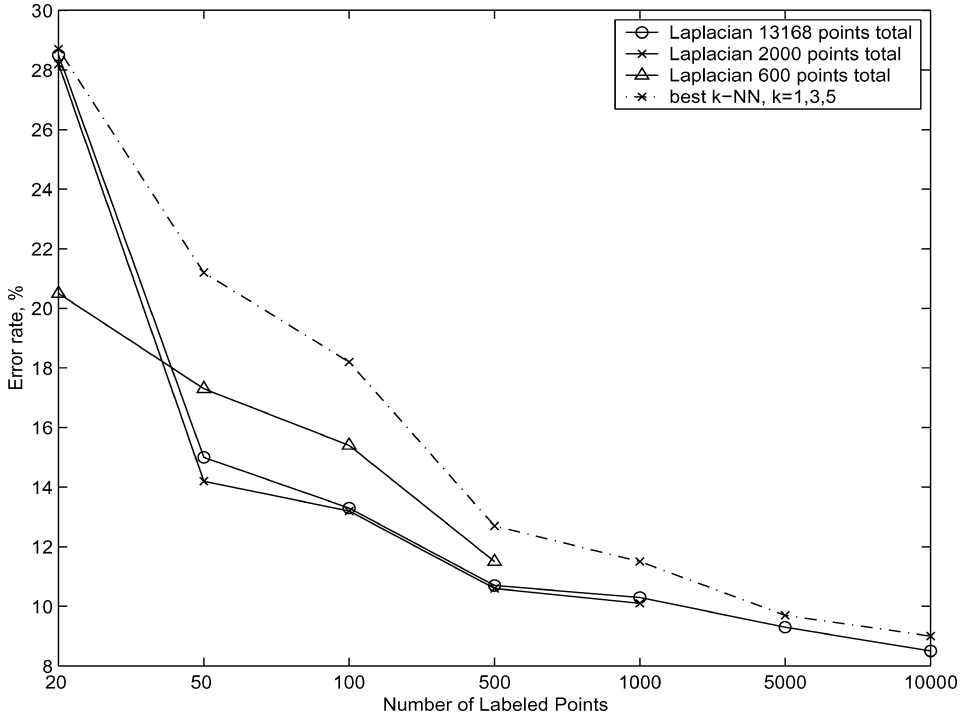

Figure 4. TIMIT dataset. Error rates for different numbers of labeled and unlabeled points compared to best k-NN baseline.

6.5. 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 6 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.[9] 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.

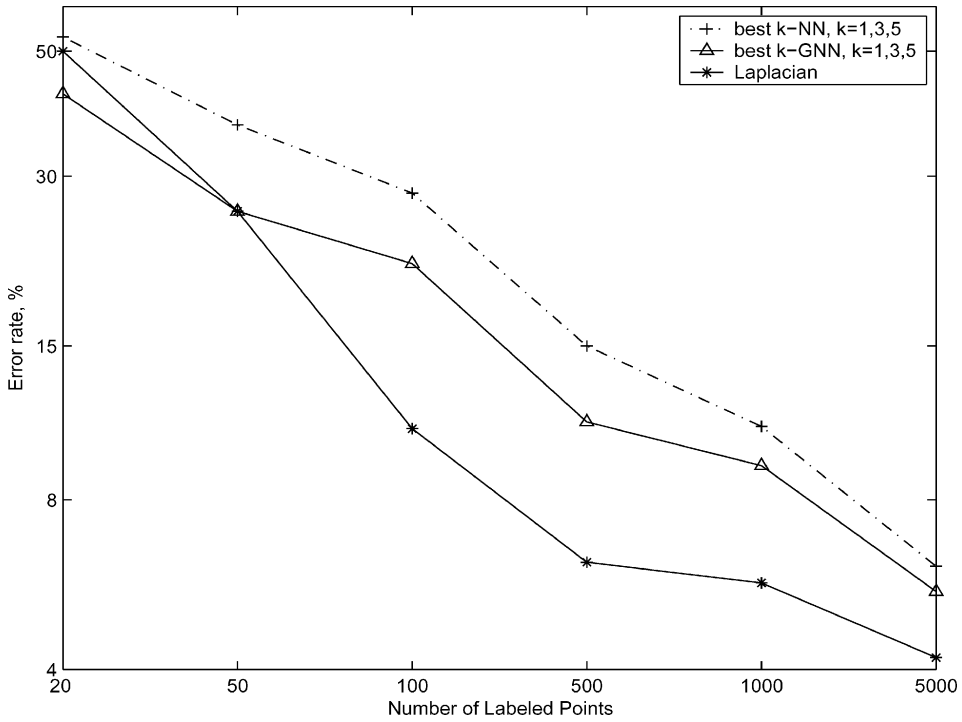

Figure 5. MNIST dataset. Error rates for different numbers of labeled and unlabeled points compared to best k-NN and best k-GNN. The total number of points is 10000.

7. Perspective

Much of this paper has focused on the problem of utilizing labeled and unlabeled examples in a coherent fashion. In this section, we take a broader perspective of the manifold learning framework and make connections to a number of different problems and issues.

7.1. Regularization on manifolds and graphs

We see that the Laplace-Beltrami operator might be used to provide a basis for $\mathcal{L}^2(\mathcal{M})$, the set of square integrable functions on the manifold. In general, one might take various classes of functions that are invariantly defined on the manifold and solve a problem of the following sort

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda P(f) \tag{1}$$

where $H : \mathcal{M} \to \mathbb{R}$.

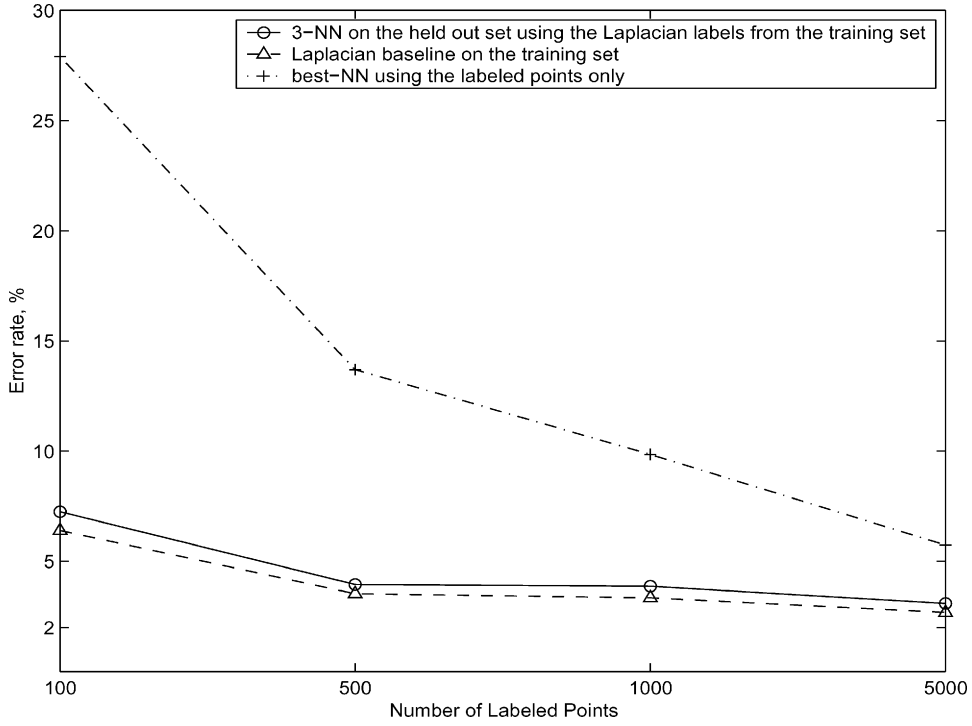

Figure 6. Results on the held out data set. Randomly chosen labeled are used to label the rest of the 60000 point training set using the Laplacian classifier. Then a 3-NN classifier is applied to the held out 10000 point test set.

The first term in the objective function is the empirical risk measured in terms of the least squared error. The second term $P(f)$ is a *stabilizer* that is added within the framework of Tikhonov regularization (Tikhonov & Arsenin, 1977). For example, a simple and natural choice is to take

$$P(f) = \int_{\mathcal{M}} \langle \nabla f, \nabla f \rangle = \sum_i \alpha_i^2 \lambda_i$$

where $f = \sum_i \alpha_i e_i$, the e_i's and the λ_i's are the eigenfunctions and eigenvalues respectively of the manifold Laplacian Δ.

In this setting, one may take H to be the following

$$H = \left\{ f = \sum_i \alpha_i e_i \;\middle|\; P(f) < \infty \right\}$$

It is easy to check that the optimization problem provided by Eq. (1) reduces to a quadratic problem in the α_i's.

Various other choices for $P(f)$ are possible. For example, one may take

$$P(f) = \sum_{i=1}^{k} \|\Delta^i f\|^2$$

where the norm $\|\cdot\|$ is the $\mathcal{L}^2(\mathcal{M})$ norm and Δ^i is the *iterated Laplacian* (iterated i times). It is easy to check that for $f = \sum_j \alpha_j e_j$

$$\sum_{i=1}^{k} \|\Delta^i f\|^2 = \sum_{i=1}^{k} \left\| \sum_j \lambda_j^i \alpha_j e_j \right\|^2 = \sum_{i=1}^{k} \sum_j \lambda_j^{2i} \alpha_i^2$$

Again, the optimization problem reduces to a quadratic problem in the $\alpha_i's$.

These problems are best handled within the framework of regularization where H is an appropriately chosen Reproducing Kernel Hilbert Space (RKHS). A RKHS is a Hilbert space of functions where the evaluation functionals (functionals that simply evaluate a function at a point) $E_x f = f(x)$ are bounded, linear functionals. It is possible to show that for each RKHS there corresponds a kernel $K : X \times X \to \mathbb{R}$ such that

$$f(x) = \langle f(\cdot), K(x, \cdot) \rangle_H$$

where the inner product $\langle \cdot, \cdot \rangle_H$ is the one defined naturally in the RKHS (see Aronszjan, 1950; Wahba, 1990), for more details). In our setting, the domain X is the manifold $\mathcal{M}$ and we are therefore interested in kernels $K : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$. Let us go through the construction of an RKHS that is invariantly defined on the manifold.

Fix an infinite sequence of non-negative numbers $\{\mu_i; i = 1, 2, \ldots\}$ such that $\sum_i \mu_i < \infty$. Now define the following linear space of continuous functions

$$H = \left\{ f = \sum_i \alpha_i f_i \;\middle|\; \sum_i \frac{\alpha_i^2}{\mu_i} < \infty \right\}$$

Define an inner product on this space in the following way. For any two functions $f = \sum_i \alpha_i f_i$ and $g = \sum_j \beta_j f_j$, the inner product is defined as

$$\langle f, g \rangle = \sum_i \frac{\alpha_i \beta_i}{\mu_i}$$

It can be verified that H is a RKHS with the following kernel

$$K(p, q) = \sum_i \mu_i e_i(p) e_i(q)$$

Now one may solve the following optimization problem

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \| f \|_H^2 \tag{2}$$

This problem is well posed and has a unique solution given by

$$f_* = \sum_{i=1}^{n} a_i K(x_i, x)$$

where $\mathbf{a} = (a_1, \ldots, a_n)^T$ is the unique solution of the well posed linear system in $\mathbb{R}^n$ given by

$$(\lambda n I_n + K_{\mathbf{x}})\mathbf{a} = \mathbf{y}$$

In the above expression I_n is the $n \times n$ identity matrix; $K_{\mathbf{x}}$ is a $n \times n$ Gram matrix whose (i, j) element is equal to $K(x_i, x_j)$ and $\mathbf{y} = (y_1, \ldots, y_n)^T$.

If one considers the domain X to be the graph rather than the manifold, then one obtains regularization procedures on the graph very naturally. The algorithm presented in this paper is a special case of this general paradigm.

Remark. Various choices of kernels are possible by choosing μ_i's differently. For example, letting $\mu_i = e^{-t\lambda_i}$ (λ_i's being the eigenvalues of Δ) one obtains the *heat kernel* corresponding to diffusion of heat on the manifold. Alternatively, one might let $\mu_i = 0 \ \forall i > N$. This corresponds to solving the optimization problem in a finite dimensional subspace and is the basis of the algorithms that were actually used in the experiments reported in the previous section. Some of these connections have been explored in a different form by Kondor and Lafferty (2002) in the context of developing kernels on graphs. The diffusion kernels in that work refer to the heat kernel on the graph and its intimate relations with the Laplacian.

7.2. Convergence and sample complexity

It is worthwhile to reflect on the nature of the convergence theorems that are needed to characterize the behavior of the manifold learning algorithms as a function of data. We do this within the context of regularization described previously. A complete technical treatment is beyond the scope of the current paper but we hope to provide the reader with a sense of how one might proceed to provides bounds on the performance of such learning algorithms.

Recall that $H : \mathcal{M} \to \mathbb{R}$ is a RKHS invariantly defined on the manifold $\mathcal{M}$. Then a key goal is to minimize the regularized true risk as follows:

$$E_\lambda = \min_{f \in H} E[(y - f(x))^2] + \lambda \| f \|_H^2 \tag{3}$$

This is what the learner would do if (i) infinite amounts of labeled examples were available and (ii) the manifold $\mathcal{M}$ were known. Note that $E_0 = \min_{f \in H} E[(y - f(x))^2]$ is the best that the learner could possibly do under any circumstances. Thus $\lim_{\lambda \to 0} E_\lambda = E_0$ and the regularization constant λ may be suitably chosen to condition various aspects of the learning problem at different stages.

In reality, of course, infinite amounts of labeled examples are not available but rather only a finite number n of such examples are provided to the learner. On the basis of this, the learner minimizes the *empirical risk* and solves the following optimization problem instead (same as Eq. (2)).

$$\hat{E}_{\lambda,n} = \min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \| f \|_H^2 \tag{4}$$

Now one might ask, how far is $\hat{E}_{\lambda,n}$ from $\hat{E}_\lambda$? In order to get a handle on this question, one may proceed by building on the techniques described in Cucker and Smale (2001), Bousquet and Elisseeff (2001), and Kutin and Niyogi (2002). We only provide a flavor of the kinds of results that may be obtained.

Let $f_{\text{opt}} = \arg\min R(f)$ and $\hat{f}_n = \arg\min R_{\text{emp}}(f)$ where $R(f) = E[(y - f(x))^2] + \lambda \| f \|_H^2$ and $R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \| f \|_H^2$ respectively. Note that for a fixed function f, $R_{\text{emp}}(f)$ is a random variable whose mean is $R(f)$. Let labeled examples be drawn according to a measure μ on $\mathcal{M} \times Y$ where Y is a compact subset of $\mathbb{R}$. Without loss of generality, we may take $Y = [0, M]$. Then the following statements may be made:

Theorem 1. *For the fixed function f_{opt}, the empirical risk $R_{\text{emp}}(f_{\text{opt}})$ converges to the true risk $R(f_{\text{opt}})$ with probability* 1. *Further, with probability $> 1 - \delta$, the following holds*

$$|R_{\text{emp}}(f_{\text{opt}}) - R(f_{\text{opt}})| < M \sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{n}}$$

This is simply a statement of the law of large numbers with a Hoeffding bound on the deviation between the empirical average and the true average of a random variable. One additionally needs to make use of the fact $R(f_{\text{opt}}) < R(\mathbf{0}) < M^2$ where $\mathbf{0}$ denotes the constant function that takes the value 0 everywhere.

A much harder theorem to prove is

Theorem 2. *Let $A = \sup_{p \in \mathcal{M}} K(p, p)$ where $K(p, p) = \sum_i \mu_i f_i^2(p)$. Then with probability $> 1 - \delta$, we have*

$$R(\hat{f}_n) \leq R_{\text{emp}}(\hat{f}_n) + \frac{2M^2A^2}{\lambda n} + \left(\frac{8M^2A^2}{\lambda} + M \right) \sqrt{\frac{\ln(1/\delta)}{2n}}$$

In order to prove this we make use of the notion of algorithmic stability (Bousquet & Elisseeff 2001; Kutin & Niyogi, 2002) and the fact that regularization is uniformly hypothesis stable. The above theorem shows that as n becomes larger and larger, we have that

$R(\hat{f}_n)$ is essentially upper bounded by $R_{\text{emp}}(\hat{f}_n)$. Now one may make use of the following two simple observations:

$$R(f_{\text{opt}}) \leq R(\hat{f}_n) \tag{5}$$

and

$$R_{\text{emp}}(\hat{f}_n) \leq R_{\text{emp}}(f_{\text{opt}}) \tag{6}$$

Inequality 5 holds because f_{opt} minimizes $R(f)$ while inequality 6 holds because $\hat{f}_n$ minimizes $R_{\text{emp}}(f)$.

Putting all of this together, we have with high probability ($>1-2\delta$), the following chain of inequalities

$$R(f_{\text{opt}}) \leq R(\hat{f}_n) \leq R_{\text{emp}}(\hat{f}_n) + B \leq R_{\text{emp}}(f_{\text{opt}}) + B \leq R(f_{\text{opt}}) + B + C \tag{7}$$

where

$$B = \frac{2M^2A^2}{\lambda n} + \left(\frac{8M^2A^2}{\lambda} + M\right)\sqrt{\frac{\ln(1/\delta)}{2n}} \quad \textit{and} \quad C = M\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{n}}.$$

Using this we see that the quantity $|E_\lambda - \hat{E}_{\lambda,n}|$ can be made arbitrarily small and one obtains rates at which this convergence occurs.

All of the above holds when the manifold $\mathcal{M}$ is known. In our situation, the manifold is unknown and needs to be estimated from the *unlabeled* data. In particular, as we have seen, an appropriate basis of functions needs to be estimated. So we solve instead the following problem:

$$\hat{E}_{\lambda,n,m} = \min_{f \in H'} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \|f\|^2_{H'} \tag{8}$$

where n labeled and m unlabeled examples are collected. Now $H' : V \to \mathbb{R}$ is a collection of functions defined on the graph $G = (V, E)$ where the locally connected graph G is constructed in the manner described in earlier sections.

Ideally, one would like to show that $\hat{E}_{\lambda,n,m} \to \hat{E}_{\lambda,n}$ as $m \to \infty$. In other words, as the unlabeled data goes to infinity, regularized solutions on the graph converge to regularized solutions on the manifold. If this were true, then we would find that $\hat{E}_{\lambda,n,m} \to E_\lambda$ as both n (number of labeled examples) and m (number of unlabeled examples) go to infinity.

Preliminary results obtained by the authors suggest that the graph Laplacian converges to the manifold Laplacian in a certain sense. However, the nature of this convergence is not strong enough yet for us to be able to formally claim that $\hat{E}_{\lambda,n,m} \to \hat{E}_{\lambda,n}$ as $m \to \infty$. We leave this as a topic of future research.

7.3. *Dimensionality reduction*

Since we are in a setting where the manifold $\mathcal{M}$ is low dimensional, it is natural to consider the question of whether the data may be embedded in a much lower dimensional space than the original ambient one. As it turns out, the eigenfunctions of the Laplace-Beltrami operator are arranged in increasing order of smoothness. Consequently, the lower eigenmaps may be used for dimensionality reduction. It is worthwhile to reflect on this for a moment.

Suppose we desire a map $f : \mathcal{M} \to \mathbb{R}^m$ that optimally preserves locality, i.e., points nearby on the manifold are mapped to nearby points in the lower dimensional space $\mathbb{R}^m$. The discussion here follows that in Belkin and Niyogi (2003).

Let us first consider mapping 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

$$|f(\mathbf{z}) - f(\mathbf{x})| \leq \text{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{z}) \|\nabla f(\mathbf{x})\| + o(\text{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{z})) \tag{9}$$

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

$$\operatorname*{argmin}_{\|f\|_{L^2(\mathcal{M})}=1} \int_{\mathcal{M}} \|\nabla f(x)\|^2 \tag{10}$$

where the integral is taken with respect to the standard measure on a Riemannian manifold. Note that minimizing $\int_{\mathcal{M}} \|\nabla f(x)\|^2$ 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.

Recall from the discussion in Section 5, that minimizing the objective function of Eq. (10) reduces to finding eigenfunctions of the Laplace-Beltrami operator Δ:

$$\Delta f \stackrel{\text{def}}{=} -\operatorname{div} \nabla(f)$$

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

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

Since Δ is positive semidefinite, f that minimizes $\int_{\mathcal{M}} \|\nabla f\|^2$ has to be an eigenfunction of Δ. 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 that the embedding map f be orthogonal to f_0. It immediately follows that f_1 is the optimal embedding map. It is then easy to check that

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

provides the optimal m-dimensional embedding.

Remark. We note that the Laplacian eigenmaps discussed above generally *do not* provide an isometric embedding of the manifold though they have certain locality preserving properties on average. Devising an algorithmic procedure to find an isometric embedding for a potentially curved manifold from sampled data is an interesting problem for which no solution is known to the best of our knowledge. However, if the manifold is flat, Isomap (Tenenbaum, de Silva, & Langford, 2000) can be shown to converge to the optimal solution.

Of course a variety of classical embedding theorems are known in differential geometry. Again, we leave this for future investigation.

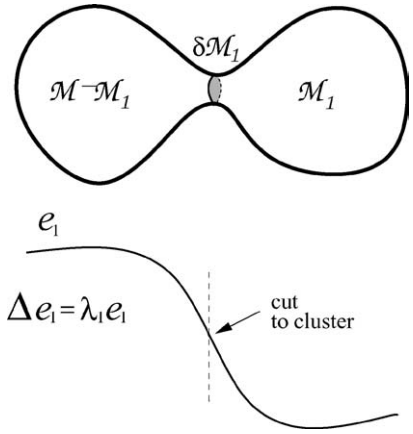

Figure 7. Clustering a manifold. Above: The manifold $\mathcal{M}$ is a three-dimensional "dumbbell". The boundary $\delta\mathcal{M}_1$ cuts the manifold in two parts optimally, so that the ratio of the area of the surface $\delta\mathcal{M}_1$ to the volume of the smallest of the two parts $\mathcal{M}_1$ and $\mathcal{M} - \mathcal{M}_1$ is minimized. Below: e_1 is the (hypothetical) second eigenfunction of Δ. Note that the eigenfunction is almost constant on the two big chunks and changes rapidly along the "neck" of the dumbbell. Cutting the manifold at the zero locus of f provides an approximation to the optimal clustering.

7.4. Clustering

Clustering is a special case of unsupervised learning where one wishes to partition the data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into a finite number of classes according to some criterion of optimality. Here we will only consider the case of two clusters, which already seems highly nontrivial. As it turns out, the Laplacian and its eigenfunctions that have played a central role in our development here, also provide us with a perspective on clustering.

We first consider what a sensible clustering would be, if we already knew the data manifold. An example of such clustering is given in figure 7. The manifold $\mathcal{M}$ (the so-called Calabi-Cheeger dumbbell) is partitioned in two parts $\mathcal{M}_1$ and $\mathcal{M} - \mathcal{M}_1$ by the membrane $\delta\mathcal{M}_1$. We use δ to denote the boundary. Intuitively, we want a surface with small area, which separates $\mathcal{M}$ in two large chunks. A formalization of this notion is the Cheeger constant (also known as the isoperimetric constant), which is essentially a measure of "goodness" for the best possible clustering:

$$h_{\mathcal{M}} = \inf_{\mathcal{B}=\delta\mathcal{M}_1} \frac{\text{vol}^{n-1}\mathcal{B}}{\min\left(\text{vol}^n(\mathcal{M}_1), \text{vol}^n(\mathcal{M} - \mathcal{M}_1)\right)}$$

While this quantity and the associated clustering are well-motivated and have a clear geometrical intuition, they are quite difficult to compute. To approximate them, one uses an idea of Cheeger (1970).[10] The (slightly reformulated) observation made by Cheeger was that if our manifold is nicely clustered, we can construct a function $\tilde{f}$, for which $\int_{\mathcal{M}} \|\nabla \tilde{f}\|^2$ is quite small and such that f is perpendicular to the constant function. To construct this

function is not hard. We put

$$\tilde{f}(x) = \begin{cases} \dfrac{1}{\mathrm{vol}(\mathcal{M}_1)} & x \in \mathcal{M}_1 \\ -\dfrac{1}{\mathrm{vol}(\mathcal{M} - \mathcal{M}_1)} & x \in \mathcal{M} - \mathcal{M}_1 \end{cases}$$

appropriately smoothing it at the boundary.

It is clear the $\int_{\mathcal{M}} \tilde{f} = 0$ and it can be shown (see Cheeger, 1970) for the details that $\frac{\int_{\mathcal{M}} \|\nabla \tilde{f}\|^2}{\int_{\mathcal{M}} \|f\|^2}$ is closely related to the Cheeger constant.

On the other hand, the second (first non-constant) eigenfunction of Δ is equal to

$$\operatorname*{argmin}_{f \perp \text{const}} \frac{\int_{\mathcal{M}} \|\nabla f\|^2}{\int_{\mathcal{M}} \|f\|^2}$$

Therefore, at least heuristically, the first nontrivial eigenfunction e_1 and the clustering function $\tilde{f}$ are close. We note that several upper and lower bounds for $h_{\mathcal{M}}$ in terms of the smallest nonzero eigenvalue of the Laplacian λ_1 are known, e.g., (Cheeger, 1970; Buser, 1982).

Thus an approximation to the optimal clustering is provided by $\mathcal{M}_1 = \{x | e_1(x) > 0\}$ and $\mathcal{M} - \mathcal{M}_1 = \{x | e_1(x) \leq 0\}$, cutting the manifold along the zero set of e_1. Thus the first nontrivial eigenfunction can be interpreted as a clustering of the manifold.

The situation with graphs is quite similar. A sensible way to cluster the graph, i.e. to cut the graph in two parts is again given by the Cheeger constant (e.g., see Chung, 1997), for a comprehensive treatment).

Let G be a connected weighted graph with the weight matrix $\mathbf{W}$. By a slight abuse of notation we will identify G with its set of vertices. Let G_1 be a subset of the vertices of G. We define

$$\mathrm{vol}(G_1) = \sum_{i \in G_1,\ j \in G} W_{ij}$$

We define the boundary δG_1 as the set of vertices connected to both G_1 and $G - G_1$. Then $\mathrm{vol}(\delta G_1)$ can be defined as the the amount of the "outward" flow from the boundary[11]

$$\mathrm{vol}(\delta G_1) = \sum_{i \in G_1,\ j \in G - G_1} W_{ij}$$

After these definitions the Cheeger constant is defined in the exactly same way as for the manifold:

$$h_G = \inf_{G_1} \frac{\mathrm{vol}(\delta G_1)}{\min\left(\mathrm{vol}(G_1), \mathrm{vol}(G - G_1)\right)}$$

Intuitively, we are trying to minimize the flow between G_1 and $G - G_1$.

Suppose now, G_1 and $G - G_1$ realize this optimal partition for which $\mathrm{vol}(\delta(G_1))$ is, presumably, small.

Consider the following function $\mathbf{f}$ on the graph:

$$f_i = \begin{cases} \dfrac{1}{\mathrm{vol}(G_1)} & i \in G_1 \\ -\dfrac{1}{\mathrm{vol}(G - G_1)} & i \in G - G_1 \end{cases}$$

It is clear that $\mathbf{f} \perp \mathbf{1}$, where $\mathbf{1} = (1, \ldots, 1)$. Recall that 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, $D_{ii} = \sum_j W_{ij}$.

It is not hard to verify that

$$\mathbf{f}^T \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{i,j} (x_i - x_j)^2 W_{ij} = \left(\frac{1}{\mathrm{vol}(G_1)} + \frac{1}{\mathrm{vol}(G - G_1)} \right)^{-2} \mathrm{vol}(\delta G_1)$$

On the other hand,

$$\mathbf{f}^T \mathbf{D} \mathbf{f} = \sum_i f_i^2 D_{ii} = \frac{1}{\mathrm{vol}(G_1)} + \frac{1}{\mathrm{vol}(G - G_1)}$$

Noticing that

$$\left(\frac{1}{\mathrm{vol}(G_1)} + \frac{1}{\mathrm{vol}(G - G_1)} \right)^{-1} < \frac{1}{2} \min(\mathrm{vol}(G_1), \mathrm{vol}(G - G_1))$$

we obtain:

$$\frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}} = \left(\frac{1}{\mathrm{vol}(G_1)} + \frac{1}{\mathrm{vol}(G - G_1)} \right)^{-1} \mathrm{vol}(\delta G_1) < 2h_G$$

The quantity $(\frac{1}{\mathrm{vol}(G_1)} + \frac{1}{\mathrm{vol}(G - G_1)})^{-1} \mathrm{vol}(\delta G_1)$ was introduced as the Normalized Cut in Shi and Malik (2000) in the context of image segmentation and provides a lower bound (and an approximation) for the Cheeger constant.

Similarly to the manifold case, while the direct computation of either the Cheeger constant or Normalized Cut is difficult (NP-hard), the relaxed problem

$$\tilde{\mathbf{f}} = \operatorname*{argmin}_{\mathbf{f} \perp \mathbf{1}} \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$$

is easily solved by finding the eigenvector of the normalized Laplacian (see the footnote at the beginning of Section 5.4) $\mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$ corresponding to the second smallest

(first nonzero) eigenvalue. That eigenvalue also provides a lower bound for the Cheeger constant.

As in the manifold case, the graph is then clustered by taking $G_1 = \{i \mid f_i > 0\}$, $G - G_1 = \{i \mid f_i \leq 0\}$. See Kannan, Vempala, and Vetta (2000) for some theoretical guarantees for the quality of spectral clustering.

From the data, one may construct a locally connected weighted graph following the procedure outlined in Section 4. The second eigenvector of the normalized Laplacian may then be used for clustering (bisecting) the data. It is notable, that the resulting algorithm is very similar to the algorithm proposed in Shi and Malik (2000) for image segmentation, despite the quite different setting and different motivation of the authors.

8. 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 provided only a partial glimpse of results relating to the convergence properties 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 $\mathbb{R}^k$.
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 and to Lehel Csató for helping with the figure 7.

Notes

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.

2. In fact when $\mathcal{M}$ is a circle, we do get the Fourier series.
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.
5. 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 and in in some ways closer to the Laplace-Beltrami operator on the manifold. It is taken as the definition of the graph Laplacian in Chung (1997). Since we did not see much improvement in the experimental results, we use $\mathbf{L}$ to simplify the exposition.
6. 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.
7. 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.
8. In the case of 2000 eigenvectors we take just 10 random splits since the computations are rather time-consuming.
9. 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.
10. Interestingly, Cheeger was interested in the opposite problem of estimating the analytical quantity λ_1 in terms of the geometric invariants of the manifold.
11. It is interesting to note that the volume of a hypersurface $\mathcal{B}$ on a manifold can be defined as

$$\mathrm{vol}^{n-1}(\mathcal{B}) = \lim_{t \to 0} \frac{\mathrm{vol}^n(G_t(\mathcal{B}))}{t}$$

where G_t is the geodesic flow in the direction normal to $\mathcal{B}$.

References

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation, 15:6*, 1373–1396.

Belkin, M., Matveeva, I., & Niyogi, P. (2003). Regression and regularization on large graphs. University of Chicago Computer Science, Technical Report TR-2003-11.

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the International Conference on Machine Learning.*

Bousquet, O., & Elisseeff, A. (2001). Stability and generalization. *Journal of Machine Learning Research*.

Buser, P. (1982). A note on the isoperimetric constant. *Ann. Sci. Ec. Norm. Sup. 15*.

Castelli, V., & Cover, T. M. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters, 16*.

Cheeger, J. (1970). A lower bound for the smallest eigenvalue of the laplacian. In R.C. Gunnings (Ed.), *Problems in analysis*. Princeton University Press.

Chapelle, O., Weston, J., & Scholkopf, B. (2003). Cluster kernels for semi-supervised learning. *Advances in Neural Information Processing Systems*.

Cucker, F., & Smale, S. (2001). On the mathematical foundations of learning. *Bulletin of the AMS, 39*, 1–49.

Chung, F. R. K. (1997). *Spectral graph theory*. Regional Conference Series in Mathematics, number 92.

Chung, F. R. K., Grigor'yan, A., & Yau, S.-T. (2000). Higher eigenvalues and isoperimetric inequalities on Riemannian manifolds and graphs. *Communications on Analysis and Geometry, 8*, 969–1026.

Haykin, S. (1999). *Neural networks, A comprehensive foundation.* Prentice Hall.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning.*

Kannan, R., Vempala, S., & Adrian Vetta. (2000). On clusterings: Good, bad and spectral. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science.*

Kondor, R., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the International Conference on Machine Learning.*

Kutin, S., & Niyogi, P. (2002). Almost everywhere algorithmic stability and generalization error. In *Proceedings of Uncertainty in Artificial Intelligence*.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled data. *Machine Learning, 39:2/3*.

Rosenberg, S. (1997). *The Laplacian on a riemmannian manifold*. Cambridge University Press.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*.

Schölkopf, B., Smola, A., & Mller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10:5*.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:8.*

Smola, A., & Kondor, R. (2003). Kernels and regularization on graphs. In *The Sixteenth Annual Conference on Learning Theory/The Seventh Workshop on Kernel Machines*.

Szummer, M., & Jaakkola, T. (2002). Partially labeled classification with Markov random walks. *Advances in Neural Information Processing Systems*.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*.

Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. W. H. Winston, Washington, D.C.

Wahba, G. (1990). Spline models for observational data. *Society for Industrial and Applied Mathematics.*

Zhou, D., Bousquet, O., Lal, T.N., Weston, J., & Schölkopf, B. (2003). Learning with local and global consistency, Max Planck Institute for Biological Cybernetics Technical Report.

Zhu, X., Lafferty, J., & Ghahramani, Z. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning.*

Received December 19, 2002
Revised January 29, 2004
Accepted January 30, 2004
Final manuscript January 30, 2004