Laplacian Eigenmaps for Dimensionality Reduction and Data Representation

Mikhail Belkin* Partha Niyogi†

December 24, 2001

Abstract

One of the central problems in machine learning and pattern recognition is to develop appropriate representations for complex data. We consider the problem of constructing a representation for data lying on a low dimensional manifold embedded in a high dimensional space. Drawing on the correspondence between the graph Laplacian, the Laplace Beltrami operator on the manifold, and the connections to the heat equation, we propose a geometrically motivated algorithm for representing the high dimensional data. The algorithm provides a computationally efficient approach to non-linear dimensionality reduction that has locality preserving properties and a natural connection to clustering. Some potential applications and illustrative examples are discussed.

1 Introduction

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

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

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

freedom of the camera. In this case, the space under consideration has the natural structure of a low dimensional manifold embedded in $\mathbb{R}^{n^2}$.

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

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

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

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

1. The core algorithm is very simple. It has a few local computations and one sparse eigenvalue problem. The solution reflects the intrinsic geometric structure of the manifold. It does, however, require a search for neighboring points in a high dimensional space.

2. The justification for the algorithm comes from the role of the Laplace Beltrami operator in providing an optimal embedding for the manifold. The manifold is approximated by the adjacency graph computed from the data points. The Laplace Beltrami operator is approximated by the weighted Laplacian of the adjacency graph with weights chosen

appropriately. The key role of the Laplace Beltrami operator in the heat equation enables us to use the heat kernel to choose the weights in a principled manner. Thus, the embedding maps for the data approximate the Eigenmaps of the Laplace Beltrami operator which are maps intrinsically defined on the entire manifold.

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

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

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

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

may serve as the basis for the emergence of categories in biological perception.

6. Since our approach is based on the intrinsic geometric structure of the manifold, it exhibits stability with respect to the embedding. In the example with the moving camera, different resolutions of the camera (i.e., different choices of n in the $n \times n$ image grid) should lead to embeddings of the same underlying manifold into spaces of very different dimension. Our algorithm will produce similar representations independently of the resolution.

1.1 The Problem of Dimensionality Reduction

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

In this paper, we consider the special case where $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathcal{M}$ and $\mathcal{M}$ is a manifold embedded in $\mathbb{R}^l$. We now consider an algorithm to construct representative $\mathbf{y}_i$'s for this special case. The sense in which such a representation is optimal will become clear later in this paper.

2 The Algorithm

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

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

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

   Advantages: geometrically motivated, the relationship is naturally transitive.

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

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

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

Disadvantages: less geometrically intuitive.

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

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

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

   The justification for this choice of weights will be provided later.

   (b) Simple-minded. [No parameters]. $W_{ij} = 1$ if and only if vertices i and j are connected by an edge.

   A simplification which avoids the necessity of choosing t.

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

   Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

   $$L\mathbf{f} = \lambda D\mathbf{f} \tag{1}$$

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

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

   $$L\mathbf{f}_0 = \lambda_0 D\mathbf{f}_0$$
   $$L\mathbf{f}_1 = \lambda_1 D\mathbf{f}_1$$
   $$\cdots$$

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

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

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

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

$$\mathbf{x}_i \to (\mathbf{f}_1(i), \dots, \mathbf{f}_m(i))$$

3 Justification

3.1 Optimal Embeddings

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

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

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

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

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

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

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

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

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

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

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

Therefore, the minimization problem reduces to finding

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

The constraint $\mathbf{y}^T D\mathbf{y} = 1$ removes an arbitrary scaling factor in the embedding. Matrix D provides a natural measure on the vertices of the graph. The bigger the value D_{ii} (corresponding to the ith vertex) is, the more "important" is that vertex. It follows from equation 2 that L is a positive semidefinite matrix and the vector $\mathbf{y}$ that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem

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

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

$$\underset{\substack{\mathbf{y}^T L\mathbf{y}=1 \\ \mathbf{y}^T D\mathbf{1}=0}}{\operatorname{argmin}} \mathbf{y}^T L\mathbf{y}$$

Thus, the solution is now given by the eigenvector with the smallest non-zero eigenvalue.

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

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

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

$$\underset{Y^T D Y = I}{\operatorname{argmin}} \operatorname{tr}(Y^T L Y)$$

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

3.2 The Laplace Beltrami Operator

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

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

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

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

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq \mathrm{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) \|\nabla f(x)\| + o(\mathrm{dist}_{\mathcal{M}}(\mathbf{x}, \mathbf{y})) \tag{3}$$

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

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

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

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

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

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

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

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

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

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

$$\underset{\|f\|_{L^2(\mathcal{M})}=1}{\text{argmin}} \int_{\mathcal{M}} \|\nabla f(x)\|^2 \tag{4}$$

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

It turns out that minimizing the objective function of eq. 4 reduces to finding eigenfunctions of the Laplace Beltrami operator $\mathcal{L}$. Recall that

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

where div is the divergence of the vector field. It follows from the Stokes' theorem that $-\text{div}$ and ∇ are formally adjoint operators, i.e. if f is a function and $\mathbf{X}$ is a vector field then[2] $\int_{\mathcal{M}} \langle \mathbf{X}, \nabla f \rangle = -\int_{\mathcal{M}} \operatorname{div}(\mathbf{X}) f$. Thus

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

We see that $\mathcal{L}$ is positive semidefinite. f that minimizes $\int_{\mathcal{M}} \|\nabla f\|^2$ has to be an eigenfunction of $\mathcal{L}$. The spectrum of $\mathcal{L}$ on a compact manifold $\mathcal{M}$ is known to be discrete. Let the eigenvalues (in increasing order) be $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots$ and let f_i be the eigenfunction corresponding to eigenvalue λ_i. It is easily seen that f_0 is the constant function that maps the entire manifold to a single point. To avoid this eventuality, we require (just as in the graph setting) that the embedding map f be orthogonal to f_0. It immediately follows that f_1 is the optimal embedding map. Following the arguments of the previous section, we see that

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

provides the optimal m-dimensional embedding.

[2] If $\mathcal{M}$ has a boundary, appropriate boundary conditions for f need to be assumed.

3.3 Heat Kernels and the Choice of Weight Matrix

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

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

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

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

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

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

See Rosenberg (1997) for more details.

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

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

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

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

The coefficient $\frac{1}{t}$ is global and will not affect the eigenvectors of the discrete Laplacian. Since the inherent dimensionality of $\mathcal{M}$ may be unknown, we put

$\alpha = \frac{1}{k}(4\pi t)^{-\frac{m}{2}}$. It is interesting to note that since the Laplacian of the constant function is zero, it immediately follows that $\frac{1}{\alpha} = \sum\limits_{\substack{\mathbf{x}_j \\ 0<\|\mathbf{x}_j-\mathbf{x}_i\|<\epsilon}} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}}$ and

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

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

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

4 Connections to Spectral Clustering

The approach to dimensionality reduction considered in this paper utilizes maps provided by the eigenvectors of the graph Laplacian and eigenfunctions of Laplace Beltrami operator on the manifold. Interestingly, this solution may also be interpreted in the framework of clustering and has very close ties to spectrally based clustering techniques such as those used for image segmentation (Shi and Malik, 1997), load balancing (Hendrickson, Leland, 1993), and circuit design (Hadley et al, 1992).

Here we briefly outline the ideas of spectral clustering It is often of interest to cluster a set of n objects into a finite number of clusters. Thus, given a set of n objects (visual, perceptual, linguistic or otherwise), one may introduce a matrix of pair wise similarities between the n objects. It is then possible to formulate a general graph-theoretic framework for clustering as follows. Let $G = (V, E)$ be a weighted graph, W is the matrix of weights, where the vertices are numbered arbitrarily. The weight W_{ij} associated with the edge e_{ij} is the similarity between v_i and v_j. We assume that the matrix of pairwise similarities is symmetric and the corresponding undirected graph is connected.[3]

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

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

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

Trying to minimize the $\mathrm{cut}(A, B)$ will favor cutting off weakly connected outliers which tends to lead to poor partitioning quality. To avoid that problem a measure on the set of vertices is introduced. The weight of a vertex is its "importance" relative to other vertices.

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

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

Shi and Malik (1997), define the normalized cut

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

The problem, as formulated by Shi and Malik (1997), is to minimize Ncut over all partitions of the vertex set V. [4]

It turns out the the combinatorial optimization problem as stated is NP-hard (a proof due to Papadimitrou can be found in Shi and Malik (1997)). However, if we allow relaxation of the indicator functions to real values, the problem reduces to minimizing the Laplacian of the graph, which can be easily computed in polynomial time with arbitrary precision.

Recall that

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

[4] A similar and, perhaps, more geometrically motivated quantity is the Cheeger constant

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

where $\bar{A}$ is the complement of A in V. See Chung (1997) for further reference.

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

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

We have

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

Also

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

Thus

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

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

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

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

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

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

Remark: The central observation to be made here is that the process of dimensionality reduction that preserves locality yields the same solution as clustering. It is worthwhile to compare the global algorithm presented in Tenenbaum et al (2000)with the local algorithms suggested here and in Roweis and Saul (2000). One approach to non-linear dimensionality reduction as exemplified by Tenenbaum et al attempts to faithfully approximate all geodesic distances on the manifold. This may be viewed as a global strategy. In contrast, the local approach presented here (as well as that presented in Roweis and Saul, 2000) attempts only to approximate or preserve neighborhood information. This, as we see from the preceding discussion may also be interpreted as imposing a soft clustering of the data (which may be converted to a hard clustering by a variety of heuristic techniques). In this sense the local approach to dimensionality reduction imposes a natural clustering of the data.

5 Analysis of Locally Linear Embedding Algorithm

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

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

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

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

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

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

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

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

   Notice that E is a symmetric positive semidefinite matrix.

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

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

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

Step 1:
Let us fix a data point $\mathbf{x}_i$. We now show that

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

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

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

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

where

$$\sum \alpha_j = 1$$

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

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

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

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

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

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

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

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

Step 2:
Now note that if $\sqrt{\alpha_i}\ \mathbf{v_i}$ form an orthonormal basis (which, of course, is not usually the case) then

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

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

Indeed if $\mathbf{e}_1, \ldots, \mathbf{e}_n$ form an orthonormal basis for H corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$, then using the Spectral theorem,

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

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

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

Step 3:
Putting steps 1 and 2 together, we see that

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

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

6 Examples

We now briefly consider several possible applications of the algorithmic framework developed in this paper. We begin with a simple synthetic example of a "swiss roll" considered in Tenenbaum et al (2000) and Roweis and Saul (2000). We then consider a toy example from vision with vertical and horizontal bars in a "visual field". We conclude with some low dimensional representations constructed from naturally occurring data sets in the domains of speech and language respectively. In all our experiments, we use the simplest version of the algorithm ($W_{ij} \in \{0, 1\}$) which turns out to work well in practice.

6.1 A Synthetic "Swiss roll"

In fig. 1, we show an example where data lies on a swiss roll. Note that the swiss roll is really a flat two dimensional submanifold of $\mathbb{R}^3$. Unlike, ISOMAP, our algorithm does not attempt to isometrically embed the swiss roll into $\mathbb{R}^2$. However, it manages to unroll the swiss roll thereby preserving the geometrical characteristics of the surface. It is worthwhile to point out that an isometric embedding preserving global distances such as that attempted by ISOMAP is theoretically possible only when the surface is flat, i.e., the curvature tensor is zero. For example, a well known classical result due to Gauss shows that there is no distance preserving map from any part of the 2-sphere into the plane.

6.2 A Toy Vision Example

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

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

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

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

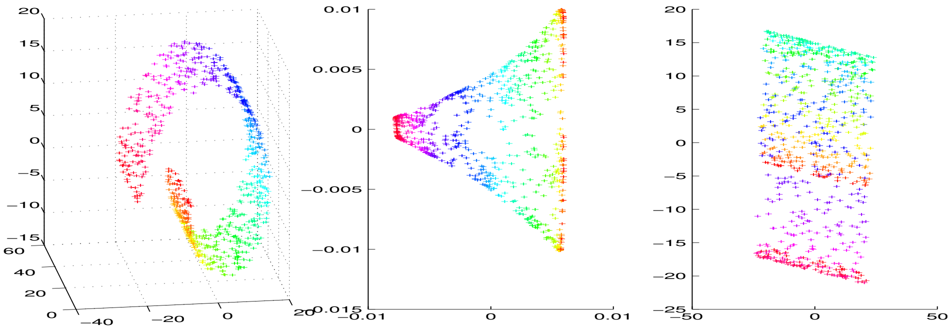


Figure 1: "Swiss roll", 2-dimensional Laplacian representation, and PCA representation, left to right. For the purposes of illustration we compare a spectral 2-dimensional representation of the "swiss roll" to principal component analysis. PCA is limited to projections and therefore cannot produce a good representation of non-linear data.

where appropriate boundary conditions apply. The space of all images of vertical bars is a two dimensional manifold. Similarly, the space of all horizontal bars is a two dimensional manifold as well. Each of these manifolds is embedded in the space of functions ($L^2([0,1] \times [0,1])$). Notice that while these manifolds do not intersect, they come close to each other.

To discretize the problem, we consider a 40×40 grid for each image. Thus each image may be represented as a 1600 dimensional binary vector. We choose 1000 images (500 containing vertical bars and 500 containing horizontal bars) at random.

In fig. 2 the left panel shows a horizontal and vertical bar to give the reader a sense of the scale of the image. The middle panel is a two dimensional representation of the set of all images using the Laplacian eigenmaps. Notice that while the local graph is connected, the two dimensional representation shows two well defined components. The right panel shows the result of a principal components analysis using the first two principal directions to represent the data.

6.3 A Linguistic Example

An experiment was conducted with the 300 most frequent words in the Brown corpus – a collection of texts containing about a million words (not distinct) available in electronic format. Each word is represented as a vector

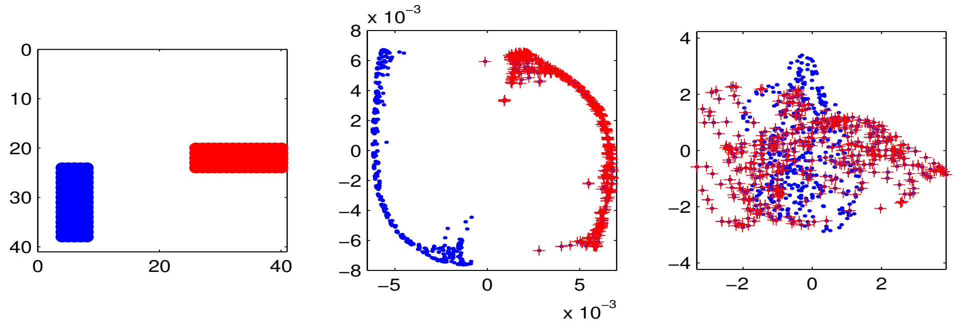


Figure 2: The left panel shows a horizontal and a vertical bar. The middle panel is a two dimensional representation of the set of all images using the Laplacian eigenmaps. The right panel shows the result of a principal components analysis using the first two principal directions to represent the data. Blue dots correspond to images of vertical bars and red '+' signs correspond to images of horizontal bars.

in a 600 dimensional space using information about the frequency of its left and right neighbors (computed from the corpus). More precisely, let the 300 words be w_1 through w_{300}. Then the representation of w_i is a 600 dimensional vector $\mathbf{v}_i$ (say) where the first three hundred dimensions of $\mathbf{v}_i$ characterize left neighbor relations and the next three hundred characterize right neighbor relations. Thus $\mathbf{v}_i(j)$ – the jth component ($j \leq 300$) of v_i is the number of times the sequence $w_j w_i$ occurs in the corpus (referred to as the bigram count). Similarly, $\mathbf{v}_i(j + 300)$ is the the count of the number of times the sequence $w_i w_j$ occurs in the corpus.

Thus there are 300 vectors in $\mathbb{R}^{600}$. Of course, we do not claim that there is a natural low dimensional manifold structure on these vectors. Nevertheless, it is useful for practical applications to construct low dimensional representations of this data. For example, the well known LSI (Latent Semantic Indexing) approach uses Principal Components Analysis to represent the documents in a vector space model for purposes of search and information retrieval. Applying the Laplacian eigenmap to the data yields a low dimensional representation shown in figs. reffig:300words and 4 respectively. Note that words belonging to similar syntactic categories seem to cluster together highlighting further the connections between clustering and dimensionality reduction as discussed in this paper.

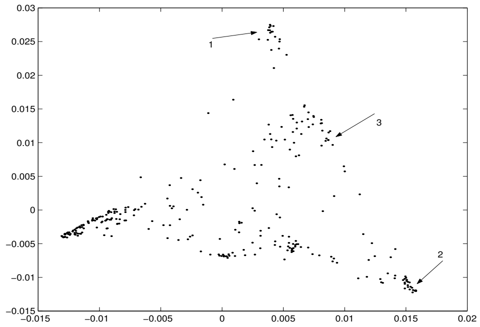
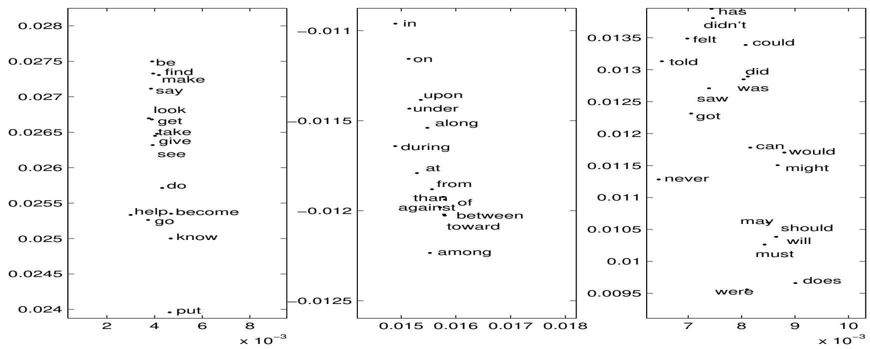


Figure 3: 300 most frequent words of the Brown corpus represented in the spectral domain.

Figure 4: Fragments labelled by arrows, form left to right. The first is exclusively infinitives of verbs, the second contains prepositions and the third mostly modal and auxiliary verbs. We see that syntactic structure is well-preserved.



6.4 Speech

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

In this example, we consider the low dimensional representations that

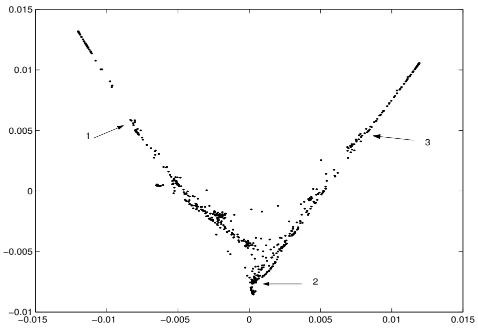


Figure 5: 685 speech data points plotted in the two dimensional Laplacian spectral representation.

arise by applying the Laplacian eigenmap algorithm to a sentence of speech sampled at 16kHz. A short time Fourier transform (with a 30 millisecond window) was computed from the speech signal at 5 millisecond intervals. This yielded a vector of Fourier coefficients[5] for every 30 ms chunk of the speech signal. There were 685 such vectors in all. Each vector was a set of 256 Fourier coefficients. Furthermore, each vector was labeled according to the identity of the phonetic segment it belonged to. These labels are not utilized by the Laplacian Eigenmap algorithm which finds a low dimensional representation for the data. Shown in fig. 5, are the speech data points plotted in the two dimensional Laplacian representation. The two "spokes" correspond predominantly to fricatives and closures respectively. The central portion corresponds mostly to periodic sounds like vowels, nasals, and semivowels. A natural clustering into the broad classes is obtained and fig. reffig:phonemesome shows three different regions of the representation space. Note the phonetic homogeneity of the data points that lie in each of these regions. Points mapped to the same region in the representation space share similar phonetic features though points with the same label may originate from different occurrences of the same phoneme.

7 Conclusions

In this paper we introduced a coherent framework for dimensionality reduction for the case where data resides on a low dimensional manifold embedded

[5]In practice, the log of the Fourier coefficients was taken for technical reasons that are beyond the scope of the current paper.

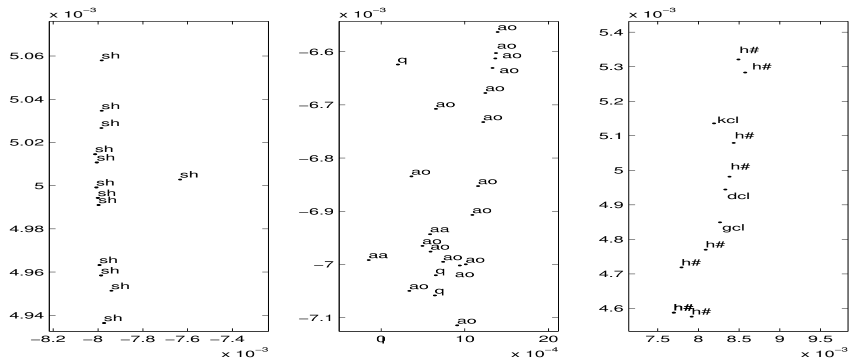


Figure 6: A blowup of the three selected regions (1,2,3) left to right. Notice the phonetic homogeneity of the chosen regions. The data points corresponding to the same region have similar phonetic identity though they may (and do) arise from occurrences of the same phoneme at different points in the utterance. The symbol "sh" stands for the fricative in the word she; "aa","ao" stand for vowels in the words dark and all respectively; "kcl","dcl","gcl" stand for closures preceding the stop consonants "k","d","g" respectively. "h#" stands for silence.

in a higher dimensional space. A number of questions remain to be answered.

1. Our approach utilizes the properties of Laplace Beltrami operator to construct invariant embedding maps for the manifold. While such maps have some demonstrable locality preserving properties they do not in general provide an isometric embedding. Although Nash's embedding theorem guarantees the existence of an isometric embedding in a low dimensional space, it remains unclear whether such an embedding is easily computable by a discrete algorithm. Furthermore, there are usually many possible isometric embeddings of a given manifold. For example, any knot in $\mathbb{R}^3$ is an isometric embedding of a circle. It is not clear how to discriminate between "good" and "bad" isometric embeddings. It would therefore be interesting to formulate more precisely what properties of an embedding make it desirable for pattern recognition and data representation problems.

2. We have not given any consideration to other geometric invariants of the manifold that may be potentially estimated from data. For

example, it is unclear how to reliably estimate even such a simple invariant as the intrinsic dimensionality of the manifold.

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

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

Acknowledgments:

We are very grateful to John Goldsmith for motivating us to consider the approach discussed here, to Peter Bickel for many insightful critical comments, and to Yali Amit, Lazslo Babai, Todd Dupont, Joshua Maher and Ridgway Scott for conversations.

References

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

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

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

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

[5] Bruce Hendrickson and Robert Leland, Multidimensional Spectral Load Balancing, short version in Proc. 6th SIAM Conf. Parallel Proc., 953-961, 1993.

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

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

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

[9] Jianbo Shi, Jitendra Malik, Normalized Cuts and Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 22, no 8, August 2000

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