Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. TR-110

A kernel view of the dimensionality reduction of manifolds

Jihun Ham,[1] Daniel D. Lee,[1] Sebastian Mika,[2]
Bernhard Schölkopf[3]

July 2003

[1] University of Pennsylvania, Philadelphia, PA, USA, email: jhham,ddlee@seas.upenn.edu
[2] Fraunhofer FIRST.IDA, Berlin, Germany, email: mika@first.fraunhofer.de
[3] Max Planck Institute for Biological Cybernetics, Tübingen, Germany, email: bs@tuebingen.mpg.de

A kernel view of the dimensionality reduction of manifolds

Jihun Ham†, Daniel D. Lee†, Sebastian Mika*, Bernhard Schölkopf‡
† University of Pennsylvania, Philadelphia, PA
* Fraunhofer FIRST.IDA, Berlin, Germany
‡ Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Abstract

We interpret several well-known algorithms for dimensionality reduction of manifolds as kernel methods. Isomap, graph Laplacian eigenmap, and locally linear embedding (LLE) all utilize local neighborhood information to construct a global embedding of the manifold. We show how all three algorithms can be described as kernel PCA on specially constructed Gram matrices, and illustrate the similarities and differences between the algorithms with representative examples.

1 Introduction

Recently, several different algorithms have been developed to perform dimensionality reduction of low-dimensional nonlinear manifolds embedded in a high dimensional space. Isomap [11] was originally proposed as a generalization of multidimensional scaling (MDS) [4]. An alternative method known as locally linear embedding (LLE) [8] was developed that solved a consecutive pair of linear least square optimizations. More recently, another method for dimensionality reduction of manifolds has been described in terms of the spectral decomposition of graph Laplacians [2]. Although all three algorithms, Isomap, graph Laplacian eigenmaps, and LLE have quite different motivations and derivations, they all can perform dimensionality reduction on nonlinear manifolds as shown in Fig. 1.

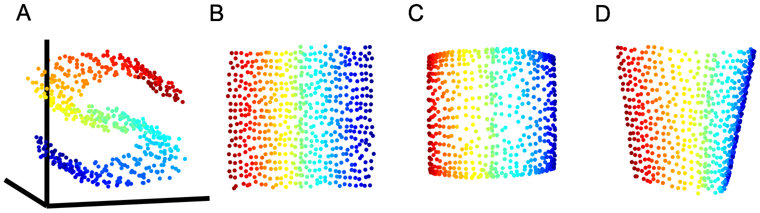

Figure 1: The two-dimensional embeddings resulting from Isomap (B), Laplacian eigenmap (C), and LLE (D) from 600 points sampled from the S-curve manifold (A). $K = 6$ nearest neighborhoods were used for computing the embeddings.

All three algorithms share a common characteristic in that they first induce a local neighborhood structure on the data, and then use this local structure to globally map the manifold to a lower dimensional space. This local neighborhood relationship is typically defined using nearest neighbors in Euclidean space and can be described by a graph $\mathcal{G}(V, E)$, where the nodes V represent different data points, and the edges E represent neighborhood relations among the points. However, the way these different algorithms use this neighborhood structure to find a global embedding is quite different. In this work, we interpret the different algorithms as kernel methods. Specifically, we will relate them to the kernel PCA (KPCA) algorithm [10].

Regarded in this context, Isomap, graph Laplacians, and LLE all share a similar strategy. They construct a kernel matrix over the finite domain of the training data that preserves some aspect of the manifold structure from the input space to a feature space. Diagonalization of this kernel matrix then gives rise to an embedding that captures the low-dimensional structure of the manifold.

In the following we will first fix our notation and provide a short review of kernel PCA (Sec. 2). We then in turn show how Isomap (Sec. 3), graph Laplacian eigenmaps (Sec. 4) and LLE (Sec. 5) can be interpreted in the context of KPCA. We conclude with a discussion of the similarities and differences between the methods.

2 Review of Kernel PCA

Suppose we are given a nonempty set $\mathcal{X}$ and a *positive definite kernel* k. By the latter, we mean a real-valued function on $\mathcal{X} \times \mathcal{X}$ with the property that there exists a map $\Phi : \mathcal{X} \to \mathcal{H}$ into a dot product space $\mathcal{H}$ such that for all $x, x' \in \mathcal{X}$, we have $\langle x, x' \rangle = k(x, x')$.[1] In kernel methods, k can be viewed as a nonlinear similarity measure.

Given data $x_1, \ldots, x_m \in \mathcal{X}$ which we assume to be in a vector space, kernel PCA computes the principal components of the points $\Phi(x_1), \ldots, \Phi(x_m)$. Since $\mathcal{H}$ may be infinite-dimensional, the PCA problem needs to be transformed into a problem that can be solved in terms of the kernel k. To this end, we consider the covariance matrix in $\mathcal{H}$,

$$\mathbf{C} := \frac{1}{m} \sum_{i=1}^{m} \Phi(x_i) \Phi(x_i)^T, \tag{1}$$

where $\Phi(x_i)^T$ denotes the linear form mapping $\mathbf{v}$ to $\langle \Phi(x_i), \mathbf{v} \rangle$. To diagonalize $\mathbf{C}$ even if $\mathcal{H}$ is infinite-dimensional, we first observe that all solutions to

$$\mathbf{C}\mathbf{v} = \lambda \mathbf{v} \tag{2}$$

with $\lambda \neq 0$ must lie in the span of Φ-images of the training data (as can be seen by substituting (1) and dividing by λ). Thus, we may expand the solution $\mathbf{v}$ as

$$\mathbf{v} = \sum_{i=1}^{m} \alpha_i \Phi(x_i), \tag{3}$$

thereby reducing the problem to that of finding the α_i. The latter can be shown to take the form

$$m\lambda \boldsymbol{\alpha} = K \boldsymbol{\alpha}, \tag{4}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^T$ and $K_{ij} = k(x_i, x_j)$. Absorbing the m factor into the eigenvalue λ, one can moreover show that the p-th feature extractor takes the form

$$\langle \mathbf{v}^p, \Phi(x) \rangle = \frac{1}{\sqrt{\lambda^p}} \sum_{i=1}^{m} \alpha_i^p k(x_i, x). \tag{5}$$

[1] Note that this is sometimes called a *positive semidefinite* kernel. In the kernel literature, *positive definite* is more common, with the term *strictly positive definite* being used for the case where the associated kernel matrix is full rank. We use the same terminology for matrices.

This is derived by computing the dot product between a test point $\Phi(x)$ and the p-th eigenvector in feature space; the $\frac{1}{\sqrt{\lambda^p}}$ factor ensures that $\langle \mathbf{v}^p, \mathbf{v}^p \rangle = 1$.

Below we will make use of the following observation: The p-th feature values extracted by KPCA on the training example x_n is proportional to the expansion coefficients α_n^p. This can be seen as follows: Substituting $x = x_n$ in (5), we get

$$\langle \mathbf{v}^p, \Phi(x_n) \rangle = \frac{1}{\sqrt{\lambda^p}} (K\alpha^p)_n = \frac{1}{\sqrt{\lambda^p}} (\lambda^p \alpha^p)_n = \sqrt{\lambda^p} \alpha_n^p. \tag{6}$$

Finally, we should mention one modification. In (1), we have implicitly assumed that the data in the feature space have zero mean. In general, we cannot assume this, and therefore we need to subtract the mean $(1/m) \sum_i \Phi(x_i)$ from all points. This leads to a slightly different eigenvalue problem, where we diagonalize

$$K' = (I - ee^T) K (I - ee^T) \tag{7}$$

(with $e = m^{-1/2}(1, \ldots, 1)^T$) rather than K.

3 Isomap

As in multidimensional scaling (MDS), Isomap first constructs a matrix of pairwise distances between the different data points [11]. However, instead of directly using Euclidean distance in the high-dimensional space, Isomap constructs a symmetric adjacency graph using criteria such as symmetric nearest neighborhoods or ϵ-ball neighborhoods. It then weights each of the edges in this graph by the Euclidean distance between neighboring points (a variant called C-Isomap also normalizes these weights [5]). Now Dijkstra's algorithm is used to compute the shortest path among edges in the neighborhood graph to define the total distance between pairs of points. Finally, MDS is applied to this shortest path distance matrix and the embedding is given by the coefficients of the smallest eigenvectors of this matrix. As pointed out in [12], one can interpret metric multidimensional scaling as kernel PCA (with the main difference being that kernel PCA also provides an embedding for test points, whereas MDS only embeds the training points). In a similar fashion, one can take the distances used in Isomap and consider the following "kernel":

$$K_{\text{Isomap}} = -\frac{1}{2}(I - ee^T) S (I - ee^T), \tag{8}$$

where S is the squared distance matrix, and $e = m^{-1/2}(1, \ldots, 1)^T$ is the uniform vector of unit length. This will center K_{Isomap}; but there is no theoretical guarantee that it will be positive definite. However, in the continuum limit for a smooth manifold, the geodesic distance between points on the manifold will be proportional to Euclidean distance in the low-dimensional parameter space of the manifold [6]. It is known that $k(x, x') = -\|x - x'\|^\beta$ is conditionally positive definite for $0 < \beta \leq 2$. In the continuum limit, $(-S)$ will thus be conditionally positive definite and K_{Isomap} will be positive definite (see pp. 49 and 51 in [9]; see also p. 440 for an example of kernel PCA using $k(x, x') = -\|x - x'\|^\beta$, i.e., with $S_{ij} = \|x_i - x_j\|^\beta$).

Now recall (6); since the final embedding found by Isomap is given by the *largest* eigenvectors of (8) we see that using the projections given by the largest eigenvectors of KPCA using K_{Isomap} yields, up to scaling by $\sqrt{\lambda^p}$, an identical solution. Shown in Fig. 2 are the results of Isomap applied to the S-curve manifold, showing the resulting spectrum of K_{Isomap} and plots of the associated metric S.

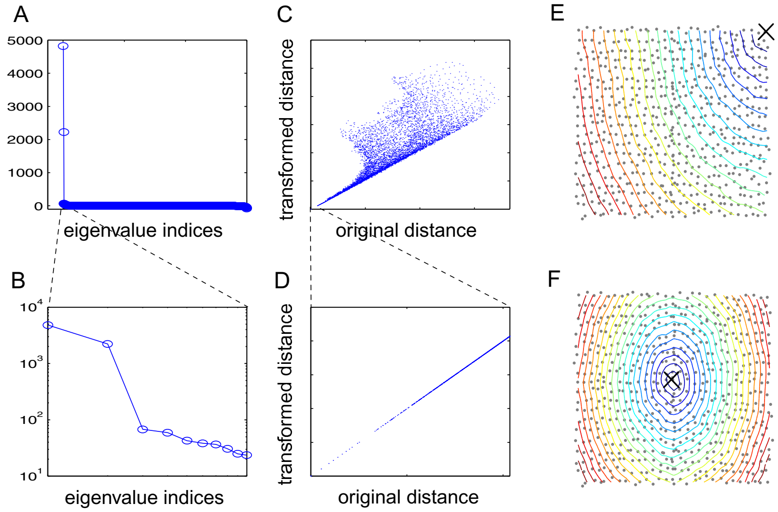

Figure 2: Left column shows the spectrum of K_{Isomap} on a linear (A) and on a log-scale (B). Middle column compares distances between data vectors in feature space versus their distances in the original input space on a global (C) and local (up to mean radius of neighbors)(D) scale. Contour plots of distance from a point (marked by x) on the boundary (E) and in the center (F) of the manifold with the induced metric.

4 Graph Laplacian

The graph Laplacian eigenmap algorithm [2] also incorporates directed or undirected graph structure describing the local neighborhood relations between data points. As in Isomap, these neighbor relations can be defined in terms of symmetric nearest neighbors or a small distance criterion. The neighborhood relations are summarized by the adjacency matrix W where $W_{ij} > 0$ if the ith and jth data points are neighbors ($i \sim j$), assumed to be symmetric, otherwise $W_{ij} = 0$. The non-zero weights in W can be chosen from $\{0, 1\}$, or according to $W_{ij} = e^{-|x_i - x_j|^2/2\sigma^2}$ (a Gaussian kernel) where σ is an adjustable parameter. The generalized graph Laplacian L is defined in terms of the adjacency matrix W as:

$$L_{ij} := \begin{cases} d_i, & \text{if } i = j, \\ -W_{ij}, & \text{if } i \sim j, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $d_i = \sum_{j \sim i} W_{ij}$ is the degree of the ith vertex. The normalized graph Laplacian $\mathcal{L}$ is a symmetric matrix related to L by $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ with the diagonal matrix $D_{ij} = \delta_{ij} d_i$. We assume that the graph is connected, so that L will have a single zero eigenvalue associated with the uniform vector e.

Belkin and Niyogi [2] motivate the role of the graph Laplacian for dimensionality reduction by showing that a plausible cost for a one-dimensional embedding of the nodes of the graph $\psi : V \mapsto \mathcal{R}$ is given by:

$$\boldsymbol{\psi}^{\boldsymbol{T}} L \boldsymbol{\psi} = \frac{1}{2} \sum_{i,j} (\psi_i - \psi_j)^2 W_{ij} \tag{10}$$

which also shows that L is positive definite. Minimizing the quadratic form (10) involves finding the eigenvectors with the smallest eigenvalues of either the graph Laplacian L or $\mathcal{L}$, depending upon the constraints used in the optimization.

4.1 Diffusion kernel

The graph Laplacian is closely related to a description of diffusion on the graph [7]. As a continuous time dynamical system, the evolution of a diffusing field on the graph is given by the differential equation:

$$\frac{\partial\psi(t)}{\partial t} = -L\psi(t) \tag{11}$$

The solution to this equation is related to the matrix exponential of L, otherwise known as the Green's function or heat kernel:

$$K_t = \exp(-Lt) = \sum_p \phi_p\phi_p^T e^{-\lambda_p t} \tag{12}$$

where ϕ_p and λ_p are the eigenvectors and eigenvalues of L. In terms of the heat kernel, the generic solution to (11) is given by $\psi(t) = K_t\psi(0)$. Thus, the eigenvectors of L with the smallest eigenvalues correspond to the most slowly decaying modes under diffusion, and the uniform vector corresponding to zero eigenvalue is the stationary distribution. The heat kernel can be related to the covariance of the time evolved field [7]:

$$\begin{aligned}\langle\psi(t)\psi(t)^T\rangle &= K_t\langle\psi(0)\psi(0)^T\rangle K_t & (13)\\ \langle\delta\psi(t)\delta\psi(t)^T\rangle &= K_t\langle\psi(0)\psi(0)^T\rangle K_t - ee^T & (14)\end{aligned}$$

Assuming the statistics of the initial condition are independent, $\langle\psi(0)\psi(0)^T\rangle = I$, we can integrate the covariance over all time to obtain the positive definite matrix:

$$K_L = \int_0^\infty \langle\delta\psi(t)\delta\psi(t)^T\rangle\, dt = \int_0^\infty \left[\sum_p \phi_p\phi_p^T e^{-2\lambda_p t} - ee^T\right] dt = \frac{1}{2}L^\dagger \tag{15}$$

where $L^\dagger$ is the pseudo-inverse of the graph Laplacian, known as the discrete Green's function [3].

We can relate K_L to a metric distance by considering L_{ij} as the transition rate from state j to state i in a continuous-time Markov chain. The properties of this Markov chain are given by the fundamental matrix [1]:

$$Z = \int_0^\infty \left[\exp(-Lt) - ee^T\right] dt = L^\dagger, \tag{16}$$

where $\exp(-Lt)_{ij}$ gives the probability of being in state i starting from state j after time t. From this fundamental matrix we can derive the commute time C_{ij} which is the expected time for the random process to travel from node j to reach node i and then return:

$$C_{ij} \propto L^\dagger_{ii} + L^\dagger_{jj} - L^\dagger_{ij} - L^\dagger_{ji} \tag{17}$$

The commute time satisfies the triangle inequality, so it can be viewed as a proper metric on the graph. Thus, from (17) we see that $K_L = L^\dagger$ can be regarded as a kernel associated with a distance that is proportional to the commute time. Thus, the embedding constructed by taking the smallest eigenvectors of the graph Laplacian is equivalent to performing kernel PCA on the matrix K_L associated with the commute times of diffusion on the graph. As in our analysis of the Isomap algorithm, kernel PCA on K_L is also equivalent to multidimensional scaling of the graph commute times. The spectrum of K_L and plots of the induced commute time metric for the S-curve manifold are shown in Fig. 3.

5 LLE

The LLE algorithm [8] first constructs a weight matrix W whose ith row contains the linear coefficients that sum to unity and optimally reconstruct x_i from its p nearest neighbors. Defining

$$M := (I - W)(I - W^T)$$

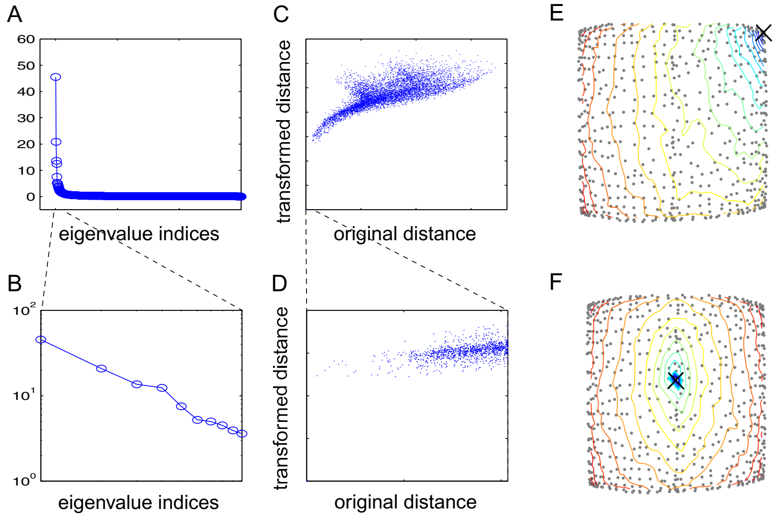

Figure 3: Left column shows the spectrum of K_L on a linear (A) and on a log-scale (B). Middle column compares distances between data vectors in feature space versus their distances in the original input space on a global (C) and local (up to mean radius of neighbors)(D) scale. Contour plots of distance from a point (marked by x) on the boundary (E) and in the center (F) of the manifold with the induced metric.

, which has a maximum eigenvalue λ_{max}, one can show that M's smallest eigenvalue is 0 and the corresponding eigenvector is the uniform vector e. Since the other eigenvectors are orthogonal to e, their coefficients sum to 0. In LLE, the coordinate values of the m-dimensional eigenvectors $m-d, \ldots, m-1$ give an embedding of the m data points in $\mathbb{R}^d$. If we define

$$K := (\lambda_{max} I - M), \tag{18}$$

then by construction, K is a positive definite matrix, its leading eigenvector is e, and the coordinates of the eigenvectors $2, \ldots, d+1$ provide the LLE embedding. This straightforward connection was pointed out in [9, Exercise 14.17]. However, the link between kernel PCA and LLE goes further than that. Equivalently, we can project out e, and then use the eigenvectors $1, \ldots, d$ of the resulting matrix as

$$(I - ee^T)K(I - ee^T). \tag{19}$$

Note that this is identical to the centered kernel matrix (7) which is used in kernel PCA.

So far we thus know that the coordinates of the leading eigenvectors of kernel PCA performed on K yield the LLE embedding. This, together with the considerations summarized in (6), shows that the LLE embedding is equivalent to the KPCA projections up to a multiplication with $\sqrt{\lambda^p}$. This corresponds to the whitening step which is performed in LLE in order to fix the scaling, but not normally in kernel PCA, where the scaling is determined by the variance of the data along the respective directions in $\mathcal{H}$.

Note that there need (and probably will) not be an analytic form of a kernel k which gives rise to the LLE kernel matrix K. Accordingly, there need not be a feature map Φ corresponding to it which is defined on the whole input domain. Nevertheless, one can at least give a feature map defined on the training points. To this end, write $K = SDS^T$, with an orthogonal matrix S (with rows S_i) and a diagonal matrix D with nonnegative entries.

Then the gram matrix is given by

$$k(x_i, x_j) = (SDS^T)_{ij} = \langle S_i, DS_j \rangle = \left\langle \sqrt{D}S_i, \sqrt{D}S_j \right\rangle . \tag{20}$$

5.1 Graph operator interpretation

The symmetric, positive definite matrix M in LLE can also be regarded as an operator acting on fields defined over a graph. In that regard, it acts similar to the square of the graph Laplacian [2]. However, LLE differs from other spectral graph techniques in its construction of M by explicitly minimizing $\sum_{ij} M_{ij}(x_i \cdot x_j)$ where the dot product of the data is in the original input space. If we define a continuous time dynamics for fields over the graph using the operator M:

$$\frac{\partial \psi(t)}{\partial t} = -M\psi(t) \tag{21}$$

we see that the choice of M is equivalent to minimizing $\psi^T \frac{\partial \psi}{\partial t}$ when the field ψ is initialized with the coordinates of the original data points. In analogy with the graph Laplacian embedding as the slowest decaying eigenmodes of the diffusion operator, the LLE embedding is given by the slowest decaying eigenmodes of (21).

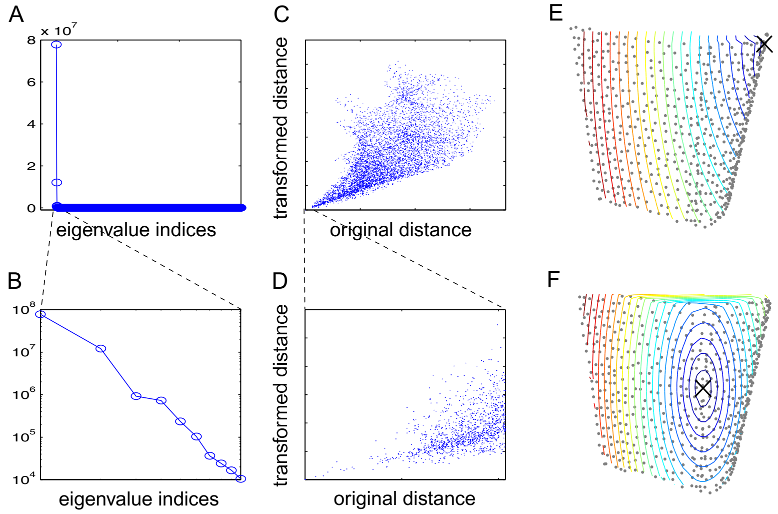

Figure 4: Left column shows the spectrum of the pseudo-inverse $K_\dagger$ on a linear (A) and on a log-scale (B). Middle column compares distances between data vectors in feature space versus their distances in the original input space on a global (C) and local (D) scale. Contour plots of distance from a point on the boundary (E) and in the center (F) of the manifold with the induced metric.

We can construct an alternative kernel for LLE that is analogous to the heat kernel for the graph Laplacian by considering the Green's function of M, $K_t = \exp(-Mt)$. Similar to the diffusion kernels, this kernel is related to the covariance of the time evolved field under (21). Integrating this covariance over time yields the pseudo-inverse kernel $K_\dagger = M^\dagger$ which is positive definite and centered. As noted before, performing kernel PCA on $K_\dagger$ is then equivalent to LLE up to scaling factors. The properties of $K_\dagger$ when LLE is applied to the S-curve data is shown in Fig. 4.

6 Discussion

We have seen that all three algorithms, Isomap, graph Laplacian eigenmaps, and LLE can be interpreted as kernel PCA with different kernel matrices. The construction of a kernel matrix is equivalent to mapping the data to points $p_1, \ldots, p_m$ in a Hilbert space so that $K_{ij} = \langle p_i, p_j \rangle$ is positive definite. For Isomap, the kernel matrix is related to the Dijkstra shortest path distance between the points; for graph Laplacians, the kernel is related to commute times; and for LLE, the kernel can be associated with a specially constructed graph operator.

Note that the kernel matrix in all these algorithms is defined only on the training data. Moreover, in contrast to traditional kernels such as the Gaussian kernel, the element K_{ij} in the kernel matrix not only depends on the inputs x_i and x_j, but also on all the other training points. This can be seen in the figures where the induced feature distance defined by the kernels does not depend simply on distance in the input space. However, for small distances, there appears to be more of a direct relationship indicating the role of local structure in constructing the kernel. The contour maps of the induced feature distance for the graph Laplacian eigenmap and LLE are ellipsoidal in shape, reflecting the difference in eigenvector normalization between the algorithms and kernel PCA.

For all three algorithms, the existence of a kernel formulation indicates that the algorithms may be viewed as a warping of the input space into a feature space where the manifold is flat. We are currently working to elucidate more of the geometrical properties of these types of transformations. Finally, we would also like to thank Sam Roweis, Olivier Chapelle, and Lawrence Saul for useful discussions.

References

[1] D. Aldous and J. Fill. *Reversible markov chains and random walks on graphs*. In preparation.

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

[3] F. Chung and S.-T. Yau. Discrete green's functions. *Journal of Combinatorial Theory (A)*, (91):191–214, 2000.

[4] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman and Hall, London, 1994.

[5] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, 2002.

[6] C. Grimes and D. L. Donoho. When does isomap recover the natural parametrization of families of articulated images. Technical Report 27, Stanford University, 2002.

[7] I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of ICML'2002*, 2002.

[8] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[9] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[10] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[11] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[12] C. K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.