A kernel view of the dimensionality reduction of manifolds

Jihun Ham JHHAM@SEAS.UPENN.EDU
Daniel D. Lee DDLEE@SEAS.UPENN.EDU
Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA

Sebastian Mika MIKA@FIRST.FRAUNHOFER.DE
Fraunhofer FIRST.IDA, Berlin, Germany

Bernhard Schölkopf BS@TUEBINGEN.MPG.DE
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 (Tenenbaum et al., 2000) was originally proposed as a generalization of multidimensional scaling (MDS) (Cox & Cox, 1994). An alternative method known as locally linear embedding (LLE) (Roweis & Saul, 2000) 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 (Belkin & Niyogi, 2003). 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.

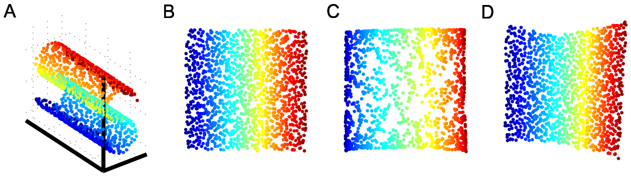

Figure 1. The two-dimensional embeddings resulting from Isomap (B), Laplacian eigenmap (C), and LLE (D) from 1350 points sampled from the S-curve manifold (A). $K = 10$ nearest neighborhoods were used for computing the embeddings.

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 (Schölkopf et al., 1998).

Previous studies have pointed out relationships among various manifold learning algorithms (Weiss, 1999; Williams, 2001; Bengio et al., 2004). In this work, we specifically relate Isomap, graph Laplacians, and

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

LLE within a kernel framework. Regarded in this context, the three algorithms all share a similar strategy. They construct an implicit mapping of the training points to a feature space which preserves some aspect of manifold structure in the data. This mapping is described by a kernel matrix which represents the inner products between the points in feature space. Diagonalization of this kernel matrix then gives rise to an embedding that captures the low-dimensional structure of the manifold.

The resulting kernel matrices are distinctive in several ways. Unlike typical kernels, these kernels do not possess an explicit functional form, and thus can be analytically characterized only in the limit of infinite sampling (Bengio et al., 2004). We give a graph operator interpretation of the kernel that is different from (Kondor & Lafferty, 2002), that yields a natural interpretation of a proper metric on the graph. Also, we show empirically that the kernel matrices defined by these algorithms are consistent in the limit of large data.

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. Empirical results on several examples are provided to illustrate the properties of the resulting kernel matrices. We conclude with a discussion of the similarities and differences between the various 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 $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$.[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 $\mathcal{H}$,

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

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

$$\mathbf{C}\mathbf{v} = \lambda \mathbf{v} \qquad (2)$$

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

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

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

$$m\lambda \boldsymbol{\alpha} = K \boldsymbol{\alpha}, \qquad (4)$$

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

$$\langle \mathbf{v}_p, \Phi(x) \rangle = \frac{1}{\sqrt{\lambda_p}} \sum_{i=1}^{m} \alpha_{p,i} k(x_i, x). \qquad (5)$$

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 $\alpha_{p,n}$. This can be seen as follows: Substituting $x = x_n$ in (5), we get

$$\begin{aligned} \langle \mathbf{v}_p, \Phi(x_n) \rangle &= \frac{1}{\sqrt{\lambda_p}} (K \boldsymbol{\alpha}_p)_n \\ &= \frac{1}{\sqrt{\lambda_p}} (\lambda_p \boldsymbol{\alpha}_p)_n = \sqrt{\lambda_p} \alpha_{p,n}. \qquad (6) \end{aligned}$$

Finally, we should mention one modification. In Eq. (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

$$K' = (I - \boldsymbol{e}\boldsymbol{e}^T) K (I - \boldsymbol{e}\boldsymbol{e}^T) \qquad (7)$$

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

[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.

3. Isomap

As in multidimensional scaling (MDS), Isomap first constructs a matrix of pairwise distances between the different data points (Tenenbaum et al., 2000). 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 (de Silva & Tenenbaum, 2002)). Dijkstra's algorithm is next 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 (Williams, 2001), 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":

$$K_{\mathrm{Isomap}} = -\frac{1}{2}(I - \boldsymbol{e}\boldsymbol{e}^T)S(I - \boldsymbol{e}\boldsymbol{e}^T), \qquad (8)$$

where S is the matrix of squared distance, and $\boldsymbol{e} = m^{-1/2}(1, \ldots, 1)^T$ 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 (Grimes & Donoho, 2002). 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 (Schölkopf & Smola, 2002); 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. In the figure are the resulting spectrum of K_{Isomap} and plots of the associated metric distances in S.

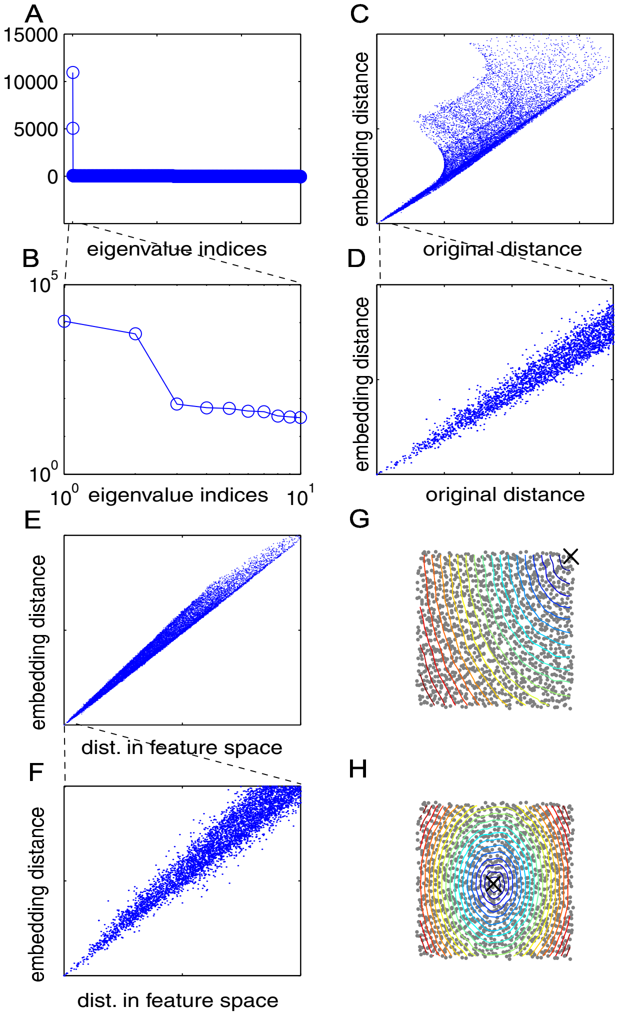

Figure 2. The spectrum of K_{Isomap} for the S-curve is shown on a linear (A) and log-scale (B). A change in slope is noticeable between the second and the third eigenvalues. The pairwise distances of the embedding are compared to the distances in the original input space on a global (C) and local (D) scale. Embeddings are computed using the two eigenvectors with the largest eigenvalues of the kernel K_{Isomap}. The distances in this two-dimensional subspace are compared with distances in feature space under K_{Isomap} in (E) and (F). The embedding is superimposed with contour plots of distance in feature space from a point (marked with an ×) on the boundary (G) and from a point in the center (H). The contour plots with perfect dimensionality reduction would look like ellipses, with the eccentricity of the ellipses reflecting the difference in the two largest eigenvalues of K_{Isomap}. Note that the linearity of (E) and (F) indicates that we have found a good kernel.

4. Graph Laplacian

The graph Laplacian eigenmap algorithm (Belkin & Niyogi, 2003) directly incorporates a 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 an ϵ-ball 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 ($i \sim j$), otherwise $W_{ij} = 0$. The symmetric, non-zero weights in W can be chosen from $\{0, 1\}$, or according to a Gaussian dropoff $W_{ij} = e^{-|x_i - x_j|^2/2\sigma^2}$ 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 the rescaling $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ where 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 uniform vector $\boldsymbol{e}$.

The role of the graph Laplacian for dimensionality reduction was motivated (Belkin & Niyogi, 2003) 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:

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

This quadratic form also explicitly shows that L is positive definite. Optimal embeddings are then given by minimizing Eq. (10). The optimal solutions to this minimization are given by the eigenvectors of L with the smallest eigenvalues, excluding the uniform vector $\boldsymbol{e}$. Using a different normalization constraint on the optimization gives rise to optimal embeddings described by the eigenvectors of the normalized graph Laplacian $\mathcal{L}$ instead.

The optimal solution of Eq. (10) can also be interpreted as finding the eigenvectors with the largest eigenvalues of the pseudo-inverse $L^\dagger$. Thus, the graph Laplacian eigenmap algorithm is equivalent to kernel PCA using $L^\dagger$ as a kernel matrix. Note that since $L\boldsymbol{e} = L^\dagger \boldsymbol{e} = 0$, then $L^\dagger = (I - \boldsymbol{e}\boldsymbol{e}^T) L^\dagger (I - \boldsymbol{e}\boldsymbol{e}^T)$ so that this kernel matrix is automatically centered. We next give an interpretation of this kernel matrix in terms of a metric described by the commute times on the graph.

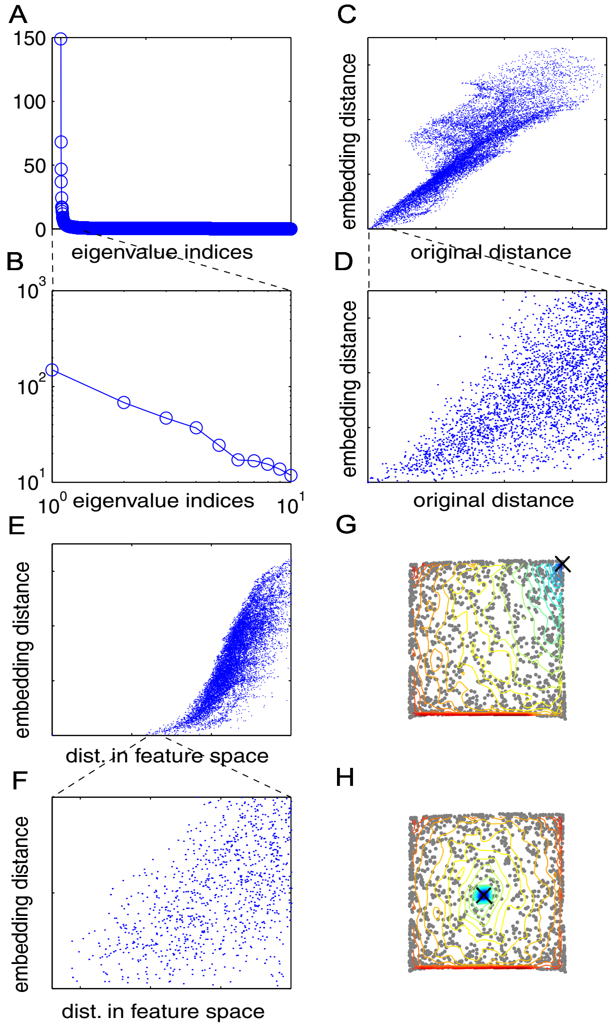

Figure 3. The spectrum of K_L for the S-curve is shown on a linear (A) and log-scale (B). The pairwise distances of the two-dimensional embedding computed using K_L are compared to the distances in the original input space on a global (C) and local (D) scale. The distances in this two-dimensional subspace are compared with distances in feature space under K_L in (E) and (F). The embedding is superimposed with contour plots of distance in feature space from a point (marked with an ×) on the boundary (G) and from a point in the center (H).

4.1. Diffusion kernel

It is known that the graph Laplacian is closely related to a description of diffusion on the graph (Kondor & Lafferty, 2002). 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 (Kondor & Lafferty, 2002):

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

where ϕ_p are the eigenvectors (forming a complete orthonormal system) and λ_p are the eigenvalues of L, i.e., $L = \sum_p \phi_p \phi_p^T \lambda_p$. In terms of the heat kernel, the generic solution to Eq. (11) is given by:

$$\psi(t) = K_t \psi(0) = \sum_p \phi_p^T \psi(0) e^{-\lambda_p t} \phi_p, \tag{13}$$

where $\psi(0)$ is the initial state of the field at time $t = 0$. It is clear that the eigenvectors ϕ_p of L with the smallest eigenvalues correspond to the most slowly decaying modes under diffusion. In particular, the uniform vector $\phi_1 = \boldsymbol{e}$ with zero eigenvalue $\lambda_1 = 0$ is the stationary distribution since $K_t \boldsymbol{e} = \boldsymbol{e}$.

We define $\delta\psi(t)$ as the deviation from the long-time stationary distribution

$$\delta\psi(t) = \psi(t) - \psi(\infty) = K_t \psi(0) - \boldsymbol{e}. \tag{14}$$

We consider statistics of this field under random realizations of the inital conditions $\psi(0)$ such that the mean $\langle \psi(0) \rangle = \boldsymbol{e}$. In this case, the heat kernel can then be related to the covariance of the time evolved field (Kondor & Lafferty, 2002):

$$\begin{aligned} \langle \psi(t)\psi(t)^T \rangle &= K_t \langle \psi(0)\psi(0)^T \rangle K_t & (15) \\ \langle \delta\psi(t)\delta\psi(t)^T \rangle &= K_t \langle \psi(0)\psi(0)^T \rangle K_t - \boldsymbol{e}\boldsymbol{e}^T. & (16) \end{aligned}$$

Assuming that the variance $\langle \psi_i(0)\psi_j(0)^T \rangle = \delta_{ij}$ is independent of the components of $\psi(0)$, the covariance matrix in Eq. (16) can be written as: $\langle \delta\psi(t)\delta\psi(t)^T \rangle = K_{2t} - \boldsymbol{e}\boldsymbol{e}^T$. We then integrate this covariance matrix over time to get the positive definite (kernel) matrix:

$$\begin{aligned} & \int_0^\infty \langle \delta\psi(t)\delta\psi(t)^T \rangle \, dt & (17) \\ = & \int_0^\infty \left[\sum_{p=1}^\infty \phi_p \phi_p^T e^{-2\lambda_p t} - \boldsymbol{e}\boldsymbol{e}^T \right] dt & (18) \\ = & \sum_{p=2}^\infty \phi_p \phi_p^T \int_0^\infty e^{-2\lambda_p t} \, dt & (19) \\ = & \frac{1}{2} \sum_{p=2}^\infty \frac{\phi_p \phi_p^T}{\lambda_p} = \frac{1}{2} L^\dagger. & (20) \end{aligned}$$

As before, $L^\dagger$ is the pseudo-inverse of the graph Laplacian, also known as the discrete Green's function (Chung & Yau, 2000).

We now show that $L^\dagger$ can indeed be considered a proper "kernel" for the graph by relating it to a proper metric distance. The matrix $-L_{ij}$ in Eq. (11) can be regarded as the transition rates of a continuous-time Markov chain (Aldous & Fill, 2002). In this interpretation, the evolving field $\psi_i(t)$ can be interpreted as a probability distribution describing the likelihood of occupying state i at time t, given initial probabilities $\psi(0)$. The statistics of this Markov chain is described by the fundamental matrix, which is equivalent to the pseudo-inverse of L:

$$Z = \int_0^\infty \left[\exp(-Lt) - \boldsymbol{e}\boldsymbol{e}^T \right] dt = L^\dagger. \tag{21}$$

An element Z_{ij} of the fundamental matrix is related to the expected time spent in node j starting from node i under the Markov process. From this fundamental matrix, we can derive the commute time C_{ij}, the expected time for the Markov chain to start from node i, reach node j, and then return to node i (Aldous & Fill, 2002):

$$\begin{aligned} C_{ij} &= m(Z_{ii} + Z_{jj} - Z_{ij} - Z_{ji}) & (22) \\ &= m(L^\dagger_{ii} + L^\dagger_{jj} - L^\dagger_{ij} - L^\dagger_{ji}). & (23) \end{aligned}$$

The commute times are nonnegative, $C_{ij} \geq 0$, symmetric, $C_{ij} = C_{ji}$, and satisfy the triangle inequality, $C_{ij} \leq C_{ik} + C_{kj}$. Thus, the commute times are a proper induced metric on the graph under this Markov process. From Eq. (22), we see that the commute times are also directly related to an inner product relationship given by a kernel matrix $K_L = L^\dagger$. The graph Laplacian algorithm is therefore equivalent to performing kernel PCA on the kernel matrix K_L that is associated with the commute times of diffusion on the underlying graph. As in our analysis of the Isomap algorithm, the graph Laplacian algorithm can also be regarded as multidimensional scaling on 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.

This analysis also provides insight into the difference between Isomap and the graph Laplacian algorithm. The former is based upon shortest paths on the graph induced by the data points, whereas the latter uses commute times of a Markov chain on the graph. In other words, the graph Laplacian algorithm not only considers the shortest path, but integrates over all paths connecting points on the graph to derive its kernel matrix.

5. LLE

The LLE algorithm (Roweis & Saul, 2000) 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 $M := (I - W^T)(I - W)$, 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 $\boldsymbol{e}$. Since the other eigenvectors are orthogonal to $\boldsymbol{e}$, 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

$$K := (\lambda_{max} I - M), \tag{24}$$

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 (Schölkopf & Smola, 2002, Exercise 14.17); see also (Bengio et al., 2004). However, the link between kernel PCA and LLE goes further than that. Equivalently, we can project out the uniform vector $\boldsymbol{e}$, and then use the eigenvectors $1, \ldots, d$ of the resulting matrix as

$$(I - \boldsymbol{e}\boldsymbol{e}^T)K(I - \boldsymbol{e}\boldsymbol{e}^T). \tag{25}$$

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

So 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.

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

$$k(x_i, x_j) = (SDS^T)_{ij} = \langle S_i, DS_j \rangle = \left\langle \sqrt{D}S_i, \sqrt{D}S_j \right\rangle. \tag{26}$$

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 (Belkin & Niyogi, 2003). However, LLE differs from other spectral graph techniques in its construction of M by explicitly minimizing $\sum_{ij} M_{ij} \langle x_i, x_j \rangle$ 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:

$$\frac{\partial \psi(t)}{\partial t} = -M\psi(t), \tag{27}$$

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 Eq. (27).

However, the interpretation of Eq. (27) is somewhat different from diffusion on a graph in that off-diagonal elements of M may be both positive and negative, and thus cannot be described by simple dissipative diffusion. One physical interpretation of Eq. (27) is to relate M to a quadratic energy coupling, and $\psi(t)$ to the positions of a set of colored particles. Depending on the colors of the particles, they may either interact with attractive or repulsive linear forces in a pairwise manner. The eigenmodes of M with the smallest eigenvalues would then correspond to the lowest energy modes of this interacting system.

We can then 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 graph diffusion kernels, this kernel is related to the covariance of the time evolved fields under Eq. (27). 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.

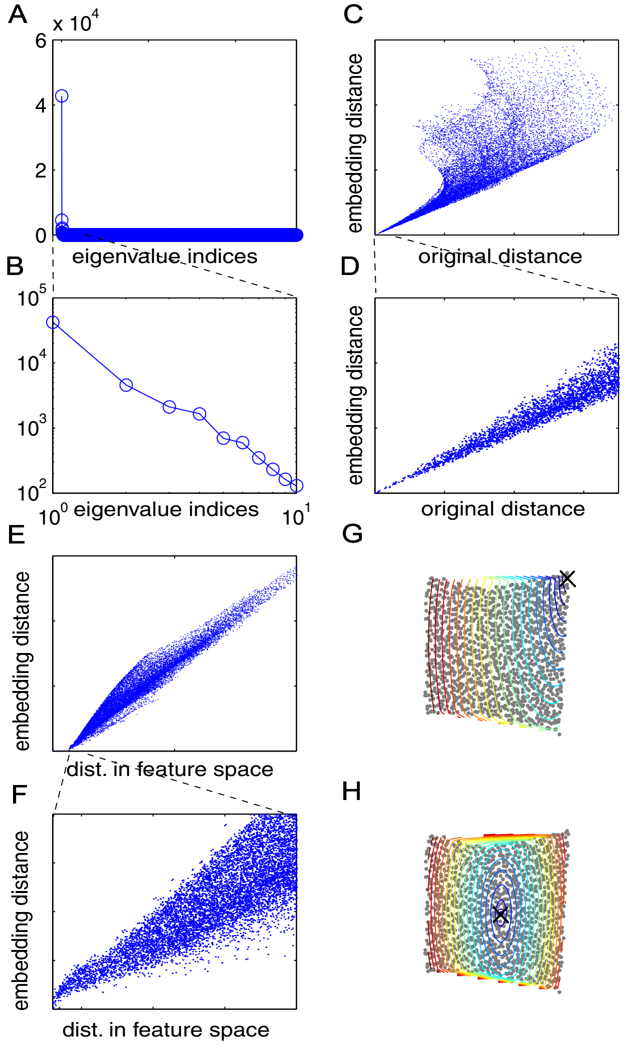

Figure 4. The spectrum of $K_\dagger$ for the S-curve is shown on a linear (A) and log-scale (B). The pairwise distances of the two-dimensional embedding computed using K_L are compared to the distances in the original input space on a global (C) and local (D) scale. The distances in this two-dimensional subspace are compared with distances in feature space under K_L in (E) and (F). The embedding is superimposed with contour plots of distance in feature space from a point (marked with an ×) on the boundary (G) and from a point in the center (H).

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 matrices in all these algorithms are 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 experimental results where the induced feature distance defined by the kernels does not depend simply on distance in the input space. However, there does appears to be more of a direct relationship at small distances indicating some local structure in the construction of the kernel. The contour maps of the induced feature distance for the three algorithms are generally ellipsoidal in shape, reflecting the difference in eigenvector normalization between the algorithms and KPCA.

We also can empirically test to see how consistent the elements of the defined kernel matrices are under different data samplings of the manifold. Fig. 5 shows the representative behavior of several different kernel matrix coefficients as the number of data points changes. After normalizing for an overall scale factor in the kernel matrices that does not influence the resulting embeddings, we see that the kernel coefficients are relatively stable to different samplings. This indicates that the kernel matrices as defined are not inconsistent under these empirical data distributions.

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. This warping is defined using the local neighborhood structure in the data. The embedding is then calculated by projecting these vectors onto a low dimensional subspace. We are currently working to better elucidate the geometrical properties of these kernel matrices.

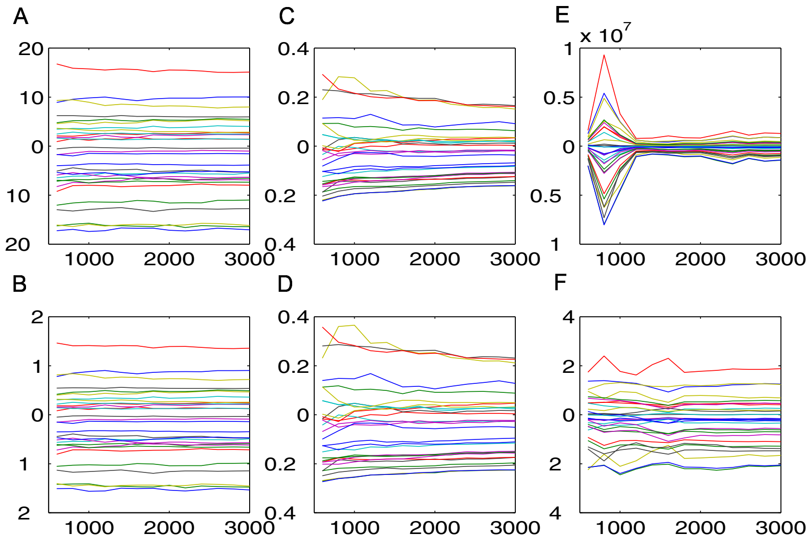

Figure 5. Representative coefficients in the kernel matrices for the three algorithms: Isomap (A,B), Laplacian eigenmap (C,D), and LLE (E,F). The coefficients are calculated with varying numbers (600–3000) of data points. Starting with $m = 600$ samples, additional data points were appended and the kernel matrices were recalculated until $m = 3000$. (A-C) show the behavior of the raw kernel matrix coefficients, and (D-F) show the matrix coefficients normalized by an overall scale factor calculated from the trace of kernel matrix. The fluctuations in the normalized coefficients generally decrease with larger data sizes.

Acknowledgments

JHH and DDL acknowledge the support of the U.S. National Science Foundation and the U.S. Army Research Office. Finally, we would also like to thank Sam Roweis, Olivier Chapelle, and Lawrence Saul for useful discussions.

References

Aldous, D., & Fill, J. (2002). Reversible Markov chains and random walks on graphs. In preparation.

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

Bengio, Y., Paiement, J.-F., & Vincent, P. (2004). Out-of-sample extension for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in Neural Information Processing Systems 15*. MIT Press.

Chung, F., & Yau, S. (2000). Discrete green's function. *Journal of Combinatorical Theory (A)*, *91*, 191–214.

Cox, T., & Cox, M. (1994). *Multidimensional scaling*. London: Chapman and Hall.

de Silva, V., & Tenenbaum, J. (2002). Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, *15*.

Grimes, C., & Donoho, D. (2002). *When does isomap recover the natural parameterization of families of articulated images?* (Technical Report 27). Stanford University.

Kondor, I., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete structures. *Proceedings of ICML'2002*.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Tenenbaum, J., de Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Weiss, Y. (1999). Segmentation using eigenvectors: a unifying view. *IEEE Proceedings of ICCV* (pp. 975–982).

Williams, C. K. I. (2001). On a connection between kernel PCA and metric multidimensional scaling. *Advances in Neural Information Processing Systems 13*. MIT Press.