Analysis and Extension of Spectral Methods for Nonlinear Dimensionality Reduction

Fei Sha FEISHA@CIS.UPENN.EDU
Lawrence K. Saul LSAUL@CIS.UPENN.EDU
Department of Computer & Information Science, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104

Abstract

Many unsupervised algorithms for nonlinear dimensionality reduction, such as locally linear embedding (LLE) and Laplacian eigenmaps, are derived from the spectral decompositions of sparse matrices. While these algorithms aim to preserve certain proximity relations on average, their embeddings are not explicitly designed to preserve local features such as distances or angles. In this paper, we show how to construct a low dimensional embedding that maximally preserves angles between nearby data points. The embedding is derived from the bottom eigenvectors of LLE and/or Laplacian eigenmaps by solving an additional (but small) problem in semidefinite programming, whose size is independent of the number of data points. The solution obtained by semidefinite programming also yields an estimate of the data's intrinsic dimensionality. Experimental results on several data sets demonstrate the merits of our approach.

1. Introduction

The problem of discovering low dimensional structure in high dimensional data arises in many areas of information processing (Burges, 2005). Much recent work has focused on the setting in which such data is assumed to have been sampled from a low dimensional submanifold. Many algorithms, based on variety of geometric intuitions, have been proposed to compute low dimensional embeddings in this setting (Roweis & Saul, 2000; Tenenbaum et al., 2000; Belkin & Niyogi, 2003; Donoho & Grimes, 2003; Weinberger & Saul, 2004). In contrast to linear methods such as principal component analysis (PCA), these "manifold learning" algorithms are capable of discovering highly nonlinear structure. Nevertheless, their main optimizations are quite tractable—involving (for example) nearest neighbor searches, least squares fits, dynamic programming, eigenvalue problems, and semidefinite programming.

One large family of algorithms for manifold learning consists of approaches based on the spectral decomposition of sparse matrices (Chung, 1997). Algorithms in this family include locally linear embedding (LLE) (Roweis & Saul, 2000) and Laplacian eigenmaps (Belkin & Niyogi, 2003). The matrices in these algorithms are derived from sparse weighted graphs whose nodes represent high dimensional inputs and whose edges indicate neighborhood relations. Low dimensional embeddings are computed from the bottom eigenvectors of these matrices. This general approach to manifold learning is attractive for computational reasons because it reduces the main problem to solving a sparse eigensystem. In addition, the resulting embeddings tend to preserve proximity relations without imposing the potentially rigid constraints of isometric (distance-preserving) embeddings (Tenenbaum et al., 2000; Weinberger & Saul, 2004). On the other hand, this general approach also has several shortcomings: (i) the solutions do not yield an estimate of the underlying manifold's dimensionality; (ii) the geometric properties preserved by these embedding are difficult to characterize; (iii) the resulting embeddings sometimes exhibit an unpredictable dependence on data sampling rates and boundary conditions.

In the first part of this paper, we review LLE and Laplacian eigenmaps and provide an extended analysis of these shortcomings. As part of this analysis, we derive a theoretical result relating the distribution of smallest eigenvalues in these algorithms to a data set's intrinsic dimensionality.

In the second part of the paper, we propose a framework to remedy the key deficiencies of LLE and Laplacian eigenmaps. In particular, we show how to construct a more robust, angle-preserving embedding from the spectral decompositions of these algorithms (one of which must be run as a first step). The key aspects of our framework are the following: (i) a d-dimensional embedding is computed from the m bottom eigenvectors of LLE or Laplacian eigenmaps with $m > d$, thus incorporating informa-

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005.

tion that the original algorithm would have discarded for a similar result; (ii) the new embeddings explicitly optimize the degree of neighborhood similarity—that is, equivalence up to rotation, translation, and scaling—with the aim of discovering conformal (angle-preserving) mappings; (iii) the required optimization is performed by solving an additional (but small) semidefinite program (Vandenberghe & Boyd, 1996), whose size is independent of the number of data points; (iv) the solution of the semidefinite program yields an estimate of the underlying manifold's dimensionality. Finally, we present experimental results on several data sets, including comparisons with other algorithms.

2. Analysis of Existing Methods

The problem of manifold learning is simply stated. Assume that high dimensional inputs have been sampled from a low dimensional submanifold. Denoting the inputs by $\{\vec{x}_i\}_{i=1}^n$ where $\vec{x}_i \in \mathcal{R}^p$, the goal is to compute outputs $\{\vec{y}_i\}_{i=1}^n$ that provide a faithful embedding in $d \ll p$ dimensions.

LLE and Laplacian eigenmaps adopt the same general framework for solving this problem. In their simplest forms, both algorithms consist of three steps: (i) construct a graph whose nodes represents inputs and whose edges indicate k-nearest neighbors; (ii) assign weights to the edges in the graph and use them to construct a sparse positive semidefinite matrix; (iii) output a low dimensional embedding from the bottom eigenvectors of this matrix. The main practical difference between the algorithms lies in the second step of choosing weights and constructing a cost function. We briefly review each algorithm below, then provide an analysis of their particular shortcomings.

2.1. Locally linear embedding

LLE appeals to the intuition that each high dimensional input and its k-nearest neighbors can be viewed as samples from a small linear "patch" on a low dimensional submanifold. Weights W_{ij} are computed by reconstructing each input $\vec{x}_i$ from its k-nearest neighbors. Specifically, they are chosen to minimize the reconstruction error:

$$\mathcal{E}(W) = \sum_i \left\| \vec{x}_i - \sum\nolimits_j W_{ij}\vec{x}_j \right\|^2. \tag{1}$$

The minimization is performed subject to two constraints: (i) $W_{ij}=0$ if $\vec{x}_j$ is not among the k-nearest neighbors of $\vec{x}_i$; (ii) $\sum_j W_{ij} = 1$ for all i. The weights thus constitute a sparse matrix $\mathbf{W}$ that encodes local geometric properties of the data set by specifying the relation of each input $\vec{x}_i$ to its k-nearest neighbors.

LLE constructs a low dimensional embedding by computing outputs $\vec{y}_i \in \mathcal{R}^d$ that respect these same relations to their k-nearest neighbors. Specifically, the outputs are chosen to minimize the cost function:

$$\Phi(Y) = \sum_i \left\| \vec{y}_i - \sum\nolimits_j W_{ij}\vec{y}_j \right\|^2. \tag{2}$$

The minimization is performed subject to two constraints that prevent degenerate solutions: (i) the outputs are centered, $\sum_i \vec{y}_i = 0$, and (ii) the outputs have unit covariance matrix. The d-dimensional embedding that minimizes eq. (2) subject to these constraints is obtained by computing the bottom $d+1$ eigenvectors of the matrix $\mathbf{\Phi} = (\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$. The bottom (constant) eigenvector is discarded, and the remaining d eigenvectors (each of size n) yield the embedding $\vec{y}_i \in \mathcal{R}^d$ for $i \in \{1, 2, \ldots, n\}$.

2.2. Laplacian eigenmaps

Laplacian eigenmaps also appeal to a simple geometric intuition: namely, that nearby high dimensional inputs should be mapped to nearby low dimensional outputs. To this end, a positive weight W_{ij} is associated with inputs $\vec{x}_i$ and $\vec{x}_j$ if either input is among the other's k-nearest neighbors. Typically, the values of the weights are either chosen to be constant, say $W_{ij} = 1/k$, or exponentially decaying, as $W_{ij} = \exp(-\|\vec{x}_i - \vec{x}_j\|^2/\sigma^2)$ where σ^2 is a scale parameter. Let $\mathbf{D}$ denote the diagonal matrix with elements $D_{ii} = \sum_j W_{ij}$. The outputs $\vec{y}_i$ can be chosen to minimize the cost function:

$$\Psi(Y) = \sum_{ij} \frac{W_{ij}\,\|\vec{y}_i - \vec{y}_j\|^2}{\sqrt{D_{ii}D_{jj}}}. \tag{3}$$

As in LLE, the minimization is performed subject to constraints that the outputs are centered and have unit covariance. The embedding is computed from the bottom eigenvectors of the matrix $\mathbf{\Psi} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$. The matrix $\mathbf{\Psi}$ is a symmetrized, normalized form of the graph Laplacian, given by $\mathbf{D}-\mathbf{W}$. Again, the optimization is a sparse eigenvalue problem that scales well to large data sets.

2.3. Shortcomings for manifold learning

Both LLE and Laplacian eigenmaps can be viewed as spectral decompositions of weighted graphs (Belkin & Niyogi, 2003; Chung, 1997). The complete set of eigenvectors of the matrix $\mathbf{\Phi}$ (in LLE) and $\mathbf{\Psi}$ (in Laplacian eigenmaps) yields an orthonormal basis for functions defined on the graph whose nodes represent data points. The eigenvectors of LLE are ordered by the degree to which they reflect the local linear reconstructions of nearby inputs; those of Laplacian eigenmaps are ordered by the degree of smoothness, as measured by the discrete graph Laplacian. The bottom eigenvectors from these algorithms often produce reasonable embeddings. The orderings of these eigenvectors, however, do not map precisely onto notions of local

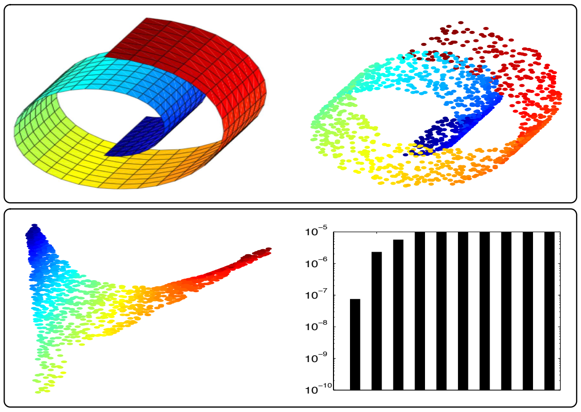

Figure 1. Top. Data set of $n = 1000$ inputs randomly sampled from a "Swiss roll". *Bottom.* Two dimensional embedding and ten smallest nonzero eigenvalues computed by LLE.

distance or angle preservation, nor do their smallest eigenvalues have a telltale gap that yields an estimate of the underlying manifold's dimensionality. Evidence and implications of these shortcomings are addressed in the next sections.

2.4. Empirical results

Fig. 1 shows the results of LLE applied to $n = 1000$ inputs sampled from a "Swiss roll" (using $k = 6$ nearest neighbors). The ten smallest nonzero eigenvalues of the matrix $\mathbf{\Phi}$ are also plotted on a log scale. While LLE does unravel this data set, the aspect ratio and general shape of its solution do not reflect the underlying manifold. There is also no apparent structure in its bottom eigenvalues (such as a prominent gap) to suggest that the inputs were sampled from a two dimensional surface. Such results are fairly typical: while the algorithm often yields embeddings that preserve proximity relations on average, it is difficult to characterize their geometric properties more precisely. This behavior is also manifested by a somewhat unpredictable dependence on the data sampling rate and boundary conditions. Finally, in practice, the number of nearly zero eigenvalues has not been observed to provide a robust estimate of the underlying manifold's dimensionality (Saul & Roweis, 2003).

2.5. Theoretical analysis

How do the smallest eigenvalues of LLE and Laplacian eigenmaps reflect the underlying manifold's dimensionality, if at all? In this section, we analyze a simple setting in which the distribution of smallest eigenvalues can be precisely characterized. Our analysis does in fact reveal a mathematical relationship between these methods' eigenspectra and the intrinsic dimensionality of the data set. We suspect, however, that this relationship is not likely to be of much practical use for estimating dimensionality.

Consider inputs $\vec{x}_i \in \mathcal{R}^d$ that lie on the sites of an infinite d-dimensional hypercubic lattice. Each input has $2d$ neighbors separated by precisely one lattice spacing. The two dimensional case is illustrated in the left panel of Fig. 2. Choosing $k = 2d$ nearest neighbors to construct a sparse graph and assigning constant weights to the edges, we obtain an (infinite) weight matrix $\mathbf{W}$ for Laplacian eigenmaps given by:

$$W_{ij} = \begin{cases} \frac{1}{2d} & \text{if } \|\vec{x}_i - \vec{x}_j\| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

A simple calculation shows that for this example, Laplacian eigenmaps are based on the spectral decomposition of the matrix $\mathbf{\Psi} = \mathbf{I} - \mathbf{W}$. Note that LLE would use the same weight matrix $\mathbf{W}$ for these inputs; it would thus perform a spectral decomposition on the matrix $\mathbf{\Phi} = \mathbf{\Psi}^2$. In what follows, we analyze the distribution of smallest eigenvalues of $\mathbf{\Psi}$; the corresponding result for LLE follows from a simple change of variables.

The matrix $\mathbf{\Psi}$ is diagonalized by a Fourier basis: namely, for each $\vec{q} \in [-\pi, \pi]^d$, it has an eigenvector of the form $\{e^{i\vec{q}\cdot\vec{x}_i}\}$ with eigenvalue:

$$\lambda(\vec{q}) = 1 - \frac{1}{d}\sum_{\alpha=1}^{d} \cos q_\alpha. \tag{5}$$

Thus, $\mathbf{\Psi}$ has a continuous eigenspectrum; in particular, *it has no gaps*. We can compute the distribution of its eigenvalues from the integral:

$$\rho_d(\lambda) = \frac{1}{(2\pi)^d}\int_\Omega \delta(\lambda - \lambda(\vec{q})) \tag{6}$$

where $\Omega = [-\pi, \pi]^d$ and $\delta(\cdot)$ denotes the Dirac delta function. For $d = 1$, the integral gives the simple result: $\rho_1(\lambda) = (1/\pi)/\sqrt{\lambda(2-\lambda)}$. For $d > 1$, the integral cannot be evaluated analytically, but we can compute its asymptotic behavior as $\lambda \to 0$, which characterizes the distribution of *smallest* eigenvalues. (This is the regime of interest for understanding LLE and Laplacian eigenmaps.) The asymptotic behavior may be derived by approximating eq. (5) by its second-order Taylor expansion $\lambda(\vec{q}) \approx \|\vec{q}\|^2/(2d)$, since $\lambda \ll 1$ implies $\|\vec{q}\| \ll 1$. With this substitution, eq. (6) reduces to an integral over a hypersphere of radius $\|\vec{q}\| = \sqrt{2d\lambda}$, yielding the asymptotic result:

$$\rho_d(\lambda) \sim \left[\frac{(d/2\pi)^{d/2}}{\Gamma(d/2)}\right]\lambda^{d/2-1} \text{ as } \lambda \to 0, \tag{7}$$

where $\Gamma(\cdot)$ is the Gamma function. Our result thus relates the dimensionality of the input lattice to the power law that characterizes the distribution of smallest eigenvalues.

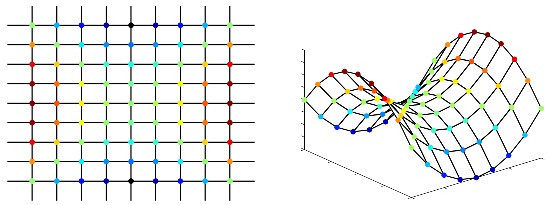

Figure 2. Left: inputs from an infinite square lattice, for which one can calculate the distribution of smallest eigenvalues from LLE and Laplacian eigenmaps. *Right*: inputs from a regularly sampled submanifold which give rise to the same results.

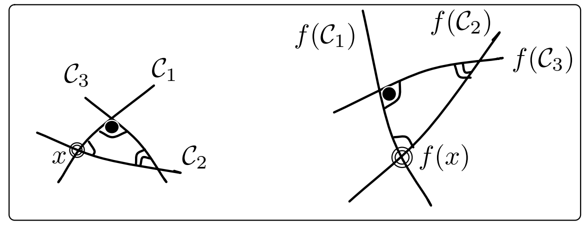

Figure 3. Schematic illustration of conformal mapping f between two manifolds, which preserves the angles (but not areas) of infinitessimally small triangles.

The power laws in eq. (7) were calculated from the weight matrix in eq. (4); thus they are valid for any inputs that give rise to matrices (or equivalently, graphs) of this form. The right panel of Fig. 2 shows how inputs that were regularly sampled from a two dimensional submanifold could give rise to the same graph as inputs from a square lattice. Could these power laws be used to estimate the intrinsic dimensionality of a data set? While in principle such an approach seems possible, in practice it seems unlikely to succeed. Fitting a power law to the distribution of smallest eigenvalues would require computing many more than the d smallest ones. The fit would also be confounded by the effects of finite sample sizes, boundary conditions, and random (irregular) sampling. A more robust way to estimate dimensionality from the results of LLE and Laplacian eigenmaps is therefore needed.

3. Conformal Eigenmaps

In this section, we describe a framework to remedy the previously mentioned shortcomings of LLE and Laplacian eigenmaps. Recall that we can view the eigenvectors from these algorithms as an orthonormal basis for functions defined on the n points of the data set. The embeddings from these algorithms are derived simply from the d bottom (non-constant) eigenvectors; as such, they are not explicitly designed to preserve local features such as distances or angles. Our approach is motivated by the following question: from the m bottom eigenvectors of these algorithms, where $m > d$ but $m \ll n$, can we construct a more faithful embedding of dimensionality d?

3.1. Motivation

To proceed, we must describe more precisely what we mean by "faithful". A conformal mapping f between two manifolds $\mathcal{M}$ and $\mathcal{M}'$ is a smooth, one-to-one mapping that looks locally like a rotation, translation, and scaling, thus preserving local angles (though not local distances). This property is shown schematically in Fig. 3. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ denote two curves intersecting at a point $x \in \mathcal{M}$, and let $\mathcal{C}_3$ denote another curve that intersects $\mathcal{C}_1$ and $\mathcal{C}_2$ in the neighborhood of x. Let Γ be the infinitessimally small triangle defined by these curves in the limit that $\mathcal{C}_3$ approaches x, and let Γ' be its image under the mapping f. For a conformal mapping, these triangles will be similar (having equal angles) though not necessarily congruent (with equal sides and areas). All isometric (distance-preserving) mappings are conformal mappings, but not vice versa.

We can now more precisely state the motivation behind our approach. Assume that the inputs $\vec{x}_i \in \mathcal{R}^p$ were sampled from a submanifold that can be conformally mapped to d-dimensional Euclidean space. Can we (approximately) construct the images of $\vec{x}_i$ under such a mapping from the bottom m eigenvectors of LLE or Laplacian eigenmaps, where $m > d$ but $m \ll n$? We will answer this question in three steps: first, by defining a measure of local dissimilarity between discrete point sets; second, by minimizing this measure over an m-dimensional function space derived from the spectral decompositions of LLE and Laplacian eigenmaps; third, by casting the required optimization as a problem in semidefinite programming and showing that its solution also provides an estimate of the data's intrinsic dimensionality d.

3.2. Cost function

Let $\vec{z}_i = f(\vec{x}_i)$ define a mapping between two discrete point sets for $i \in \{1, 2, \ldots, n\}$. Suppose that the points $\vec{x}_i$ were densely sampled from a low dimensional submanifold: to what extent does this mapping look locally like a rotation, translation, and scaling? Consider any triangle $(\vec{x}_i, \vec{x}_j, \vec{x}_{j'})$ where j and j' are among the k-nearest neighbors of $\vec{x}_i$, as well as the image of this triangle $(\vec{z}_i, \vec{z}_j, \vec{z}_{j'})$. These triangles are similar if and only if:

$$\frac{\|\vec{z}_i - \vec{z}_j\|^2}{\|\vec{x}_i - \vec{x}_j\|^2} = \frac{\|\vec{z}_i - \vec{z}_{j'}\|^2}{\|\vec{x}_i - \vec{x}_{j'}\|^2} = \frac{\|\vec{z}_j - \vec{z}_{j'}\|^2}{\|\vec{x}_j - \vec{x}_{j'}\|^2}$$

The constant of proportionality in this equation indicates the change in scale of the similarity transformation (i.e., the ratio of the areas of these triangles). Let $\eta_{ij} = 1$ if $\vec{x}_j$ is one of the k-nearest neighbors of $\vec{x}_i$ or if $i = j$. Also, let s_i denote the hypothesized change of scale induced by

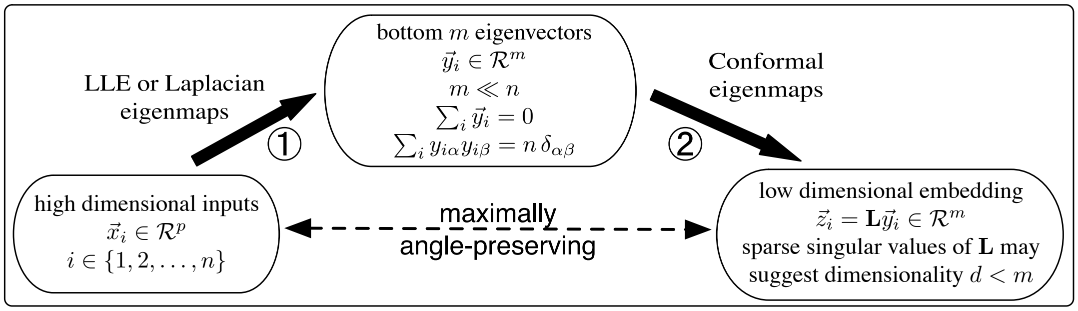

Figure 4. Steps of the algorithm for computing maximally angle-preserving embeddings from the bottom eigenvectors of LLE or Laplacian eigenmaps, as described in section 3.

the mapping at $\vec{x}_i$. With this notation, we can measure the degree to which the mapping f takes triangles at $\vec{x}_i$ into similar triangles at $\vec{z}_i$ by the cost function:

$$\mathcal{D}_i(s_i) = \sum_{jj'} \eta_{ij}\eta_{ij'} \left(\|\vec{z}_j - \vec{z}_{j'}\|^2 - s_i\|\vec{x}_j - \vec{x}_{j'}\|^2\right)^2 \tag{8}$$

A global measure of local dissimilarity is obtained by summing over all points:

$$\mathcal{D}(s) = \sum_i \mathcal{D}_i(s_i). \tag{9}$$

Note that given fixed point sets $\{\vec{x}_i\}_{i=1}^n$ and $\{\vec{z}_i\}_{i=1}^n$, it is straightforward to compute the scale parameters that minimize this cost function, since each s_i in eq. (8) can be optimized by a least squares fit. Is it possible, however, that given only $\{\vec{x}_i\}_{i=1}^n$, one could also minimize this cost function with respect to $\{\vec{z}_i\}_{i=1}^n$, yielding a nontrivial solution where the latter lives in a space of much lower dimensionality than the former? Such a solution (assuming it had low cost) would be suggestive of a conformal map for nonlinear dimensionality reduction.

3.3. Eigenvector expansion

We obtain a well-posed optimization for the above problem by constraining the points $\vec{z}_i$ to be spanned by the bottom m eigenvectors returned by LLE or Laplacian eigenmaps. In particular, denoting the outputs of these algorithms by $\vec{y}_i \in \mathcal{R}^m$, we look for solutions of the form:

$$\vec{z}_i = \mathbf{L}\vec{y}_i, \tag{10}$$

where $\mathbf{L} \in \mathcal{R}^{m\times m}$ is a general linear transformation and $i \in \{1, 2, \ldots, n\}$. Thus our goal is to compute the scale parameters s_i and the linear transformation $\mathbf{L}$ that minimize the "dissimilarity" cost function in eqs. (8–9) with the substitution $\vec{z}_i = \mathbf{L}\vec{y}_i$. We also impose the further constraint

$$\mathrm{trace}(\mathbf{L}^T\mathbf{L}) = 1 \tag{11}$$

in order to rule out the trivial solution $\mathbf{L} = \mathbf{0}$, which zeroes the cost function by placing all $\vec{z}_i$ at the origin.

Before showing how to optimize eqs. (8–9) subject to the constraints in eq. (10–11), we make two observations. First, by equating $\mathbf{L}$ to a multiple of the identity matrix, we recover the original m-dimensional embedding (up to a global change of scale) obtained from LLE or Laplacian eigenmaps. In general, however, we shall see that this setting of $\mathbf{L}$ does not lead to maximally angle-preserving embeddings. Second, if we are allowed to express $\vec{z}_i$ in terms of the *complete* basis of eigenvectors (taking $m = n$), then we can reconstruct the original inputs $\vec{x}_i$ up to a global rotation and change of scale, which does not yield any form of dimensionality reduction. By choosing $m \ll \min(n, p)$ where $\vec{x}_i \in \mathcal{R}^p$, however, we can force a solution that achieves some form of dimensionality reduction.

3.4. Semidefinite programming

Minimizing the cost function in eqs. (8–9) in terms of the matrix $\mathbf{L}$ and the scale parameters s_i can be cast as a problem in semidefinite programming (SDP) (Vandenberghe & Boyd, 1996). Details are omitted due to lack of space, but the derivation is easy to understand at a high level. The optimal scaling parameters can be computed in closed form as a function of the matrix $\mathbf{L}$ and eliminated from eqs. (8–9). The resulting form of the cost function only depends on the matrix $\mathbf{L}$ through the positive semidefinite matrix $\mathbf{P} = \mathbf{L}^T\mathbf{L}$. Let $\mathcal{P} = \mathrm{vec}(\mathbf{P})$ denote the vector obtained by concatenating the columns of $\mathbf{P}$. Then, using Schur complements, the optimization can be written as:

$$\begin{aligned}
&\text{minimize} && t && (12)\\
&\text{such that} && \mathbf{P} \succeq 0, && (13)\\
& && \mathrm{trace}(\mathbf{P}) = 1, && (14)\\
& && \begin{pmatrix} \mathbf{I} & \mathbf{S}\mathcal{P} \\ (\mathbf{S}\mathcal{P})^T & t \end{pmatrix} \succeq 0, && (15)
\end{aligned}$$

where eq. (13) indicates that the matrix $\mathbf{P}$ is constrained to be positive semidefinite and eq. (14) enforces the earlier constraint from eq. (11). In eq. (15), $\mathbf{I}$ and $\mathbf{S}$ are $m^2 \times m^2$ matrices; $\mathbf{I}$ denotes the identity matrix, while $\mathbf{S}$ depends on $\{\vec{x}_i, \vec{y}_i\}_{i=1}^n$ but is independent of the optimization variables $\mathbf{P}$ and t. The optimization is an SDP over the m^2 unknown elements of $\mathbf{P}$ (or equivalently $\mathcal{P}$). Thus its size is independent of the number of inputs, n, as well as their extrinsic dimensionality, p. For small problems with (say) $m = 10$, these SDPs can be solved in under a few minutes on a typical desktop machine.

After solving the SDP, the linear map $\mathbf{L}$ is recovered via $\mathbf{L} = \mathbf{P}^{1/2}$, and the maximally angle-preserving embedding is computed from eq. (10). The square root operation is well defined since $\mathbf{P}$ is positive semidefinite. The solution computed from $\vec{z}_i = \mathbf{L}\vec{y}_i$ defines an m-dimensional embedding. Note, however, that if the matrix $\mathbf{L}$ has one or more small singular values, then the variance of this embedding will be concentrated in fewer than m dimensions. Thus, by examining the singular values of $\mathbf{L}$ (or equivalently, the eigenvalues of $\mathbf{P} = \mathbf{L}^T\mathbf{L}$), we can obtain an estimate of the data's intrinsic dimensionality, d. We can also output a d-dimensional embedding by simply projecting the points $\vec{z}_i \in \mathcal{R}^m$ onto the first d principal components of their covariance matrix. The overall algorithm, which we call conformal eigenmaps, is summarized in Fig. 4.

4. Experimental Results

We experimented with the algorithm in Fig. 4 to compute maximally angle-preserving embeddings of several data sets. We used the algorithm to visualize the low dimensional structure of each data set, as well as to estimate their intrinsic dimensionalities. We also compared the eigenvalue spectra from this algorithm to those from PCA, Isomap (Tenenbaum et al., 2000), and Semidefinite Embedding (SDE) (Weinberger & Saul, 2004).

4.1. Swiss roll

The top panel of Fig. 5 shows the angle-preserving embedding computed by conformal eigenmaps on the Swiss roll data set from section 2.4. The embedding was constructed from the bottom $m = 10$ eigenvectors of LLE. Compared to the result in Fig. 1, the angle-preserving embedding more faithfully preserves the shape of the underlying manifold's boundary. The eigenvalues of the matrix $\mathbf{P} = \mathbf{L}^T\mathbf{L}$, normalized by their sum, are shown in the middle panel of Fig. 5, along with similarly normalized eigenvalues from PCA, Isomap, and SDE. The relative magnitudes of individual eigenvalues are indicated by the widths of differently colored regions in each bar plot.

The two leading eigenvalues in these graphs reveal the extent to which each algorithm's embedding is confined to two dimensions. All the manifold learning algorithms (but

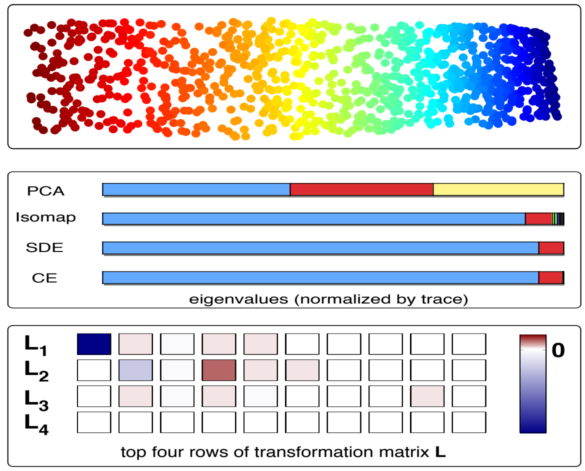

Figure 5. Top: two dimensional embedding of "Swiss roll" inputs in Fig. 1 by conformal eigenmaps (CE). *Middle*: comparison of eigenspectra from PCA, Isomap, SDE, and CE. The relative magnitudes of individual eigenvalues are indicated by the widths of differently colored regions in each bar plot. *Bottom*: top four rows of the transformation matrix $\mathbf{L}$ in eq. (10).

not PCA) correctly identify the underlying dimensionality of the Swiss roll as $d = 2$. Finally, the bottom panel in Fig. 5 shows the top four rows of the transformation matrix $\mathbf{L}$ from eq. (10). Only the first two rows have matrix elements of appreciable magnitude, reflecting the underlying two dimensional structure of this data set. Note, however, that sizable matrix elements appear in *all* the columns of $\mathbf{L}$. This shows that the maximally angle-preserving embedding exploits structure in the bottom $m = 10$ eigenvectors of LLE, not just in the bottom $d = 2$ eigenvectors.

4.2. Images of edges

Fig. 6 shows examples from a synthetic data set of $n = 2016$ grayscale images of edges. Each image has 24×24 resolution. The edges were generated by sampling pairs of points at regularly spaced intervals along the image boundary. The images lie on a two dimensional submanifold, but this submanifold has a periodic structure that cannot be unraveled in the same way as the Swiss roll from the previous section. We synthesized this data set with the aim of understanding how various manifold learning algorithms might eventually perform when applied to patches of natural images.

The top panel of Fig. 7 shows the first two dimensions of the maximally angle-preserving embedding of this data set. The embedding was constructed from the bottom $m = 10$ eigenvectors of Laplacian eigenmaps using $k = 10$ nearest neighbors. The embedding has four distinct quadrants in which edges with similar orientations are mapped to nearby points in the plane. The middle panel in Fig. 7 compares the eigenspectrum from the angle-preserving embed-

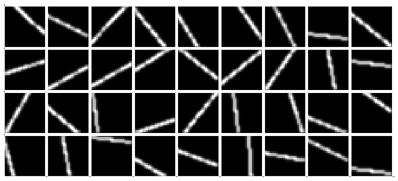

Figure 6. Examples from a synthetic data set of $n = 2016$ grayscale images of edges. Each image has 24×24 resolution.

ding to those from PCA, Isomap, and SDE. Isomap does not detect the low dimensional structure of this data set, possibly foiled by the cyclic structure of the submanifold. The distance-preserving embedding of SDE has variance in more dimensions than the angle-preserving embedding, suggesting that the latter has exploited the extra flexibility of conformal versus isometric maps. The bottom panel in Fig. 7 displays the matrix $\mathbf{L}$ from eq. (10). The result shows that the semidefinite program in eq. (12) mixes all of the bottom $m = 10$ eigenvectors from Laplacian eigenmaps to obtain the maximally angle-preserving embedding.

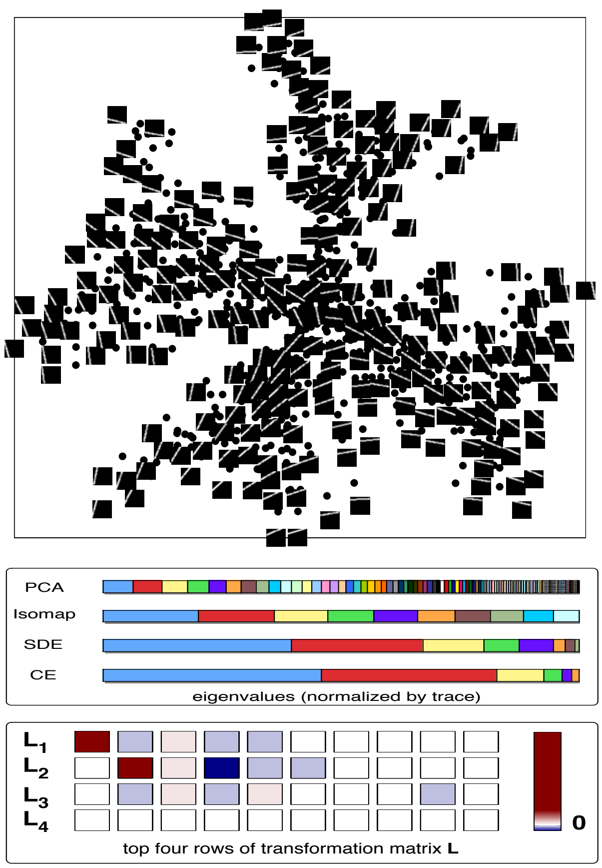

Figure 7. Top: two dimensional embedding of $n = 2016$ edge images by conformal eigenmaps (CE). *Middle:* comparison of eigenspectra from PCA, Isomap, SDE, and CE. *Bottom:* top four rows of the transformation matrix $\mathbf{L}$ in eq. (10).

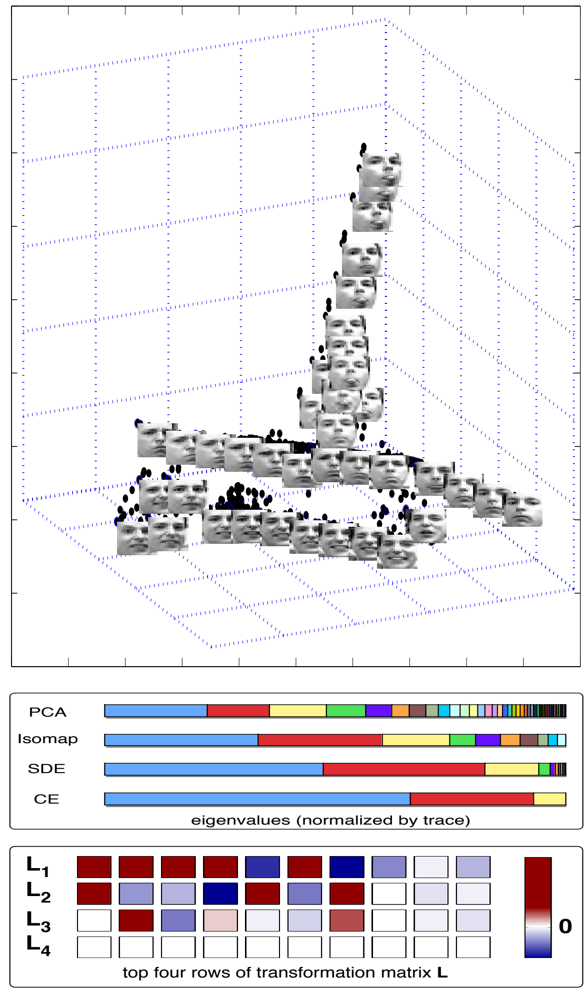

Figure 8. Top: three dimensional embedding of $n = 983$ face images by conformal eigenmaps (CE). *Middle:* comparison of eigenspectra from PCA, Isomap, SDE and CE. *Bottom:* top four rows of the transformation matrix $\mathbf{L}$ in eq. (10).

4.3. Images of faces

Fig. 8 shows results on a data set of $n = 983$ images of faces. The faces were initially processed by LLE using $k = 8$ nearest neighbors. The maximally angle-preserving embedding was computed from the bottom $m = 10$ eigenvectors of LLE. Though the intrinsic dimensionality of this data set is not known a priori, several methods have reported similar findings (Brand, 2003; Weinberger & Saul, 2004). As shown in the middle panel, the embedding from conformal eigenmaps concentrates most of its variance in three dimensions, yielding a somewhat lower estimate of the dimensionality than Isomap or SDE. The top panel of Fig. 8 visualizes the three dimensional embedding from conformal eigenmaps; the faces are arranged in an intuitive manner according to their expression and pose.

5. Discussion

In this work, we have appealed to conformal transformations as a basis for nonlinear dimensionality reduction. Our approach casts a new light on older algorithms, such as LLE and Laplacian eigenmaps. While the bottom eigenvectors from these algorithms have been used to derive low dimensional embeddings, these solutions do not generally preserve local features such as distances or angles. Viewing these bottom eigenvectors as a partial basis for functions on the data set, we have shown how to compute a maximally angle-preserving embedding by solving an additional (but small) problem in semidefinite programming. At little extra computational cost, this framework significantly extends the utility of LLE and Laplacian eigenmaps, yielding more faithful embeddings as well as a global estimate of the data's intrinsic dimensionality.

Previous studies in dimensionality reduction have shared similar motivations as this work. An extension of Isomap was proposed to learn conformal transformations (de Silva & Tenenbaum, 2003); like Isomap, however, it relies on the estimation of geodesic distances, which can lead to spurious results when the underlying manifold is not isomorphic to a convex region of Euclidean space (Donoho & Grimes, 2002). Hessian LLE is a variant of LLE that learns isometries, or distance-preserving embeddings, with theoretical guarantees of asymptotic convergence (Donoho & Grimes, 2003). Like LLE, however, it does not yield an estimate of the underlying manifold's dimensionality. Finally, there has been work on angle-preserving linear projections (Magen, 2002). This work, however, focuses on random linear projections (Johnson & Lindenstrauss, 1984) that are not suitable for manifold learning.

Our approach in this paper builds on LLE and Laplacian eigenmaps, thus inheriting their strengths as well as their weaknesses. Obviously, when these algorithms return spurious results (due to, say, outliers or noise), the angle-preserving embeddings computed from their spectral decompositions are also likely to be of poor quality. In general, though, we have found that our approach complements and extends these earlier approaches to nonlinear dimensionality reduction at very modest extra cost.

Acknowledgments

This work was supported by the National Science Foundation under award number 0238323.

References

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

Brand, M. (2003). Charting a manifold. *Advances in Neural Information Processing Systems 15* (pp. 985–992). Cambridge, MA: MIT Press.

Burges, C. J. C. (2005). Geometric methods for feature extraction and dimensional reduction. In L. Rokach and O. Maimon (Eds.), *Data mining and knowledge discovery handbook: A complete guide for practitioners and researchers*. Kluwer Academic Publishers.

Chung, F. R. K. (1997). *Spectral graph theory*. American Mathematical Society.

de Silva, V., & Tenenbaum, J. B. (2003). Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems 15* (pp. 721–728). Cambridge, MA: MIT Press.

Donoho, D. L., & Grimes, C. E. (2002). *When does Isomap recover the natural parameterization of families of articulated images?* (Technical Report 2002-27). Department of Statistics, Stanford University.

Donoho, D. L., & Grimes, C. E. (2003). Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Arts and Sciences*, *100*, 5591–5596.

Johnson, W., & Lindenstrauss, J. (1984). Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 189–206.

Magen, A. (2002). Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications. In J. Rolim and S. Vadhan (Eds.), *Randomization and approximation techniques: Sixth international workshop, RANDOM 2002*. Springer-Verlag.

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

Saul, L. K., & Roweis, S. T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, *4*, 119–155.

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

Vandenberghe, L., & Boyd, S. P. (1996). Semidefinite programming. *SIAM Review*, *38(1)*, 49–95.

Weinberger, K. Q., & Saul, L. K. (2004). Unsupervised learning of image manifolds by semidefinite programming. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04)* (pp. 988–995). Washington D.C.