Journal of Machine Learning Research x (201x) xx-xx Submitted x/xx; Published xx/xx

Enhanced Multilevel Manifold Learning

Haw-ren Fang HRFANG@CS.UMN.EDU
Yousef Saad SAAD@CS.UMN.EDU
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA.

Editor: editor

March 24, 2012

Abstract

Two multilevel frameworks for manifold learning algorithms are discussed which are based on an affinity graph whose goal is to sketch the neighborhood of each sample point. One framework is geometric and is suitable for methods aiming to find an isometric or a conformal mapping, such as isometric feature mapping (Isomap) and semidefinite embedding (SDE). The other is algebraic and can be incorporated into methods that preserve the closeness of neighboring points, such as Locally Linear Embedding (LLE) and Laplacian eigenmaps (LE).

The multilevel coarsening technique presented in this paper can be applied to both directed and undirected graphs. It is based on the degree of dependency between vertices, a parameter that is introduced to control the speed of coarsening. In the algebraic framework, we can coarsen the problem by some form of restriction and then uncoarsen the solution by prolongation, as a standard procedure in multigrid methods. In the geometric framework, the uncoarsening method can improve the embedding quality in the sense of being isometric or conformal. The methods presented provide multiscale resolution of a manifold, and a coarse embedding is very inexpensive. An application to intrinsic dimension estimation is illustrated. Results of experiments on synthetic data and two image data sets are presented.

Keywords: manifold learning, nonlinear dimensionality reduction, multilevel methods, spectral decomposition

1. Introduction

Real world high dimensional data can often be represented as points or vectors in a much lower dimensional nonlinear manifold. Examples include face databases, continuous video images, digital voices, microarray gene expression data, and financial time series. The observed dimension is the number of pixels per image, or generally the number of numerical values per data entry, and can be characterized by far fewer features.

Since around the year 2000, a number of algorithms have been developed to 'learn' the low dimensional manifold of high dimensional data sets. Given a set of high dimensional data as vectors $x_1, \ldots, x_n \in \mathbb{R}^m$, the task is to represent them with low dimensional vectors $y_1, \ldots, y_n \in \mathbb{R}^d$ with $d \ll m$. The objective is to preserve certain properties, such as local shapes or local angles, or simply to make nearby points preserve their proximity.

Linear methods of dimensionality reduction, such as the principal component analysis (PCA) and classical multidimensional scaling (MDS), can become inadequate because the meaningful low dimensional structure of high dimensional data is often nonlinear. Therefore, considerable research effort has been devoted to the development of effective nonlinear methods to discover the underlying manifolds of given data sets.

Multilevel techniques, which aim at reducing the problem size and improving computational efficiency, have been successfully applied to various scientific problems, such as graph and hypergraph partitioning (Karypis and Kumar, 1998, 2000). However, their incorporation into manifold learning methods is currently under-explored. Inspired by their success in other applications, we presented a preliminary multilevel scheme for nonlinear dimensionality reduction (Fang et al., 2010). We found several issues that needed to be resolved. First, we used trustworthiness and continuity (Venna and Kaski, 2006) to evaluate the embedding quality. These criteria are rank-based and tend to favor methods that preserve the closeness of neighboring points, such as the uncoarsening scheme (Fang et al., 2010). Manifold learning methods for isometric or conformal mappings are likely to be under-estimated by these criteria. Second, our experiments used image data sets that were then mapped into two-dimensional space for visualization and evaluation, but the intrinsic dimension can be higher than two. Finally, our coarsening method was based on the maximum independent sets, a method which often results in too small a coarse set after one level of coarsening.

This paper describes a set of enhanced multilevel methods for nonlinear dimensionality reduction. Our discussion starts with differential geometry in order to provide a theoretical support for the techniques presented. A geometric and an algebraic multilevel frameworks are described both of which consist of three phases: data coarsening, nonlinear dimensionality reduction, and data uncoarsening. The methods presented rely on an affinity graph and so they are especially useful for affinity-graph-based manifold learning methods. To coarsen the data, we employ a graph coarsening algorithm based on the dependency between vertices. After this, we map the coarsened data at the coarsest level using one of the standard manifold learning algorithms. Finally, we recursively uncoarsen the data, level by level, using the information between the graphs of adjacent levels. The geometric framework propagates geodesic information in the coarsening phase, and improves the isometric or conformal mapping in the uncoarsening phase. In the algebraic framework, we can restrict the working matrix in the coarsening phase and prolong the solution in the uncoarsening phase in the style of multigrid methods. Figure 1 provides an illustration.

Landmark versions of manifold learning algorithms by random sampling have been proposed to reduce the problem size and therefore the computational cost, e.g., landmark Isomap (L-Isomap) (de Silva and Tenenbaum, 2003b) and landmark semidefinite embedding (L-SDE) (Weinberger et al., 2005). Another approach to reduce cost is via low-rank matrix approximation techniques (Talwalkar et al., 2008).

The multilevel methods proposed in this paper offer some advantages over the landmark approach. For example, the data points in each coarse level are selected according to the dependency of vertices in the affinity graph. It typically generates a better representation of the original data than by random sampling. Indeed, the worse case random selection is prevented by multilevel approaches. In addition, by recursive coarsening we obtain a succession of graphs on which the uncoarsening scheme is based. The succession of graphs provides useful information, such as the neighborhood of points at each level and the global

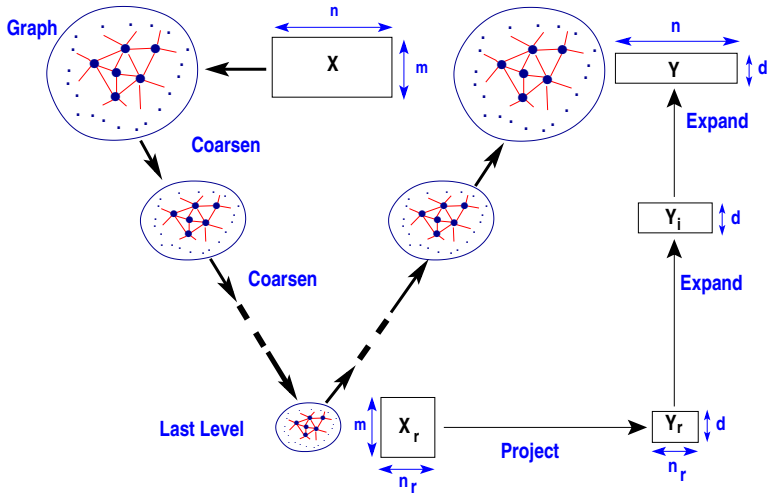


Figure 1: A sketch of the multilevel nonlinear dimensionality reduction.

geodesic information, not available with the random sampling approach. In Fang et al. (2010), we gave an example showing that bad landmarks may result in an unsatisfactory embedding which can be prevented by the multilevel approach.

In this paper we consider mainly 5 manifold learning algorithms: isometric feature mapping (Isomap) (Tenenbaum et al., 2000), locally linear embedding (LLE) (Roweis and Saul, 2000; Saul and Roweis, 2003), Laplacian eigenmaps (LE) (Belkin and Niyogi, 2001, 2003), local tangent space alignment (LTSA) (Zhang and Zha, 2004), and semidefinite embedding (SDE) (Weinberger and Saul, 2004, 2006). The conformal variants of Isomap (de Silva and Tenenbaum, 2003a,b) and SDE are also discussed.

Note that multilevel techniques are not limited to these methods. They are likely to be adaptable to other affinity-graph-based manifold learning methods, such as Hessian LLE (hLLE) (Donoho and Grimes, 2003), conformal eigenmaps (Sha and Saul, 2005), diffusion maps (Coifman and Lafon, 2006; Lafon and Lee, 2006), minimum volume embedding (MVE) (Shaw and Jebara, 2007), Riemannian manifold learning (RML) (Lin and Zha, 2008), and Greedy Procrustes (GP) and Procrustes Subspaces Alignment (PSA) (Goldberg and Ritov, 2009). Table 1 lists these manifold learning methods along with references. These algorithms can be categorized according to whether the affinity graph is directed or undirected[1], and also the characteristics of the mapping. See Section 2.2 for a discussion.

The rest of this paper is organized as follows. Section 2 reviews the background on manifold learning with different insights. Sections 3 and 4 present the geometric and the algebraic multilevel frameworks for nonlinear dimensionality reduction. Section 5 describes the quality assessment criteria for manifold embedding. Section 6 reports the results of manifold learning experiments. A conclusion is given in Section 7. Appendix A gives a unified view of the orthogonal Procrustes problem, PCA, and MDS, with applications to

1. For conformal eigenmaps, whether the affinity graph is directed or not depends on whether the basis vectors come from LE or LLE. For diffusion maps, the original formulation uses an undirected graph, which is not a restriction, though.

Table 1: Manifold Learning Algorithms.

Algorithm	Abbrev.	Affinity Graph	Mapping	Reference
isometric feature mapping	Isomap	undirected	isometric	Tenenbaum et al. (2000)
locally linear embedding	LLE	directed	proximity	Roweis and Saul (2000) and Saul and Roweis (2003)
Laplacian eigenmaps	LE	undirected	proximity	Belkin and Niyogi (2001, 2003)
conformal Isomap	C-Isomap	undirected	conformal	de Silva and Tenenbaum (2003a,b)
landmark Isomap	L-Isomap	undirected	isometric	de Silva and Tenenbaum (2003b)
Hessian LLE	hLLE	directed	geometric	Donoho and Grimes (2003)
local tangent space alignment	LTSA	directed	geometric	Zhang and Zha (2004)
semidefinite embedding	SDE	undirected	isometric	Weinberger and Saul (2004, 2006)
landmark SDE	L-SDE	undirected	isometric	Weinberger et al. (2005)
conformal eigenmaps	-	either	conformal	Sha and Saul (2005)
diffusion maps	-	either	proximity	Coifman and Lafon (2006)
minimum volume embedding	MVE	undirected	isometric	Shaw and Jebara (2007)
Riemannian manifold learning	RML	undirected	isometric	Lin and Zha (2008)
Greedy Procrustes	GP	undirected	isometric	Goldberg and Ritov (2009)
Procrustes Subspaces Alignment	PSA	undirected	isometric	Goldberg and Ritov (2009)

isometric and conformal analysis. Appendix B describes 5 manifold learning algorithms: Isomap, LLE, LE, LTSA, and SDE.

A word on notation. The column vector of ones of size k is denoted by e_k. It may be written as e, if omitting the superscript k does not cause ambiguity. An identity matrix is denoted by I, or I_k to reflect the size k-by-k. The norm $\|\cdot\|$, without a subscript, means 2-norm. We use matrices $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ and $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ $(d < m)$ to denote the high dimensional data and the corresponding low dimensional embedding, respectively.

2. Background on Manifold Learning

The underlying theory of manifold learning is closely related to differential geometry. Knowing the connection between a continuous manifold mapping and its discrete samples helps to understand the essence of the manifold learning algorithms and to design quality multilevel techniques.

2.1 Some Basics of Differential Geometry

We review some basics of differential geometry which are related to manifold learning (Do Carmo, 1976; O'Neill, 2006; Zha and Zhang, 2006).

2.1.1 Tangent Space

A function is *smooth* if it is infinitely differentiable. However, in the following discussion we often just need the function to be twice continuously differentiable. Consider a manifold $\mathcal{M} \subset \mathbb{R}^m$. A *curve* on $\mathcal{M}$ is a smooth function $\alpha : [0, 1] \to \mathcal{M}$, where without loss of generality, we have restricted the domain interval to $[0, 1]$. The *velocity vector*, also called a *tangent vector*, of the curve $\alpha(t)$ at $\alpha(t_0) \in \mathcal{M}$ is its derivative $\alpha'(t_0)$ for $t_0 \in [0, 1]$. Given

$x \in \mathcal{M}$, the *tangent space* at x, denoted by $T_x(\mathcal{M})$, is formed by all possible tangent vectors $\alpha'(0)$ with $\alpha(0) = x$.

A smooth function between two manifolds is called a *mapping*. Let $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}^m$ be a mapping such that $f(\Omega) = \mathcal{M}$. We focus on the case when Ω is open, but if Ω is a lower dimensional regular manifold, the differentiability is defined via the parameterization of Ω. The *tangent map* of $f(y)$ is a function $f_*(y, v)$ which associates a given tangent vector $v \in T_y(\Omega)$ with a tangent vector $u \in T_{f(y)}(\mathcal{M})$, such that if v is a velocity vector of a curve $\alpha : [0, 1] \to \Omega$ at $y = \alpha(0)$, then $f_*(y, v)$ is the velocity vector of the curve $f(\alpha(t))$ on $\mathcal{M}$ at $f(\alpha(0))$. In other words, the tangent map f_* takes a given tangent vector $v \in T_y(\Omega)$ and outputs the corresponding tangent vector $u \in T_{f(y)}(\mathcal{M})$ with respect to the function $f : \Omega \to \mathcal{M}$. It is clear that a tangent map transforms tangent vectors linearly. Under the assumption that Ω is open, we have

$$f_*(y, v) = J_f(y)v, \tag{1}$$

where $J_f(y)$ is the Jacobian matrix of f at y.

2.1.2 Isometry

The mapping between two manifolds $f : \Omega \to \mathcal{M}$ is called an *isometry*, if it is one-to-one and onto and for any given $v, w \in T_y(\Omega)$, we have

$$f_*(y, v)^T f_*(y, w) = v^T w. \tag{2}$$

In other words, the dot product is invariant under the tangent mapping. The condition (2) implies that the lengths of tangent vectors are preserved and vice versa, where the length of a vector v is defined as the norm induced by the dot product $\|v\| = \sqrt{v^T v}$. Riemannian geometry generalizes this by replacing the dot product in the Euclidean space by an arbitrary inner product on the tangent spaces of abstract manifolds. For the sake of concreteness, we restrict our attention to the Euclidean measure in this paper.

By (1) and (2), an isometry $f : \Omega \to \mathcal{M}$ where Ω is open implies

$$J_f(y)^T J_f(y) = I_d, \qquad y \in \Omega. \tag{3}$$

It is equivalent to the fact that the singular values of $J_f(y)$ are all one. Given a curve $\alpha : [0, 1] \to \Omega$ on Ω, the length of the curve $f(\alpha)$ on $\mathcal{M}$ is

$$\bar{L}(f(\alpha)) = \int_0^1 \left\| \frac{df(\alpha(t))}{dt} \right\| dt = \int_0^1 \|J_f(\alpha(t))\alpha'(t)\| dt = \int_0^1 \|\alpha'(t)\| dt = L(\alpha), \tag{4}$$

which is the same as the length of the curve α on Ω. Therefore, we can measure the curve length on $\mathcal{M}$ via an isometry $f : \Omega \to \mathcal{M}$.

The *geodesic distance* between two points $x_1, x_2 \in \mathcal{M}$, denoted by $\delta_{\mathcal{M}}(x_1, x_2)$, is the length of the shortest path between x_1 and x_2. If there is an isometry $f : \Omega \to \mathcal{M}$ where $\Omega \in \mathbb{R}^d$ is open and convex, then by (4),

$$\delta(f(y_1), f(y_2)) = \|y_1 - y_2\|, \tag{5}$$

for any $y_1, y_2 \in \Omega$.

According to the Theorema Egregium of Gauss, Gaussian curvature is invariant under an isometry. That is, if $f : \Omega \to \mathcal{M}$ is an isometry, then the Gaussian curvature of any point y on Ω, denoted by $K_\Omega(y)$, is the same as the Gaussian curvature of $f(y)$ on $\mathcal{M}$, denoted by $K_\mathcal{M}(f(y))$. Recall that we consider the case when Ω is open, which implies $K_\Omega(y) = 0$ for all $y \in \Omega$. Hence if $K_\mathcal{M}(x) \neq 0$ for some $x \in \mathcal{M}$, there exists no isometry $f : \Omega \to \mathcal{M}$ with Ω open. For example, we cannot find an isometry that maps an open set in $\mathbb{R}^2$ to a sub-manifold of a sphere in $\mathbb{R}^3$, since a sphere has a positive Gaussian curvature at every point on it.

2.1.3 Conformal Mapping

For manifolds for which an isometry is too much to ask, we consider weaker alternatives, for example, to preserve angles. Formally, a mapping $f : \Omega \to \mathcal{M}$ between two manifolds $\Omega \subset \mathbb{R}^d$ and $\mathcal{M} \subset \mathbb{R}^m$ is *conformal*, if there is a positive smooth function $c : \Omega \to \mathbb{R}$ such that

$$\|f_*(y, v)\| = c(y)\|v\| \tag{6}$$

for all tangent vectors $v \in T_y(\Omega)$ at each $y \in \Omega$ (O'Neill, 2006). An equivalent definition (Do Carmo, 1976) is that for $v, w \in T_p(\Omega)$,

$$f_*(y, v)^T f_*(y, w) = c(y)^2 v^T w. \tag{7}$$

The equivalence between (6) and (7) can be seen from the norm which is induced by the dot product of the tangent spaces of both Ω and $\mathcal{M}$. Note that every tangent space is a vector space and every tangent map is a linear function in terms of the tangent vectors. Assuming that f is one-to-one and onto, $c(y) \equiv 1$ implies an isometry f.

A conformal mapping preserves angles. A well-known example is that in complex analysis, every analytic function is conformal at any point where it has a nonzero derivative (Bak and Newman, 1982). Assume that the domain Ω of a conformal mapping f is open. By (1) and (7),

$$J_f(y)^T J_f(y) = c(y)^2 I_d, \qquad y \in \Omega. \tag{8}$$

In other words, a conformal mapping $f : \Omega \to \mathcal{M}$ has the Jacobian $J_f(y)$ consisting of orthonormal columns subject to a nonzero scale factor $c(y)$ for all $y \in \Omega$. Let

$$\widetilde{f}(y) = \int \frac{1}{c(y)} J_f(y) dy. \tag{9}$$

By the gradient theorem and (8), we have

$$J_{\widetilde{f}}(y) = \frac{1}{c(y)} J_f(y) \quad \Rightarrow \quad J_{\widetilde{f}}(y)^T J_{\widetilde{f}}(y) = I_d. \tag{10}$$

Hence $\widetilde{f}$ is an isometry. In this respect the manifold $\widetilde{\mathcal{M}} = \widetilde{f}(\Omega)$ is more tractable than $\mathcal{M} = f(\Omega)$, since $\widetilde{\mathcal{M}}$ is isometric to an open set $\Omega \in \mathbb{R}^d$.

2.1.4 Regular Mapping

In manifold learning, it is a natural desire to preserve the *intrinsic dimension*. Hence we consider *regular mappings*. A mapping $f : \Omega \to \mathcal{M}$ between two manifolds $\Omega \subset \mathbb{R}^d$ and $\mathcal{M} \subset \mathbb{R}^m$ is *regular* provided that at every point $y \in \Omega$, the tangent map $f_*(y, v)$ is one-to-one in terms of v (O'Neill, 2006). It is equivalent to the statement that $f_*(y, v) = 0$ implies $v = 0$ for $v \in T_y(\Omega)$, since a tangent map is a linear transformation. Assuming that $\Omega \in \mathbb{R}^d$ is open, a mapping $f \in \Omega \to \mathcal{M}$ is regular if the Jacobian $J_f(y)$ has full column rank for $y \in \Omega$ and vice versa. Here d is called the intrinsic dimension of $\mathcal{M}$, which is independent of the mapping f and its open domain Ω.

Consider the one-to-one and onto mappings $f : \Omega \to \mathcal{M}$. An isometry guarantees a conformal mapping, which implies a regular mapping.

2.2 Learning with Discrete Samples

Given a manifold $\mathcal{M} \subset \mathbb{R}^m$, we may wish to find a function $g : \mathcal{M} \to \Omega \subset \mathbb{R}^d$ which maps $\mathcal{M}$ to another manifold $g(\mathcal{M}) = \Omega$ in a lower dimensional space $\mathbb{R}^d$ ($d < m$). In practice, we often have discrete and possibly noisy sampled data points $x_1, \ldots, x_n \in \mathbb{R}^m$ of $\mathcal{M}$, and the objective is to find a low dimensional embedding $y_1, \ldots, y_n \in \mathbb{R}^d$. The goal of the mapping is to preserve certain 'local' properties, for which it is typical to employ an affinity graph.

2.2.1 Affinity Graph

We denoted by $G = (V, E)$ an affinity graph of data $x_1, \ldots, x_n \in \mathbb{R}^d$, where the vertex set $V = \{1, \ldots, n\}$ consists of data indices, and $(i, j) \in E$ if vertex j is a neighbor of vertex i. There are two ways to define the neighborhood based on distances between $x_1, \ldots, x_n$.

1. ϵ-neighborhood: vertex j is a neighbor of vertex i if $\|x_i - x_j\| < \epsilon$.

2. k-nearest-neighbor: vertex j is a neighbor of vertex i if x_j is one of the k nearest neighbors of x_i.

The first definition yields neighborhoods that are reciprocal, in that the corresponding graph is undirected, while the second does not. In practice, the k-nearest-neighbor metric is also popular, since the ϵ-neighborhood often yields a disconnected graph.

There are two types of affinity graphs used in a manifold learning algorithm. One is directed, for example, those used in LLE, hLLE, LTSA, and RML. The other is undirected, for example, those used in Isomap, LE, and SDE. See Table 1 for a summary. If the neighborhood is not reciprocal but the algorithm requires an undirected affinity graph, then we need to perform the symmetrization, by either removing (i, j) from E for $(j, i) \notin E$, or adding (j, i) to E for $(i, j) \in E$. The latter method is used in our experiments.

In what follows the graphs are considered directed. We also assume that there is no self-edge, i.e., $(i, i) \notin E$ for all $i \in V$. An affinity graph $G = (V, E)$ is canonically associated with a sequence of neighborhood sets $\mathcal{N}_1, \ldots, \mathcal{N}_n$, such that $j \in \mathcal{N}_i$ if and only if $(i, j) \in E$ for $i, j = 1, \ldots, n$. That is, $\mathcal{N}_i$ contains the neighbors of vertex i.

2.2.2 Isometric Embedding

An isometric embedding method assumes the existence of an isometry $f : \Omega \subset \mathbb{R}^d \to \mathcal{M} \subset \mathbb{R}^m$ described in Section 2.1.2. Given input $x_1, \ldots, x_n \in \mathcal{M}$ and neighborhood sets $\mathcal{N}_1, \ldots, \mathcal{N}_n$, the goal is to find an embedding $y_1, \ldots, y_n \in \Omega$ such that 'local' distances are preserved, i.e.,

$$\|y_i - y_j\| \approx \|x_i - x_j\|, \qquad i = 1, \ldots, n, \quad j \in \mathcal{N}_i, \tag{11}$$

where $\mathcal{N}_i$ contains indices of neighbors of x_i. The objective (11) can be seen from the Taylor series applied to $f(y)$

$$f(y + \Delta y) = f(y) + J_f(y)\Delta y + O(\|\Delta y\|^2), \tag{12}$$

which implies

$$\|f(y + \Delta y) - f(y)\| \approx \|J_f(y)\Delta y\| = \|\Delta y\|, \tag{13}$$

where the property of isometry $J_f(y)^T J_f(y) = I_d$ in (3) is used for the last equation. Substituting x_i for $f(y)$, x_j for $f(y + \Delta y)$, and $y_j - y_i$ for Δy in (13), we obtain (11).

A typical example of isometric embedding is SDE, which aims to preserve the local distances and at the same time maximize $\sum_{i,j} \|y_i - y_j\|^2$. It requires solving a semidefinite programming problem (Vandenberghe and Boyd, 1996). An outline of the process is given in Section B.5.

Isomap further assumes the convexity of Ω and exploits (5). More precisely, two vertices i, j are nearby if $(i, j) \in E$ in the affinity graph $G = (V, E)$. The distance between two nearby vertices i, j is defined as $\|x_i - x_j\|$. The length of the shortest path from vertex i to vertex j is used to approximate the geodesic distance $\delta(x_i, x_j)$, which equals $\|y_i - y_j\|$ according to (5). With all approximate $\|y_i - y_j\|$ for $i, j = 1, \ldots, n$ available, the MDS is applied to find the embedding $y_1, \ldots, y_n$. See, for example, Appendix B.1 for additional details.

Both Isomap and SDE are spectral methods, since they eventually form a symmetric matrix, called a *kernel*, and compute its eigenvectors for embedding. With the strong property of preserving the isometry, the number of significant eigenvalues in practice is a good indicator of the intrinsic dimensions (Saul et al., 2006; Tenenbaum et al., 2000; Weinberger and Saul, 2006). In Section 6.4, we will demonstrate an application of the multilevel technique to intrinsic dimension estimation with a coarse kernel.

There are other manifold learning methods for an isometric embedding, such as hLLE, RML, GP and PSA. These methods rely on the tangent spaces of the sample points and more or less use the techniques summarized in Appendix A.

2.2.3 Conformal Embedding

For manifolds with non-negotiable Gaussian curvatures, an alternative to use conformal mapping described in Section 2.1.3. The discrete form can be written as

$$\|y_i - y_j\| \approx \frac{1}{c_i}\|x_i - x_j\|, \qquad i = 1, \ldots, n, \quad j \in \mathcal{N}_i, \tag{14}$$

where $c_1, \ldots, c_n \in \mathbb{R}$ are positive constants, and $\mathcal{N}_i$ contains the indices of neighbors of vertex i. This can be seen from a similar discussion leading to (11). The difference is that here we use (8) instead of (3), and c_i in (14) plays the role of $c(y_i)$ defined in (8).

A manifold learning method with the objective (14) is conformal eigenmaps (Sha and Saul, 2005), which solves a semidefinite program using basis vectors from LE or LLE.

Another example is a variant of Isomap, called conformal Isomap (de Silva and Tenenbaum, 2003a,b). It is abbreviated as C-Isomap in this paper. The only algorithmic change is that in Isomap, the distance between two nearby vertices i, j is the local distance $\|x_i - x_j\|$, and in C-Isomap it is scaled by $1/\sqrt{M(i)M(j)}$, where $M(i)$ the mean distance from x_i to its neighbors. To be precise, the 'distance' between each pair of neighboring vertices i, j in C-Isomap is

$$\|x_i - x_j\|/\sqrt{M(i)M(j)}, \qquad M(i) = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\|x_i - x_j\|. \tag{15}$$

The underlying assumption is that the sample points are uniformly distributed in the parameter space Ω. Therefore, the density of points in the neighborhood of x_i is proportional to $1/c_i$ in (14). Hence $M(i)$ is a good estimate of c_i. Instead of $M(i)$, C-Isomap uses the scale factor $\sqrt{M(i)M(j)}$ for the symmetry of the scaled distances.

Note that the scaling can also be applied to isometric embedding methods which relies on the local distances, such as SDE, MVE, RML, GA, and PSA, yielding conformal versions of these algorithms. In addition to the manifold learning methods in the literature, we also used this conformal version of SDE, abbreviated as C-SDE, in our experiments. Some results are reported in Section 6.

It is worth noting that while the input $x_1, \ldots, x_n$ are points sampled from the manifold $\mathcal{M}$, C-Isomap actually considers the transformed manifold $\widetilde{f}(\Omega) = \widetilde{\mathcal{M}}$ which is isometric to Ω, where $\widetilde{f}$ is defined in (9). Here we give a new interpretation to support C-Isomap.

2.2.4 Proximity Preserving Embedding

This type of method also studies a manifold mapping $f : \Omega \to \mathcal{M}$ with an open domain $\Omega \in \mathbb{R}^d$, where $\mathcal{M} = f(\Omega) \subset \mathbb{R}^m$. However, it makes no assumption on f being conformal or an isometry. Suppose we are given two close points $y_1, y_2 \in \Omega$ and their mapped points $x_1 = f(y_1)$ and $x_2 = f(y_2)$. By the Taylor series (12), we still have $\Delta x = J_f(y)\Delta y + O(\|\Delta y\|^2)$, where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$. Therefore, small $\|\Delta y\|$ implies small $\|\Delta x\|$.

On the other hand, if the mapping f is regular, then $J_f(y)$ has full column rank and therefore $(J_f(y)^+)^T J_f(y) = I_d$, where $J_f(y)^+$ is the Moore-Penrose pseudo-inverse of $J_f(y)$. Hence $J_f(y)^+\Delta x = \Delta y + O(\|\Delta y\|^2)$. We also have the property that small $\|\Delta x\|$ implies small $\|\Delta y\|$.

In conclusion, assuming that f is regular and one-to-one, for two close points $x_1, x_2 \in \mathcal{M}$, the corresponding points $y_1, y_2 \in \Omega$, mapped by f^{-1}, are close to each other, too. Hence the goal is to make nearby points remain nearby. A typical example is LE, and another example is LLE.

There are manifold learning algorithms which utilizes local tangent spaces but the final objective is weaker than conformal mapping, e.g., LTSA and hLLE. Because local tangent spaces contain geometric information, these methods preserve more than closeness of neighboring points. Hence we mark them as 'geometric' in Table 1.

3. Geometric Multilevel Nonlinear Dimensionality Reduction

We present two multilevel frameworks for nonlinear dimensionality reduction in this section and in the next section. Both of them consist of three phases: data coarsening, nonlinear dimension reduction, and data uncoarsening. In a nutshell, a few levels of coarsening are performed leading to a sequence of smaller and smaller graphs. The analysis of the data is done at the coarsest level using a standard manifold learning method, such as Isomap or LLE. Then an 'uncoarsening' step of this low dimensional data is performed level by level backing up to the highest level.

A key property of the framework presented in this section is that, it propagates geodesic information in the coarsening phase. In the uncoarsening phase, the geodesic information will be utilized to improve the isometric or conformal embedding. Hence we call this framework geometric. It is particularly useful to incorporate the manifold learning algorithms which aim to preserve the isometric or conformal information, e.g., Isomap, C-Isomap, SDE, and C-SDE. We call the resulting methods multilevel Isomap, multilevel C-Isomap, multilevel SDE, and multilevel C-SDE.

3.1 The Coarsening Phase

As it is assumed in Section 2.2.1, the graphs are considered directed and free of any self-edges. This property will be preserved in a coarsened graph in the geometric framework. Note that since the geodesic information is propagated, it is common to use an undirected affinity graph. Nevertheless, part of the techniques presented here will be used in another multilevel framework given in Section 4, where directed graphs are also popular.

Coarsening a graph $G^{(l)} = (V^{(l)}, E^{(l)})$ means finding a 'coarse' approximation $G^{(l+1)} = (V^{(l+1)}, E^{(l+1)})$ that represents $G^{(l)} = (V^{(l)}, E^{(l)})$, where $|V^{(l+1)}| < |V^{(l)}|$. By recursively coarsening for $l = 0, \ldots, r-1$, we obtain a succession of smaller graphs $G^{(1)}, \ldots, G^{(r)}$ which approximate the original graph $G^{(0)}$. To simplify the notation, we use $G = (V, E)$ and $\widehat{G} = (\widehat{V}, \widehat{E})$ to denote, respectively, the fine graph $G^{(l)} = (V^{(l)}, E^{(l)})$ and the coarse graph $G^{(l+1)} = (V^{(l+1)}, E^{(l+1)})$ of two successive levels.

3.1.1 Vertex–based Coarsening

In a previous paper (Fang et al., 2010), we used maximum independent sets for graph coarsening. However, this requires the graph to be undirected and it often results in a rapid coarsening, which is undesirable. This paper considers another strategy based on the dependency between vertices. We start with a definition.

Definition 1 *Given a graph $G = (V, E)$, we say that $\widehat{V} \subset V$ is a degree p representation of V, if every vertex $i \in V \setminus \widehat{V}$ has at least p neighbors in $\widehat{V}$, i.e., $|\{(i, j) \in E : j \in \widehat{V}\}| \geq p$ for $i \in V \setminus \widehat{V}$. Furthermore, $\widehat{V}$ is called a minimal degree p representation of V, if any proper subset of $\widehat{V}$ is not a degree p representation of V. In addition, the complement $V \setminus \widehat{V}$, is said to be self-repellent if if for all $i, j \in V \setminus \widehat{V}$, $(i, j) \notin E$.*

An example of a coarse representation with $p = 2$ is illustrated in Figure 2. Conceptually, $(i, j) \in E$ means that vertex i depends on vertex j. If every vertex $i \in V \setminus \widehat{V}$ has at least p

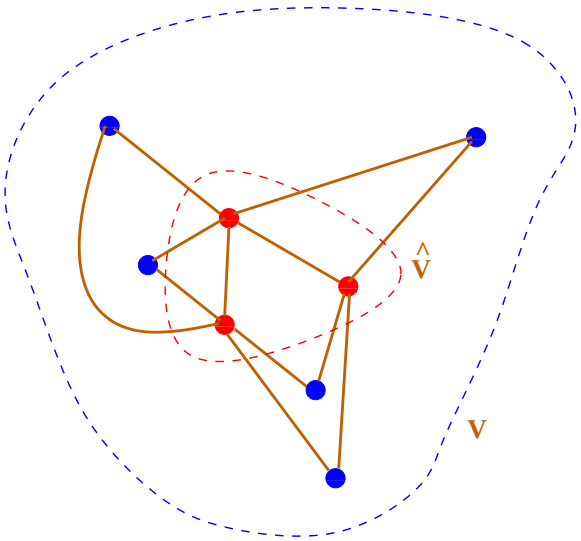


Figure 2: Illustration of a coarse representation of V with $p = 2$; vertices of $\widehat{V}$ are in red.

neighbors in $\widehat{V}$ it means that $V \setminus \widehat{V}$ depends on $\widehat{V}$ to degree p, and therefore $\widehat{V}$ can be used as a coarse representation of V. The parameter p controls the coarseness of $\widehat{V}$.

The rationale for having a self-repellent complement $V \setminus \widehat{V}$ is that when we use $\widehat{V}$ as a coarse representation of V and exclude $V \setminus \widehat{V}$, the vertices dropped do not cluster since they are adjacent to each other in the graph. Note however that having a self-repellent complement $V \setminus \widehat{V}$ is a strong constraint. It makes the coarsening slow and insensitive to the degree p, as long as p is small. An example will be illustrated at the end of Section 6.1.

An interesting property is that when the graph $G = (V, E)$ is undirected, a self-repellent complement $V \setminus \widehat{V}$ is an independent set, and hence the representation $\widehat{V}$ is a vertex cover of G. Moreover, if the graph $G = (V, E)$ is undirected and connected and the degree of dependency is $p = 1$, then a self-repellent complement $V \setminus \widehat{V}$ is an independent set and vice versa.

A minimal representation $\widehat{V}$ of V to degree p can be found by a greedy algorithm. Initialize $\widehat{V}$ as V, visit $k \in \widehat{V}$ in a graph traversal, and remove this k from $\widehat{V}$ if after $\widehat{V}$ remains a degree p representation after the removal. The self-repellent complement constraint can be incorporated. Algorithm 1 gives the pseudo-code.

Consider the main loop of Algorithm 1. If repel = **true**, it is required to find all edges $(i, k) \in E$ efficiently for a given vertex $k \in V$, i.e., it is required to find the parents i of k. We can store the graph $G = (V, E)$ as a boolean sparse matrix $B \in \{0, 1\}^{n \times n}$ in some sparse storage format, e.g., the Compressed Sparse Column (CSC) format (Saad, 1994), where $B(i, j) = 1$ if $(i, j) \in E$ and otherwise $B(i, j) = 0$. On the other hand, if repel = **false**, we need to visit not only all the edges (i, k) but also the edges (k, i) for a given vertex k. This would require the sparsity patterns of both B and its transpose, but this is unnecessary if the graph is undirected.

Algorithm 1 also determines $\widehat{E}$, the edge set of $\widehat{G} = (\widehat{V}, \widehat{E})$. The rule is that for $i, j \in \widehat{V}$ with $i \neq j$, $(i, j) \in \widehat{E}$ if $(i, j) \in E$ or there is a vertex $k \in V \setminus \widehat{V}$ such that $(i, k), (k, j) \in E$. It is clear that if $G = (V, E)$ is undirected and connected, then $\widehat{G} = (\widehat{V}, \widehat{E})$ remains undirected

function $\widehat{G}$=CoarseGraph(G, p, repel)
  input: Graph $G = (V, E)$ with $V = \{1, \ldots, n\}$, and the degree p.
  output: Coarsened $\widehat{G} = (\widehat{V}, \widehat{E})$ with $\widehat{V}$ a minimal representation of V to degree p.
  remark: If repel = **true**, then $V \setminus \widehat{V}$ is a self-repellent complement of $\widehat{V}$.
  $\widehat{V} \leftarrow V$
  $\widehat{U} \leftarrow \emptyset$ ▷ complement set of $\widehat{V}$
  for all $k \in V$ **do**
    $n(k) \leftarrow |\{j : (k, j) \in E\}|$ ▷ number of neighbors of vertex k in $\widehat{V}$
  end for
  if repel = **true then**
    for all $k \in V$ **do**
      if $n(k) \geq p$, and $\forall (i, k) \in E$ or $(k, i) \in E$, $i \notin \widehat{U}$ **then**
        $\widehat{V} \leftarrow \widehat{V} \setminus \{k\}$
        $\widehat{U} \leftarrow \widehat{U} \cup \{k\}$
      end if
    end for
  else
    for all $k \in V$ **do**
      if $n(k) \geq p$, and $\forall (i, k) \in E$ with $i \in \widehat{U}$, $n(i) > p$ **then**
        $\widehat{V} \leftarrow \widehat{V} \setminus \{k\}$
        $\widehat{U} \leftarrow \widehat{U} \cup \{k\}$
        for all $(i, k) \in E$ with $i \in \widehat{U}$ **do**
          $n(i) \leftarrow n(i) - 1$
        end for
      end if
    end for
  end if
  $\widehat{E} \leftarrow \{(i, j) : (i, j) \in E \wedge i, j \in \widehat{V}\}$ ▷ edge set of $\widehat{G}$
  for all $i, j \in \widehat{V}$ and $i \neq j$ **do**
    if $\exists k \in \widehat{U}$ such that $(i, k), (k, j) \in E$ **then**
      $\widehat{E} \leftarrow \widehat{E} \cup \{(i, j)\}$
    end if
  end for
end function

Algorithm 1: Graph coarsening according to the dependency between vertices.

and connected. In some manifold learning algorithms, it is important that the affinity graph be undirected and connected. For example, a connected graph is required for Isomap so that the shortest path between each pair of vertices is defined. For SDE, the objective function is unbounded if the affinity graph is not connected.

3.1.2 Propagation of Geodesic Information

For algorithms aiming for an isometric embedding, such as Isomap and SDE, we propagate the geodesic information in the coarsening phase as follows. For each edge $(i, j) \in E$ of the

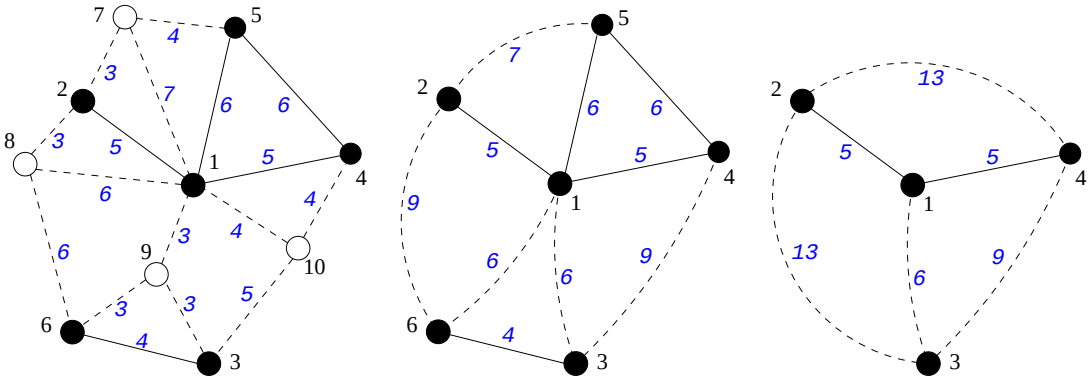
fine graph $G = (V, E)$, we use $\delta(i, j)$ to denote the distance between vertex i and vertex j. For two vertices i, j of the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$, we define the distance $\widehat{\delta}(i, j)$ by

$$\widehat{\delta}(i,j) = \min\{\delta(i,j), \min_{(i,k),(k,j)\in E} \delta(i,k) + \delta(k,j)\}, \tag{16}$$

where we let $\delta(i, j) = \infty$ for $(i, j) \neq E$ for notational convenience. If $G = (V, E)$ is the affinity graph $G^{(0)}$ at the top level, then we set $\delta(i, j) = \|x_i - x_j\|$ for $(i, j) \in E$, in which case (16) implies $\widehat{\delta}(i, j) = \delta(i, j)$ for $(i, j) \in \widehat{E} \cap E$.

Assuming that the graph is undirected, the distance function defined by the recursion (16) indeed approximates geodesic distances across levels. Note that the geodesic distances may not satisfy triangle inequality. As discussed in Section 2.1.2, if the mapping $x = f(y)$ to discover is isometric and has an open convex domain, then the Euclidean distance $\|y_i - y_j\|$ between y_i and y_j is the geodesic distance between x_i and x_j. In manifold learning, the distances $\|x_i - x_j\|$ for $(i, j) \in E$ are used, since $\|x_i - x_j\|$ approximates the geodesic distance between x_i and x_j while x_i and x_j are close to each other. In the proposed multilevel framework, the geodesic distances approximated from iteratively applying (16) are better than the Euclidean distances when the goal is to preserve isometry.

The propagation (16) does not rely on whether or not the graphs are undirected. Hence it is possible to use directed affinity graphs, in which case the computed distances can be asymmetric. In practice, if we apply a dimensionality reduction algorithm based on local distances at the bottom level, then we use an undirected graph $G^{(0)}$ at the top level and propagate distances by (16). For every coarse graph $G^{(l)} = (V^{(l)}, E^{(l)})$ and for each $(i, j) \in E^{(l)}$, the length of the shortest path from vertex i to vertex j in $G^{(l)}$ is the same as the length of the shortest path from vertex i to vertex j in $G^{(0)}$. This is an ideal property to have for Isomap.



(a) $G^{(0)} = (V^{(0)}, E^{(0)})$ (b) $G^{(1)} = (V^{(1)}, E^{(1)})$ (c) $G^{(2)} = (V^{(2)}, E^{(2)})$

Figure 3: Illustration of multilevel graph coarsening.

An example of multilevel graph coarsening is illustrated in Figure 3. We use a symmetrized kNN graph $G^{(0)} = (V^{(0)}, E^{(0)})$ with $k = 3$, shown in Figure 3(a), where the vertex indices are in black and edge distances are in blue italic. We apply Algorithm 1 to coarsen $G^{(0)} = (V^{(0)}, E^{(0)})$ with the degree of dependency $p = 3$, where the vertices are visited in

this order $(7), 5, 1, 2, (8), 6, (9), 3, (10), 4$, with the vertices in $V^{(0)} \setminus V^{(1)}$ are in parentheses. In Figure 3(a), we use solid circles for vertices in $V^{(1)}$, hollow circles for vertices in $V^{(0)} \setminus V^{(1)}$, solid lines for edges between vertices in $V^{(1)}$, and dashed lines for the other edges. The resulting coarse graph $G^{(1)} = (V^{(1)}, E^{(1)})$ is shown in Figure 3(b), where we have applied (16) for lengths of newly added edges, which are marked by dashed arcs. We apply Algorithm 1 again to coarsen $G^{(1)} = (V^{(1)}, E^{(1)})$ with the degree of dependency $p = 3$, where the vertices are visited in the order $(5), 1, 2, (6), 3, 4$, with vertices $5, 6$ in parentheses since they are in $V^{(1)} \setminus V^{(2)}$. The resulting graph $G^{(2)} = (V^{(2)}, E^{(2)})$ is shown in Figure 3(c).

3.1.3 Propagation of Conformal Information

Now consider C-Isomap, which assumes the given high dimensional points $x_1, \ldots, x_n$ are sampled from a manifold $\mathcal{M} \in \mathbb{R}^m$ which is the image of a conformal manifold mapping $f : \Omega \to \mathcal{M}$ with an open convex set $\Omega \in \mathbb{R}^d$. Instead of f, C-Isomap studies the transformed mapping $\widetilde{f}$ defined in (9) as an isometry. Assuming the data points are sampled uniformly in the parameter space Ω, C-Isomap uses the scaled local Euclidean distances (15) as the approximate local Euclidean distances on $\widetilde{\mathcal{M}} = \widetilde{f}(\Omega)$ defined in Section 2.1.3.

Incorporating C-Isomap into the multilevel framework, we use (15) as the distance measure for the affinity graph $G^{(0)}$ at the top level, and propagate distances by (16), approximating the geodesic information on $\widetilde{\mathcal{M}}$.

3.2 The Dimension Reduction Phase

In manifold learning, we apply a dimensionality reduction algorithm to a given data set $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$ and obtain the low dimensional embedding $Y = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^{d \times n}$ $(d < m)$, such that Y preserves certain neighborhood information of X.

In the geometric multilevel framework, the dimensionality reduction method is applied to the data set $X^{(r)} \in \mathbb{R}^{m \times |V^{(r)}|}$ of the bottom, i.e., the coarsest, level and results in a coarse embedding $Y^{(r)} \in \mathbb{R}^{d \times |V^{(r)}|}$ $(d < m)$. The dimensionality reduction methods considered for this task, such as Isomap and SDE, are based on an affinity graph and local distances between neighboring vertices. Instead of building a kNN graph at the bottom level, we use the graph from the coarsening phase. We also use the approximate geodesic distances from the propagation for the local distances in the dimensionality reduction algorithm.

3.3 The Uncoarsening Phase

The objective of this phase is to obtain a reduced representation $Y \in \mathbb{R}^{d \times n}$ of the data $X \in \mathbb{R}^{m \times n}$ at the finest level, starting from the reduced representation $Y^{(r)} \in \mathbb{R}^{d \times |V^{(r)}|}$ of data $X^{(r)} \in \mathbb{R}^{m \times |V^{(r)}|}$ at the coarsest level. Note that $Y = Y^{(0)}$ and $n = |V^{(0)}|$,

We recursively uncoarsen the data, level by level, in the low dimensional space as follows. We denote by $G = (V, E)$ and $\widehat{G} = (\widehat{V}, \widehat{E})$ the two affinity graphs of adjacent levels l and $(l+1)$, respectively. For each level $l = r-1, r-2, \ldots, 0$, we recursively build the reduced representation $Y = [y_i]_{i \in V}$ of level l from $\widehat{Y} = [y_i]_{i \in \widehat{V}}$ of level $(l+1)$. Since $y_i \in \mathbb{R}^d$ is known for $i \in \widehat{V}$, the goal is to determine $y_i \in \mathbb{R}^d$ for $i \in V \setminus \widehat{V}$.

At first glance, the problem is similar to that of out-of-sample extension (Bengio et al., 2004). However, we should utilize the geodesic information from the coarsening phase instead of the high dimensional coordinates. We consider the following approaches.

3.3.1 Greedy Isometric Refinement

Consider the manifold learning methods for an isometric embedding, e.g., Isomap and SDE. If such a method is used at the bottom level to compute the coarse embedding, then the objective for an isometric embedding should be retained in the uncoarsening phase.

Assume for now that $\widehat{V}$ has a self-repellent complement, and hence for $i, j \in V \setminus \widehat{V}$, $(i, j) \notin E$. A discussion will follow for the case without a self-repellent complement. Let

$$\widehat{\mathcal{N}}_i = \{j \in \widehat{V} : (i, j) \in E\} \quad \text{and} \quad \widetilde{\mathcal{N}}_i = \widehat{\mathcal{N}}_i \cup \{i\} \tag{17}$$

for each vertex $i \in V \setminus \widehat{V}$. That is, $\widehat{\mathcal{N}}_i$ is the set of neighbors of vertex i, and $\widetilde{\mathcal{N}}_i$ in addition includes vertex i.

We perform local Isomap on vertices in $\widetilde{\mathcal{N}}_i$ with the subgraph of $G = (V, E)$ induced by $\widetilde{\mathcal{N}}_i$, and obtain an embedding $\widetilde{Z}_i = [z_j^{(i)}]_{j \in \widetilde{\mathcal{N}}_i} \in \mathbb{R}^{d \times |\widetilde{\mathcal{N}}_i|}$ in the lower dimensional space. Note that we use the geodesic information propagated in the coarsening phase instead of computing the Euclidean distances in the high dimensional space. This is an ideal local embedding for each $i \in V \setminus \widehat{V}$. See Appendix A.4 for a discussion.

Let $\widetilde{Y}_i = [y_j]_{j \in \widetilde{\mathcal{N}}_i}$, in which the only column to determine is y_i for $i \in V \setminus \widehat{V}$. Ideally there is an orthogonal matrix $Q_i \in \mathbb{R}^{d \times d}$ and a translation vector γ_i such that $\widetilde{Y}_i = Q_i \widetilde{Z}_i + \gamma_i e_{n_i}^T$. Due to the approximation errors and potential noise in the data, it is unrealistic to expect the existence of such a pair of Q_i and γ_i to satisfy the equation. Therefore, we consider

$$\begin{cases} \underset{Q_i, \gamma_i}{\text{minimize}} & \|\widehat{Y}_i - Q_i \widehat{Z}_i - \gamma_i e_{n_i}^T\|_F^2 \\ \text{subject to} & Q_i^T Q_i = I_d,\ Q_i \in \mathbb{R}^{d \times d},\ \gamma_i \in \mathbb{R}^d, \end{cases} \tag{18}$$

where $\widehat{Y}_i = [y_j]_{j \in \widehat{\mathcal{N}}_i}$ and $\widehat{Z}_i = [z_j^{(i)}]_{j \in \widehat{\mathcal{N}}_i}$, from removing y_i in $\widetilde{Y}_i$ and $z_i^{(i)}$ in $\widetilde{Z}_i$, respectively. Problem (18) is related to the orthogonal Procrustes analysis and it can be solved optimally. See Appendix A.1 for details. Finally, we use the minimizer Q_i and γ_i of (18) to compute $y_i = Q_i z_i^{(i)} + \gamma_i$. The procedure is repeated for all $i \in V \setminus \widehat{V}$ to complete one level of uncoarsening. The embedding points y_i can be computed in parallel for $i \in V \setminus \widehat{V}$.

So far we have assumed that $\widehat{V}$ has a self-repellent complement, and hence there is no edge between vertices in $V \setminus \widehat{V}$. If $V \setminus \widehat{V}$ is not self-repellent, then the above uncoarsening scheme still works. On the other hand, the embedding quality can be improved as follows. We keep a set of embedded vertices, denoted by $\bar{V}$, and replace the definition of $\mathcal{N}_i$ in (17) by $\widehat{\mathcal{N}}_i = \{j \in \bar{V} : (i, j) \in E\}$. The set $\bar{V}$ is initialized as $\widehat{V}$, and has vertex i added whenever y_i is determined for $i \in V \setminus \widehat{V}$. The points y_i for $i \in V \setminus \widehat{V}$ can be embedded in random order, or in the sequence that vertex i has the most embedded neighbors at the point right before y_i is embedded. The trade-off of this approach is that we lose the parallel nature of the algorithm.

To summarize, we use y_i for $i \in \widehat{V}$ as anchors to obtain y_j for $j \in V \setminus \widehat{V}$ with the goal to preserve local shapes. This idea has been exploited in RML (Lin and Zha, 2008) and

GP (Goldberg and Ritov, 2009). The key difference is that instead of local PCA, we use local Isomap to obtain the low dimensional local coordinates. Compared to the local PCA approach, the uncoarsening presented here is economical since it does not have to compute distances in the high dimensional space. In addition, we utilize the approximate geodesic information which is better than the Euclidean distances in manifold learning.

3.3.2 Alternating Isometric Refining

Greedy isometric refining is simple and efficient. On the other hand, since $G = (V, E)$ contains more information than $\widehat{G} = (\widehat{V}, \widehat{E})$, the embedding in the coarse level $\widehat{Y} = [y_i]_{i \in \widehat{V}_i}$ may no longer be optimal with respect to $G = (V, E)$. Therefore, we may modify the existing y_i for $i \in \widehat{E}$ in order to improve the embedding quality.

Let $\mathcal{N}_i = \{j \in V : (i, j) \in E\} \cup \{i\}$ for each vertex $i \in V$. We perform local Isomap on vertices in $\mathcal{N}_i$ with the subgraph of $G = (V, E)$ induced by $\mathcal{N}_i$, and obtain a local embedding $Z_i = [z_j^{(i)}]_{j \in \mathcal{N}_i}$ for $i \in V$. If $V \setminus \widehat{V}$ is self-repellent, then Z_i obtained in this step is the same as $\widetilde{Z}_i$ in Section 3.3.1 for $i \in V \setminus \widehat{V}$. We consider the program

$$\begin{cases} \underset{Q_i, \gamma_i}{\text{minimize}} & \sum_{i \in V} \|Y_i - Q_i Z_i - \gamma_i e_{n_i}^T\|_F^2 \\ \text{subject to} & Q_i^T Q_i = I_d,\ Q_i \in \mathbb{R}^{d \times d},\ \gamma_i \in \mathbb{R}^d,\ \forall i \in V, \end{cases} \tag{19}$$

where $Y_i = [y_j]_{j \in \mathcal{N}_i}$ contains the global coordinates and $Z_i = [z_j^{(i)}]_{j \in \mathcal{N}_i}$ contains the local coordinates for all $i \in V$. As shown in Appendix A.1, when the minimum of $\|Y_i - Q_i Z_i - \gamma_i e_{n_i}^T\|_F^2$ is reached, we have $\gamma_i = \frac{1}{n_i}(Y_i - Q_i Z_i) e_{n_i}$, where $n_i = |\mathcal{N}_i|$. Therefore, the program (19) is equivalent to

$$\begin{cases} \underset{Q_i}{\text{minimize}} & \sum_{i \in V} \|(Y_i - Q_i Z_i) J_{n_i}\|_F^2 \\ \text{subject to} & Q_i^T Q_i = I_d,\ Q_i \in \mathbb{R}^{d \times d},\ \forall i \in V, \end{cases} \tag{20}$$

where $J_{n_i} = I_{n_i} - \frac{1}{n_i} e_{n_i} e_{n_i}^T$ is the centering matrix.

Let B_i be the boolean selection matrix such that $Y_i = Y B_i$ for each $i \in V$. Then the objective function of (20) can be written as

$$\mathcal{F}(Y, \{Q_i\}_{i \in V}) = \sum_{i \in V} \operatorname{trace}\big((Y B_i - Q_i Z_i) J_{n_i} (Y B_i - Q_i Z_i)^T\big).$$

Note that J_{n_i} is a projection matrix and therefore $J_{n_i}^T = J_{n_i} = J_{n_i}^2$.

To minimize $\mathcal{F}$ in terms of Y, we set $\partial \mathcal{F} / \partial Y = 0$ and obtain

$$\frac{1}{2} \partial \mathcal{F} / \partial Y = \sum_{i \in V} (Y B_i - Q_i Z_i) J_{n_i} B_i^T = Y \Big(\sum_{i \in V} B_i J_{n_i} B_i^T\Big) - \Big(\sum_{i \in V} Q_i Z_i J_{n_i} B_i^T\Big) = 0, \tag{21}$$

which is a symmetric linear system with multiple right-hand sides Y. Since $\mathcal{F}$ is convex in terms of Y, the condition (21) is necessary and sufficient to reach the minimum. It is a standard orthogonal Procrustes problem to minimize $\|(Y_i - Q_i Z_i) J_{n_i}\|_F^2$ in terms of Q_i subject to $Q_i^T Q_i = I$. See, for example, Appendix A.1. Therefore, it is straightforward to minimize $\mathcal{F}$ in terms of Q_i subject to $Q_i^T Q_i$ for $i \in V$. The discussion leads to an alternating algorithm to solve (19).

Note that Problem (19) is not convex. Therefore, a global solution is not guaranteed, even though we alternatively solve the two subproblems optimally. Initialization is important to get a good local solution, and we use the existing y_i for $i \in \widehat{V}$ and the y_j from the greedy isometric refining method for $j \in V \setminus \widehat{V}$ for initialization.

Our discussion parallels that in the alternating algorithm for an isometric variant of LTSA discussed in the appendix of Zhang and Zha (2004). There are two key differences. First, instead of local PCA which relies on the high dimensional coordinates, we use local Isomap with the geodesic information propagated in the coarsening phase. Second, the initialization utilizes the embedding in the coarse level, which is not available in the setting of Zhang and Zha (2004).

3.3.3 Conformal Refining

The isometric refining schemes in Sections 3.3.1 and 3.3.2 can be extended for conformal refining as follows. We add a scale factor $c_i \geq 0$ to the program (18) and obtain

$$\begin{cases} \underset{c_i, Q_i, \gamma_i}{\text{minimize}} & \|\widehat{Y}_i - c_i\, Q_i \widehat{Z}_i - \gamma_i e_{n_i}^T\|_F^2 \\ \text{subject to} & c_i \geq 0,\ Q_i^T Q_i = I_d,\ Q_i \in \mathbb{R}^{d\times d},\ \gamma_i \in \mathbb{R}^d, \end{cases} \tag{22}$$

for each vertex $i \in V \setminus \widehat{V}$. A solution to (22) is given in Appendix A.5, which follows Sibson (1978). With the minimizer c_i, Q_i, γ_i of (22), we can embed $y_i = c_i Q_i z_i - \gamma_i$. The procedure is repeated for all $i \in V \setminus \widehat{V}$ to complete one level of uncoarsening. We call the scheme *greedy conformal refining*. If $V \setminus \widehat{V}$ is not self-repellent, an enhancement can be made in a similar way to that of Section 3.3.1.

To improve the quality of the conformal embedding, we can add, for $i \in V$, a scalar $c_i \geq 0$ to the program (19) and obtain

$$\begin{cases} \underset{c_i, Q_i, \gamma_i}{\text{minimize}} & \sum_{i\in V} \|Y_i - c\, Q_i Z_i - \gamma_i e_{n_i}^T\|_F^2 \\ \text{subject to} & c_i \geq 0,\ Q_i^T Q_i = I_d,\ Q_i \in \mathbb{R}^{d\times d},\ \gamma_i \in \mathbb{R}^d,\ \forall i \in V, \end{cases} \tag{23}$$

where $Y_i = [y_j]_{j\in\mathcal{N}_i}$ contains the global coordinates and $Z_i = [z_j^{(i)}]_{j\in\mathcal{N}_i}$ contains the local coordinates for all $i \in V$. The problem (23) can be solved sub-optimally by an alternating algorithm similar to that for (19). We call the resulting method *alternating conformal refining*. Note that the solution to (46) in Appendix A.1 is an ingredient for solving the program (19), whereas we use the solution to (53) in Appendix A.5 for solving (23).

4. Algebraic Multilevel Nonlinear Dimensionality Reduction

We consider a class of spectral methods for manifold learning, including LLE, LE, and LTSA. These methods use the bottom eigenvectors of a symmetric positive semidefinite matrix M to define the embedding. The multilevel framework presented in this section can coarsen the matrix M by restriction and uncoarsen the solution by prolongation, as it is common in algebraic multigrid methods (AMG). Hence we call this framework algebraic. The goal of this section is to develop multilevel techniques to incorporate LLE, LE, and LTSA. The resulting methods are called multilevel LLE, multilevel LE, and multilevel LTSA.

Recall that the symmetric matrix in Section 3 is built at the bottom level using the geodesic information from the coarsening phase. The key difference here is that we construct the symmetric positive semidefinite matrix M at the top level, and this matrix M is the same as the one in the original manifold learning method.

Another type of spectral embedding is to use the low rank approximation of a symmetric matrix, called a *kernel* matrix. Isomap and SDE are two such examples. To obtain a kernel matrix, Isomap solves an all-pairs shortest path problem and SDE resorts to semidefinite programming. In both cases, the aim is to preserve the isometry, and the cost of the kernel matrix dominates the whole computation of manifold learning. For computational efficiency, the geometric multilevel framework in Section 3 is favored for these two methods, as well as for their conformal variants C-Isomap and C-SDE.

4.1 The Coarsening Phase

The manifold learning algorithms considered here utilize an affinity graph and eventually compute the eigenvectors of a symmetric matrix M for embedding. The meaning of coarsening is twofold. The first is that of graph coarsening (geometric), and the second is that of find a coarse representation of the symmetric matrix M in the form $P^T MP$ (algebraic), where P is the prolongation matrix. However the two concepts are correlated closely with each other. The discussion is focused on one level of coarsening. The extension for a multilevel mechanism is straightforward. We begin with LLE, and will generalize the scheme for LTSA and LE.

4.1.1 Edge Weights of LLE

Consider LLE, which uses a directed affinity graph $G = (V, E)$ with $V = \{1, \ldots, n\}$, where each edge $(i, j) \in E$ is associated with a weight $w_{ij} \in \mathbb{R}$. The weights form a matrix $W = [w_{ij}] \in \mathbb{R}^{n\times n}$, satisfying $w_{ij} = 0$ for $(i, j) \notin E$.

Algorithm 1 can be used to coarsen the affinity graph $G = (V, E)$ to obtain $\widehat{G} = (\widehat{V}, \widehat{E})$, where $\widehat{V} \subset V$. To apply LLE with the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$, we need to assign a weight $\widehat{w}_{ij} \in \mathbb{R}$ to each edge $(i, j) \in \widehat{E}$.

Without loss of generality, we reorder the vertex indices such that $\widehat{V} = \{1, \ldots, \widehat{n}\}$. The objective is to find a prolongation matrix $P \in \mathbb{R}^{n\times\widehat{n}}$ and use its transpose P^T as the restriction matrix. This is standard in an algebraic multigrid (AMG) methods. Eventually we will use P^T to 'restrict' the weights w_{ij} of the fine graph $G = (V, E)$ to obtain the weights $\widehat{w}_{ij}$ of the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$. The goal of the discussion here is to develop this prolongation matrix P. While the derivation is inspired by the LLE algorithm, the resulting P can be used with other manifold learning methods, such as LE and LTSA.

We impose the constraint $Pe_{\widehat{n}} = e_n$, so that each element in $Pv \in \mathbb{R}^n$ is a weighted average of elements of any given vector $v \in \mathbb{R}^{\widehat{n}}$. Unless otherwise noted, there is no assumption that the weights are nonnegative, just like the case that the weights of LLE can be negative. With the weights w_{ij} available, we can compute p_{ij} by

$$p_{ij} = \begin{cases} 1, & i = j; \\ 0, & i \neq j,\ i \in \widehat{V}; \\ w_{ij}/\sum_{k\in\widehat{V}} w_{ik}, & i \in V \setminus \widehat{V}. \end{cases} \tag{24}$$

This formula (24) has been used by Wang and Zhang (2006) in the context of multilevel semi-supervised clustering, where they constrain the weights to be symmetric and nonnegative, which is not a restriction in our case.

In both LLE and diffusion maps, the weights satisfy $\sum_{k\in V} w_{ik} = 1$. We further assume that $V \setminus \widehat{V}$ is self-repellent. Then $w_{ik} = 0$ for $i, k \in V \setminus \widehat{V}$. Therefore we have $\sum_{k\in\widehat{V}} w_{ik} = 1$ for $i \in V \setminus \widehat{V}$, which implies that in (24), $p_{ij} = w_{ij}$ for $i \in V \setminus \widehat{V}$.

There are manifold learning algorithms that do not require forming edge weights, e.g., LTSA. For these methods, we consider another approach. Let $X = [x_i]_{i\in V}$ and $\widehat{X} = [x_i]_{i\in\widehat{V}}$. The following program is considered.

$$\begin{cases} \underset{P\in\mathbb{R}^{n\times\widehat{n}}}{\text{minimize}} & \|X^T - P\widehat{X}^T\|_F^2 \\ \text{subject to} & \sum_{j=1}^{\widehat{n}} p_{ij} = 1 \qquad \forall i = 1,\dots,n; \\ & p_{ij} = 0 \qquad \forall (i,j) \notin E \wedge i \neq j. \end{cases} \tag{25}$$

This formulation (25) was adopted by Weinberger et al. (2005) for their L-SDE method, in the setting where edge weights w_{ij} are not available. The justification is as follows. We approximate each x_i with $i \in V \setminus \widehat{V}$ by a weighted average of its neighbors in $\widehat{V}$, i.e., x_j with $(i,j) \in E$ and $j \in \widehat{V}$. For $i \in \widehat{V}$, the best approximation of x_i is x_i itself. In other words,

$$\forall i \in \widehat{V},\ p_{ii} = 1 \quad \text{and} \quad \forall i,j \in \widehat{V} \text{ and } i \neq j,\ p_{ij} = 0. \tag{26}$$

Since we have reordered the vertex indices such that $\widehat{V} = \{1,\dots,\widehat{n}\}$, the upper $\widehat{n}$-by-$\widehat{n}$ submatrix of P is an identity $I_{\widehat{n}}$. The rest is to determine p_{ij} for $i = \widehat{n}+1,\dots,n$ and $j = 1,\dots,\widehat{n}$.

Let $\widehat{\mathcal{N}}_i = \{j : j \in \widehat{V} \wedge (i,j) \in E\}$. We divide the program (25) into the following subproblems for $i \in V \setminus \widehat{V}$.

$$\begin{cases} \underset{p_{ij}}{\text{minimize}} & \|x_i - \sum_{j\in\widehat{\mathcal{N}}_i} p_{ij} x_j\|^2 \\ \text{subject to} & \sum_{j\in\widehat{\mathcal{N}}_i} p_{ij} = 1. \end{cases} \tag{27}$$

The program (27) is in the same form as that used in LLE to determine the weights, and the minimizer can be found from solving a linear system. Aggregating the results from (26) and (27), we obtain the minimizer $P \in \mathbb{R}^{\widehat{n}\times n}$ of (25).

If $V \setminus \widehat{V}$ is self-repellent, then all neighbors of vertex i are in $\widehat{V}$ for $i \in V \setminus \widehat{V}$, in which case (27) gives exactly the LLE weights, and therefore the solution to (25) is identical to the weight formula (24).

Now we describe how to form a weight matrix $\widehat{W} = [\widehat{w}_{ij}] \in \mathbb{R}^{\widehat{n}\times\widehat{n}}$ for the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$ with a given weight matrix $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ of the fine graph $G = (V, E)$ using a prolongation matrix $P \in \mathbb{R}^{n\times\widehat{n}}$. Denote by w_i the row vector formed by the row i of W, and by $\widehat{w}_i$ the row vector formed by the row i of $\widehat{W}$. By restriction, we compute $\widehat{w}_i = w_i P$ for $i = 1\dots,\widehat{n}$. Equivalently, $\widehat{W} \in \mathbb{R}^{\widehat{n}\times\widehat{n}}$ is formed by dropping the last $n - \widehat{n}$ rows of WP. To write this succinctly, we partition W and P as

$$W = \begin{array}{c} \\ \widehat{n} \\ n-\widehat{n} \end{array} \begin{array}{c} \begin{array}{cc} \widehat{n} & n-\widehat{n} \end{array} \\ \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \end{array}, \qquad P = \begin{array}{c} \\ \widehat{n} \\ n-\widehat{n} \end{array} \begin{array}{c} \widehat{n} \\ \begin{bmatrix} I_{\widehat{n}} \\ P_2 \end{bmatrix} \end{array}. \tag{28}$$

Then $\widehat{W} = W_{11} + W_{12}P_2$. Three properties should be noted.

1. If there is no edge between vertices i and j, then we would like the corresponding weight to be zero. Since we constrain the prolongation matrix $P = [p_{ij}]$ to have $p_{ij} = 0$ for $i \in V \setminus \widehat{V}$ and $(i,j) \notin E$, this property is inherited in the weight matrix $\widehat{W} = [\widehat{w}_{ij}]$ of the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$. That is, $(i,j) \notin \widehat{E}$ implies $\widehat{w}_{ij} = 0$, under the assumption that $(i,j) \notin E$ implies $w_{ij} = 0$.

2. In both LLE and diffusion maps, the weight matrix $W \in \mathbb{R}^{n\times n}$ satisfies $We_n = e_n$. Since we impose the constraint $Pe_{\widehat{n}} = e_n$, this property is also inherited in $\widehat{W}$, i.e., $\widehat{W}e_{\widehat{n}} = e_{\widehat{n}}$ if $We_n = e_n$.

3. The diagonal of $\widehat{W}$ may not be zero even if W has a zero diagonal. In other words, the corresponding $\widehat{G} = (\widehat{V}, \widehat{E})$ may contain self-edges. Definition 1 is still valid, but Algorithm 1 should be revised to handle self-edges for the next level of coarsening.

4.1.2 Coarsening by Restriction

Given the input high dimensional data, LLE constructs an affinity graph, forms a weight matrix $W \in \mathbb{R}^{n\times n}$ which satisfies $We_n = e_n$, and eventually computes the bottom eigenvectors of $(I-W)^T(I-W)$ for a low dimensional embedding. With a prolongation matrix $P \in \mathbb{R}^{n\times\widehat{n}}$, it is a natural attempt to treat $P^T(I_n-W)^T(I_n-W)P$ as a coarse presentation of $(I_n-W)^T(I_n-W)$, and use the bottom eigenvectors of $P^T(I_n-W)^T(I_n-W)P$ for an embedding of the coarse level. The procedure is inspired by the algebraic multigrid methods (AMG), and it can also be justified from the graph point of view as follows.

Recall that the weight matrix $\widehat{W}$ of the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$ is obtained from dropping the last $n - \widehat{n}$ rows of WP. We partition W and P into the form (28), and then write $(I_n - W)P$ as

$$(I_n - W)P = P - WP = \begin{bmatrix} I_{\widehat{n}} \\ P_2 \end{bmatrix} - \begin{bmatrix} \widehat{W} \\ W_{21} + W_{22}P_2 \end{bmatrix} = \begin{bmatrix} I_{\widehat{n}} - \widehat{W} \\ P_2 - W_{21} - W_{22}P_2 \end{bmatrix}. \quad (29)$$

Now assume that $V \setminus \widehat{V}$ is self-repellent. Then there is no edge between vertices in $V \setminus \widehat{V}$, and hence $W_{22} = 0$. In addition, if P is from (24) and $We_n = e_n$, then $P_2 = W_{21}$. Therefore, $P_2 - W_{21} + W_{22}P_2 = 0$ in (29). That is, $(I_n - W)P$ is essentially the same as $I_{\widehat{n}} - \widehat{W}$, except that the bottom $n - \widehat{n}$ rows of $(I_n - W)P$ are zero. We conclude that if $V \setminus \widehat{V}$ is self-repellent and the prolongation matrix P is obtained by (24), then

$$P^T(I_n - W)^T(I_n - W)P = (I_{\widehat{n}} - \widehat{W})^T(I_{\widehat{n}} - \widehat{W}).$$

Thus, using the bottom eigenvectors of $P^T(I_n-W)^T(I_n-W)P$ for embedding is equivalent to applying LLE to the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$ with edge weights $\widehat{W}$. Even if $V \setminus \widehat{V}$ is not self-repellent, we can still use the eigenvectors of $P^T(I_n - W)^T(I_n - W)P$ for an embedding, ignoring the interpolation deviation $P_2 - W_{21} + W_{22}P_2$ in (29).

Unlike LLE which uses a directed affinity graph, LE uses an undirected affinity graph $\bar{G} = (\bar{V}, \bar{E})$ and forms a symmetric matrix $\bar{W} \in \mathbb{R}^{n\times n}$ consisting of edge weights. We have intentionally added a bar on the top of each symbol, to distinguish it from the notation

used for LLE. As outlined in Appendix B.3, the embedding of LE is formed by the bottom (generalized) eigenvectors of the Laplacian matrix $\bar{L} = \bar{D} - \bar{W}$, where $\bar{D}$ is the diagonal matrix formed by the elements in $\bar{W}e_n$. Hence $\bar{L}$ of LE plays the role of $(I_n - W)^T(I_n - W)$ of LLE. The graph $\bar{G}$ of LE is intrinsically different from the graph G of LLE. Hence we propose the following coarsening scheme for LE.

We use Algorithm 1 to find a representation $\widehat{V}$ of V, but do not use the edge set $\widehat{E}$ in Algorithm 1. The prolongation matrix $P \in \mathbb{R}^{n \times \widehat{n}}$ is obtained by (24). Since the sparsity pattern of the Laplacian matrix $\bar{L}$ is canonically associated with a graph, we use $P^T \bar{L} P$ to define the coarse graph and its edge weights, which are used for the next level of coarsening. The idea of using the sparsity pattern to define a coarse graph has been utilized in the multilevel semi-supervised clustering (Wang and Zhang, 2007).

4.2 The Dimensionality Reduction Phase

The manifold learning methods considered in this section use the bottom eigenvectors of a symmetric positive semidefinite matrix, denoted by M, for embedding. To be specific, these methods minimize trace(YMY^T) subject to certain constraints. Three such examples are listed below.

1. LLE minimizes trace$(Y(I_n - W)^T(I_n - W)Y^T)$ subject to $YY^T = I_n$ and $Ye_n = 0$, where W is a generally asymmetric weight matrix which satisfies $We_n = e_n$.

2. LE minimizes trace$(Y(\bar{D} - \bar{W})Y^T)$ subject to $Y\bar{D}Y^T = I_n$ and $Y\bar{D}e_n = 0$, where $\bar{W}$ is a symmetric weight matrix, $\bar{D}$ is the diagonal matrix formed by elements in $\bar{W}e_n$.

3. LTSA minimizes trace$(Y(SHH^TS^T)Y^T)$ subject to $YY^T = I_n$ and $Ye_n = 0$, where H is an aggregated transformation matrix and S is an aggregated boolean selection matrix.

More information of these manifold learning methods can be found in Appendix B. Using the multilevel coarsening scheme in Section 4.1, we obtain a succession of smaller graphs $G^{(1)}, \ldots, G^{(r)}$ which approximate the original affinity graph $G^{(0)}$. A sequence of prolongation matrices $P^{(1)}, \ldots, P^{(r)}$ are also generated concurrently. The corresponding coarse versions of matrix M can be obtained by

$$M^{(l)} = (P^{(l)})^T M^{(l-1)} P^{(l)}, \qquad l = 1, \ldots, r, \tag{30}$$

where the base case is $M^{(0)} = M$. Note that $M^{(l)}$ is symmetric positive semidefinite and $M^{(l)}e = 0$ for $l = 0, \ldots, r$, where e is a column vector of ones of appropriate size.

In addition to (30), an alternative to obtain $M^{(l)}$ is to apply the original manifold learning algorithm to $X^{(l)} = [x_i]_{i \in V^{(l)}}$ with the affinity graph $G^{(l)} = (V^{(l)}, E^{(l)})$. We use this way for multilevel LTSA since we found that it usually yields better embedding quality than (30) in our experiments. On the other hand, we simply use (30) for multilevel LLE and multilevel LE, since none of the two methods showed significant advantage over the other.

At the bottom level, we compute the embedding using the coarsest $M^{(r)}$. To be more specific, if the matrix $M^{(0)}$ at the top level is from LLE or LTSA, the embedding at the bottom level is formed by the d eigenvectors corresponding to the second to the $(d+1)$st

smallest eigenvalues of $M^{(r)}$. If M at the top level is from LE, we solve the generalized eigenvalue problem $M^{(r)}v = \lambda D^{(r)}v$, where $D^{(r)}$ is the diagonal matrix formed by the diagonal of $M^{(r)}$. Then the embedding is formed by the d eigenvectors corresponding to the second to the $(d+1)$st smallest generalized eigenvalues.

4.3 The Uncoarsening Phase

We present three schemes for uncoarsening the low dimensional embedding at the bottom level. The first scheme follows the standard AMG approach (Saad, 2003; Trottenberg et al., 2000). The other two schemes are inspired by the semi-supervised manifold learning (Ham et al., 2005).

4.3.1 Refining by Prolongation

For simplicity, we first consider one level of coarsening and refining. A prolongation matrix P can be obtained from solving (25), which approximates X by $\widehat{X}P^T$. Hence it is natural to apply the prolongation matrix P for refining, i.e., $Y = \widehat{Y}P^T$.

We have a sequence of prolongation matrices $P^{(1)}, \ldots, P^{(l)}$ and the coarse embedding $Y^{(r)}$ at the coarsest level. The uncoarsening is performed level by level via

$$Y^{(l-1)} = Y^{(l)}(P^{(l)})^T, \qquad l = r, r-1, \ldots, 1, \tag{31}$$

where $Y^{(0)}$ at the top level is an embedding of all input high dimensional points.

This refining method is generic, and can be used with a wide variety of manifolding learning algorithms. Indeed, it is also applicable in the geometric multilevel framework given in Section 3. The prolongation matrix can still be determined using the affinity graphs in adjacent levels. If the edge weights are available, we can use the formula (24). Otherwise, we can solve (25) with the high dimensional data. In both cases, we do not need have to form a symmetric matrix used for spectral embedding at the top level. Hence (31) can be incorporated in the framework in Section 3.

Note that (31) is a linear method. Chaining all the prolongation matrices across all levels, the embedding at top level is $Y^{(0)} = Y^{(l)}(P^{(l)})^T \cdots (P^{(1)})^T$, a linear projection of the embedding at the bottom level. What follows are two nonlinear methods which aim to minimize the same objective function of the manifold learning algorithm applied at the bottom level.

4.3.2 Landmark-based Refining

Consider one level of refining for Y. Without loss of generality, we permute the columns of $Y \in \mathbb{R}^{d\times n}$ such that Y can be written as $Y = [Y_1, Y_2]$, where the columns of $Y_1 \in \mathbb{R}^{d\times\widehat{n}}$ are the points in the lower dimensional space already embedded at the coarse level, and $Y_2 \in \mathbb{R}^{d\times(n-\widehat{n})}$ contains the points to be determined. We also perform the corresponding symmetric permutation and partition of M. The partitioned M and Y can be written as

$$M = \begin{array}{c} \\ \widehat{n} \\ n-\widehat{n} \end{array} \begin{array}{c} \begin{array}{cc} \widehat{n} & n-\widehat{n} \end{array} \\ \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \end{array}, \qquad Y = \begin{array}{c} \begin{array}{cc} \widehat{n} & n-\widehat{n} \end{array} \\ \begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \end{array}. \tag{32}$$

Using the notation in (30) and (31), Y is $Y^{(l-1)}$, Y_1 is $Y^{(l)}$, and M is $M^{(l-1)}$ at level l.

In LLE, LTSA, or LE, the objective function to minimize can be written as

$$\begin{aligned} \operatorname{trace}(YMY^T) &= \operatorname{trace}\left(\begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix} \right) \\ &= \operatorname{trace}(Y_1 M_{11} Y_1^T) + 2\operatorname{trace}(Y_1 M_{12} Y_2^T) + \operatorname{trace}(Y_2 M_{22} Y_2^T). \end{aligned} \tag{33}$$

We use the determined Y_1 as landmarks, and set the gradient of $\operatorname{trace}(YMY^T)$ in terms of Y_2 to be zero and obtain

$$\frac{\partial}{\partial Y_2}\operatorname{trace}(YMY^T) = Y_1 M_{12} + Y_2 M_{22} = 0 \quad \Rightarrow \quad Y_2 M_{22} = -Y_1 M_{12}. \tag{34}$$

The matrix M is symmetric positive semidefinite. Therefore so is M_{22}, which implies that the quadratic optimization problem is convex. Hence (34) implies that Y_2 is a global minimizer and vice versa. In LE, M is the graph Laplacian, which is a discrete version of the continuous Laplacian operator, and the condition (34) means that the corresponding continuous function y is harmonic in the unsupervised part (Zhu et al., 2003).

In the proposed multilevel framework the matrix M in (33) and (34) is substituted by the coarse $M^{(l)}$ defined in (30) at each uncoarsening level $l = r-1, r-2 \ldots, 0$, where we have $M^{(0)} = M$ at the top level. Note that the prolongation matrix $P^{(l)}$ has full column rank for $l = 1, \ldots, r$. Recall that the positive semidefiniteness of M is inherited by all coarsened matrices $M^{(1)}, \ldots, M^{(r)}$.

Sometimes the matrix M can be written as AA^T and therefore $\operatorname{trace}(YMY^T) = \|YA\|_F^2$. For example, we have $A = (I - W)^T$ in LLE and $A = SH$ in LTSA. See Appendix B for more information. In such cases, we can simply minimize

$$\|YA\|_F^2 = \left\| \begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \right\|_F^2 = \|Y_1 A_1 + Y_2 A_2\|_F^2, \tag{35}$$

where we partition the rows of A corresponding to to the column partition of Y. Note that minimizing (35) in terms of Y_2 is a standard least square problem, which is equivalent to solving the following linear system for Y_2

$$Y_2(A_2 A_2^T) = -Y_1 A_1 A_2^T. \tag{36}$$

This in turn is equivalent to (34), since $A_2 A_2^T = M_{22}$ and $A_1 A_2^T = M_{12}$.

Note that this strategy can be extended to a multilevel version. More precisely, at level l, we use $A^{(l)} = (P^{(l)})^T A^{(l-1)}$ to replace A in (35) and (36) for $l = 1, \ldots, r$. At the top level we have $A^{(0)} = A$.

4.3.3 Regularized Regression Refining

In the landmark-based refining, once a point is embedded in the low dimensional space, it is fixed. Here we consider a related alternative which allows to alter existing embedded points in the uncoarsening phase. Let $Y = [y_i]_{i \in V}$ be the embedding of the fine level to be determined, and $\widehat{Y} = [\widehat{y}_i]_{i \in \widehat{V}}$ be the embedding from the coarse level which has been

determined, where $\widehat{V} \subset V$. We also let $Y_1 = [y_i]_{i\in\widehat{V}}$. With the embedded points $\widehat{y}_i$ for $i \in \widehat{V} \subset V$ from the coarse level, we minimize in terms of Y,

$$\text{trace}(YMY^T) + \sum_{i\in\widehat{V}} c_i\|y_i - \widehat{y}_i\|^2 = \text{trace}(YMY^T) + \|(Y_1 - \widehat{Y})C^{1/2}\|_F^2, \tag{37}$$

where $c_i > 0$ is a preset penalty parameter and $C \in \mathbb{R}^{\widehat{n}\times\widehat{n}}$ is the diagonal matrix formed by c_i for $i \in \widehat{V}$. The first term of (37) corresponds to the *smooth constraint* and the second term corresponds to the *fitting constraint* in semi-supervised learning (Ham et al., 2005; Zhou et al., 2003).

Setting the gradient of (37) to be zero, we obtain

$$YM + (Y_1 - \widehat{Y})\begin{bmatrix} C & Z \end{bmatrix} = 0 \quad \Rightarrow \quad Y\begin{bmatrix} M_{11} + C & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} \widehat{Y}C & Z \end{bmatrix}, \tag{38}$$

where Z is the zero matrix of size $\widehat{n}$-by-$(n-\widehat{n})$, and we have partitioned M as that in (32).

Driving $\min\{c_i : i \in \widehat{V}\} \to \infty$, then there is no tolerance of the fitting constraint; hence $Y_1 = \widehat{Y}$ and the solution to (38) converges to a solution to (33). In practice, we set the fitting parameter $c_i = c > 0$ straight for $i \in \widehat{V}$ for simplicity.

5. Embedding Quality Assessment

The quality of the nonlinearly mapped data can be evaluated in various ways, e.g., the measurement of isometric error and the conformal error (Goldberg and Ritov, 2009; Sibson, 1978). For a proximity preserving embedding, we can use a rank-based method to assess the quality (Lee and Verleysen, 2009; Venna and Kaski, 2006).

5.1 Isometric Measurement

Given $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ and $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$, the *Procrustes static* (Goldberg and Ritov, 2009; Sibson, 1978) is defined as

$$\mathcal{G}(X, Y) = \min_{Q,\gamma;\, Q^TQ=I_d} \sum_{i=1}^{n} \|x_i - Qy_i - \gamma\|^2 = \min_{Q,\gamma;\, Q^TQ=I_d} \|X - QY - \gamma e_n^T\|_F^2. \tag{39}$$

The minimization can be transformed to solving a singular value problem. See, for example, Appendix A.1. If X and Y consist of close sample points on two manifolds $\Omega \subset \mathbb{R}^d$ and $\mathcal{M} \subset \mathbb{R}^n$, respectively, then (39) measures the local isometric error. See Appendix A.4 for a discussion.

A manifold learning algorithm maps the input $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ to the output $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$. Let $\mathcal{N}_1, \ldots, \mathcal{N}_n$ be the neighborhood sets of the n points. We define $X_i = [x_j]_{j\in\mathcal{N}_i}$ and $Y_i = [y_j]_{j\in\mathcal{N}_i}$ as matrices consisting of the neighborhood points of x_i and y_i, respectively. We consider the isometric-preserving algorithms, such as Isomap and SDE, and define the *isometric measure* by

$$\mathcal{R}(X, Y, \{\mathcal{N}_i\}_{i=1}^n) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{G}(X_i, Y_i), \tag{40}$$

and the *normalized isometric measure* by

$$\mathcal{R}_N(X, Y, \{\mathcal{N}_i\}_{i=1}^n) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{G}(X_i, Y_i)/\|X_i J_{|\mathcal{N}_i|}\|_F, \tag{41}$$

where $J_k = I_k - \frac{1}{k}e_k e_k^T$ is the centering matrix. Both (40) and (41) measure the isometric errors. They were introduced by Goldberg and Ritov (2009).

5.2 Conformal Measurement

For conformal measurement, we add a scale factor $c \geq 0$ into (39) and obtain

$$\mathcal{G}_C(X, Y) = \min_{c\geq 0,\, Q,\gamma;\; Q^TQ=I_d} \sum_{i=1}^{n} \|x_i - c\,Qy_i - \gamma\|^2 = \min_{c\geq 0,\, Q,\gamma;\; Q^TQ=I_d} \|X - c\,QY - \gamma e_n^T\|_F^2. \tag{42}$$

Appendix A.5 provides a solution to (39) which, to our knowledge, was first given by Sibson (1978). If X and Y consist of close sample points on two manifolds $\Omega \subset \mathbb{R}^d$ and $\mathcal{M} \subset \mathbb{R}^n$, respectively, then (39) measures the local conformal error.

Relying on (42), Goldberg and Ritov (2009) introduced the *normalized conformal measure*

$$\mathcal{R}_C(X, Y, \{\mathcal{N}_i\}_{i=1}^n) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{G}_C(X_i, Y_i)/\|X_i J_{|\mathcal{N}_i|}\|_F, \tag{43}$$

where $J_k = I_k - \frac{1}{k}e_k e_k^T$ is the centering matrix. This measure (43) is essentially the conformal error, which is used to access the embedding quality of a manifold learning algorithm which aims to preserve local angles, e.g., C-Isomap.

5.3 Rank-based Criteria

We describe two rank-based evaluation metrics, the *trustworthiness* and *continuity* of the proximity relationships of data entries (Venna and Kaski, 2006). Other rank-based criteria can be found in Lee and Verleysen (2009).

Let $x_1, \ldots, x_n$ be the points in the high dimensional space, and $y_1, \ldots, y_n$ be the mapped points in the low dimensional space. Denote by $r(i, j)$ the rank of x_j in the ordering according to the distance from x_i. The longest vertex x_j from x_i has $r(i, j) = 1$, and the shortest vertex x_j from x_i has $r(i, j) = n-1$. Likewise, denote by $\bar{r}(i, j)$ the rank of y_j in the ordering according to the distance from y_i. In the case of ties in rank ordering, all compatible rank orders are assumed equally likely. The trustworthiness is defined by

$$T(h) = \frac{2}{nh(2n - 3h - 1)} \sum_{i=1}^{n} \sum_{j\in\mathcal{U}_h(i)} (r(i, j) - h), \tag{44}$$

where $\mathcal{U}_h(i)$ contains the indices of h nearest neighbors of y_i in the low dimensional space. The continuity is defined by

$$C(h) = \frac{2}{nh(2n - 3h - 1)} \sum_{i=1}^{n} \sum_{j\in\mathcal{V}_h(i)} (\bar{r}(i, j) - h), \tag{45}$$

where $\mathcal{V}_h(i)$ contains the indices of h nearest neighbors of x_i in the high dimensional space. Note that we have modestly assumed the parameter $h < N/2$.

The higher the trustworthiness or continuity, the better the manifold mapping. Both $T(h)$ and $C(h)$ are bounded above by 1. The upper bound 1 is reached if and only if $\mathcal{U}_h(i) = \mathcal{V}_h(i)$ for $i = 1, \ldots, n$, which means that the h nearest neighbors for each data entry in the high dimensional space coincide with those in the low dimensional space.

The larger the number of neighbors h, the lower the value of the trustworthiness $T(h)$ and the continuity $C(h)$. In practice, it is typical to use a kNN graph in a manifold learning algorithm. Hence we set h as k, the number of nearest neighbors for each vertex in the kNN graph. In general, $\mathcal{U}_h(i)$ in (44) and $\mathcal{V}_h(i)$ in (45) can be replaced by $\mathcal{N}_i$, the set of neighbors of vertex i.

6. Experiments

This section illustrates the application of the proposed multilevel manifold learning methods. We incorporate the following nonlinear dimensionality reduction methods into the multilevel framework: Isomap, C-Isomap, LLE, LE, LTSA, SDE, and C-SDE. Note that C-SDE is a conformal variant of SDE described in Section 2.2.3, whereas the other 6 methods are from the literature. See Table 1.

All experiments were performed in Matlab on a PC equipped with a four-core Intel Xeon E5504 @ 2.0GHz processor and 4GB memory. The kNN graph construction is by a brute-force algorithm, which can be improved by an approximation algorithm (Chen et al., 2009). We used a C/C++ implementation of Dijkstra's algorithm (Dijkstra, 1959) by John Boyer to solve the all-pair shortest path problem, which arises in Isomap and C-Isomap. We used the software package CSDP (Borchers, 1999) to solve the semidefinite programming problems in SDE and C-SDE, where we set the maximum number of iterations to 50. We use the same setting when these manifold learning methods are incorporated into the multilevel frameworks. In addition, in the geometric framework, if the alternating isometric or conformal refining method is used, we set the number of iterations to 8. See Section 3.3.2. In the algebraic framework, if the regularized regression refining method is used, we set the fitting parameter $c = 1$. See Section 4.3.3.

The results using 4 synthetic data sets are displayed in Section 6.1. Sections 6.2 and 6.3 report the experiments on `Sculpture Face` images and `Frey Face` video frames, respectively. The primary goal here is to show the effect of the multilevel schemes in the embedding rather than the computational savings. The intrinsic dimension is assumed known in all experiments. In Section 6.4 we illustrate an application of the multilevel techniques to intrinsic dimension estimation.

6.1 Synthetic Data

We experimented on 4 synthetic data sets: `Swissroll`, `Swisshole`, `Conformal Fishbowl`, and `Uniform Fishbowl`. They are all two-dimensional manifolds in three-dimensional space. Each data set contains $n = 800$ sample points on the manifold $\mathcal{M} = f(\Omega)$, as shown in the first row of Figure 4. The second row gives the plots of the corresponding points in the parameter space Ω. The results of applying the 7 manifold learning methods are also

displayed in Figure 4. In all we used a kNN graph with $k = 7$, except for that in LLE the kNN graph was with $k = 12$. We have also included PCA for comparison purposes.

We refer to Figure 4. Compared to `Swissroll`, `Swisshole` has a hole made in it. There exists an isometry for each of them, and the isometric mapping is successfully discovered by SDE. The `Swissroll` has a convex parameter space Ω, whereas `Swisshole` does not. Recall that Isomap assumes the parameter space is convex. This explains that Isomap performs well on `Swissroll` but generates a distorted image using `Swisshole`. We will show that multilevel Isomap relaxes the convexity assumption to some degree. PCA is a linear method and hence cannot unfold `Swissroll` or `Swisshole`. LTSA uses local tangent subspaces and preserves certain geometric information. LLE and LE, both relying on the proximity of points, reveal the topology of `Swissroll` or `Swisshole`.

The `Conformal Fishbowl` and `Uniform Fishbowl` data sets are both generated with the manifold mapping $f : \Omega \subset \mathbb{R}^2 \to \mathcal{M} \subset \mathbb{R}^3$ defined by

$$f(s,t) = \left(\frac{s}{1+s^2+t^2}, \frac{t}{1+s^2+t^2}, \frac{s^2+t^2}{1+s^2+t^2} \right).$$

One can verify that its Jacobian J_f satisfies

$$J_f^T J_f = \frac{1}{(1+s^2+t^2)^2} I_2.$$

Hence the condition (8) in Section 2.1.3 is satisfied and the mapping f is conformal. This mapping f has non-negotiable Gaussian curvature; therefore a corresponding isometry does not exist. As a result, it is inappropriate to seek an isometric embedding.

The `Conformal Fishbowl` has $y_1, \ldots, y_n$ uniformly distributed in the parameter space Ω. Therefore it satisfies the assumption of C-Isomap, which is followed by C-SDE. See the discussion in Section 2.2.3. As one can expect, both C-Isomap and C-SDE nicely discover the conformal mapping of `Conformal Fishbowl`.

On the other hand, the `Uniform Fishbowl` has $x_1, \ldots, x_n$ uniformly distributed in the manifold fold $\mathcal{M} = f(\Omega)$. As exhibited in Figure 4, the mappings obtained by C-Isomap and C-SDE do not really match the conformal parameterization, since the assumption discussed in Section 2.2.3 does not hold. We will show that multilevel C-Isomap and multilevel C-SDE can recover the conformal mapping to some extent.

Figures 5 to 8 demonstrate some results of multilevel methods. In all we used a kNN graph with $k = 7$, except for that in LLE and multilevel LLE, the kNN graph was with $k = 12$. In both geometric and algebraic multilevel frameworks, we set the degree of dependency $p = k - 1$ and did not impose the self-repellent complement constraint. See Definition 1. In the geometric multilevel framework, we used the alternating refining techniques in the uncoarsening phase. See Section 3.3. In the algebraic multilevel framework, we used the landmark-based refining method in the uncoarsening phase. See Section 4.3.

We use the embedding quality measures described in Section 5. For Isomap and SDE which aim for an isometric embedding, we use the normalized isometric measure $\mathcal{R}_N$ defined in (41). For C-Isomap and C-SDE which aim for a conformal embedding, we use the normalized conformal measure $\mathcal{R}_C$ defined in (43). For LTSA, LLE, and LE, we use the rank-based measures trustworthiness T and continuity C, defined in (44) and (45), respectively. The objectives of these manifold learning methods are followed while they are incorporated into

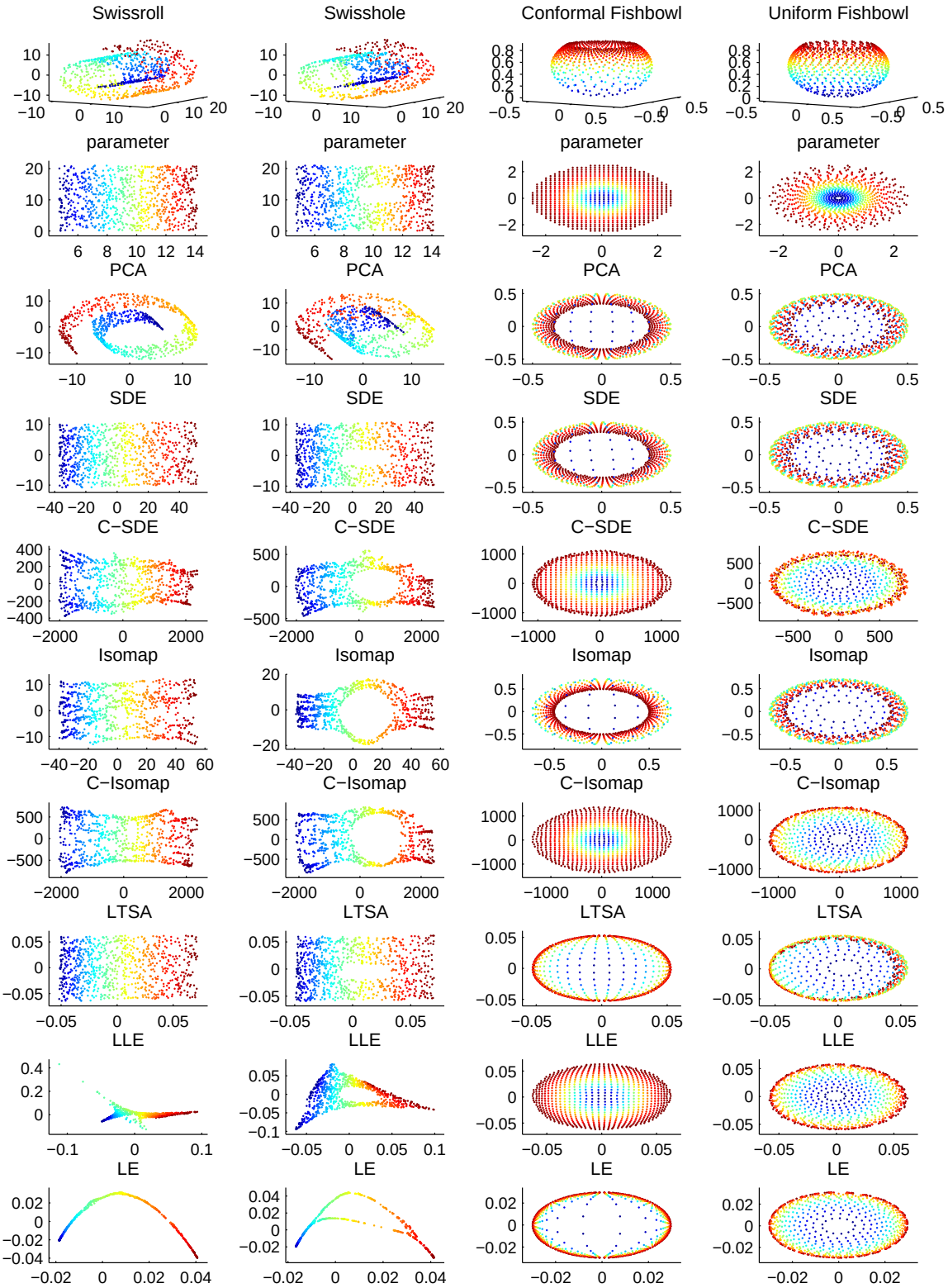


Figure 4: Results using 4 synthetic data sets.

multilevel framework. Hence we still the corresponding embedding quality measure. The measurement, as well as the number of sample points n_r at the bottom level r, is shown in each plot in Figures 5 to 8.

Figure 5 shows some two-dimensional mappings of `Swisshole`. The embedding by Isomap is unsatisfactory, due to the fact that the parameter space of `Swisshole` is not convex. The multilevel Isomap represents a substantial improvement over standard Isomap, since the isometric refining techniques aim directly to preserve local shapes. See Sections 3.3.1 and 3.3.2 for details. On the other hand, SDE does not require a convex parameter space and works very well on `Swisshole`, and the embedding quality of multilevel SDE is somewhat worse. However, recall that SDE resorts to the computationally expensive semidefinite programming which dominates the whole computation. The gain of multilevel SDE is the computational savings. In this example, CSDP (Borchers, 1999) took 462.44 seconds to solve the semidefinite program of SDE, whereas it took 290.09, 76.49, and 7.39 seconds to solve the semidefinite programs of multilevel SDE with the number of levels $r = 1, 2, 3$, respectively.

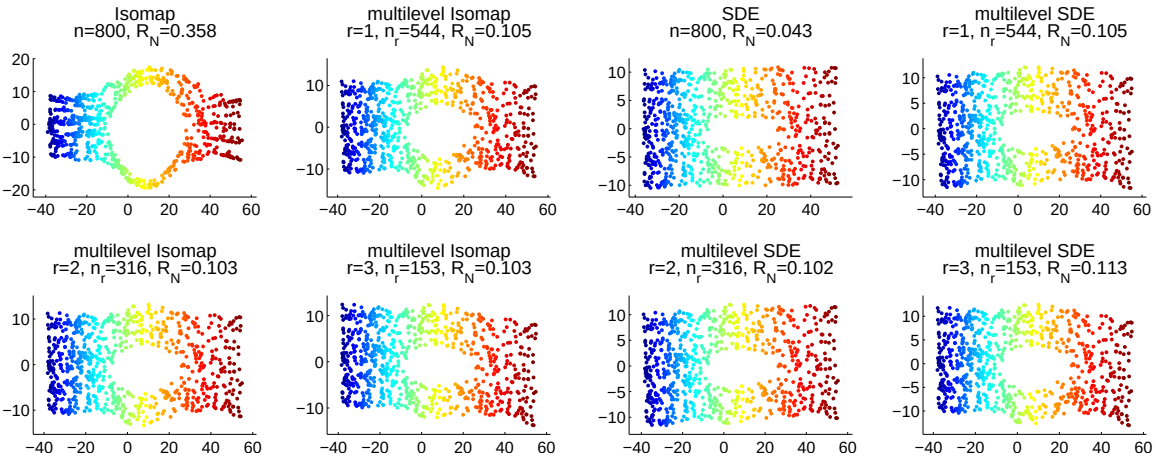


Figure 5: 2D mappings of `Swisshole` by multilevel Isomap & multilevel SDE.

Figure 6 shows some two-dimensional mappings of `Uniform Fishbowl`. In this case the sample points are uniformly distributed on the manifold instead of in the parameter space. Hence the conformal assumption in Section 2.2.3 is not satisfied. As one can expect, multilevel C-Isomap is a better scorer than C-Isomap. Indeed, the mappings by multilevel C-Isomap are visually similar to the parameter plot of `Uniform Fishbowl` in Figure 4. Compared to C-SDE, multilevel C-SDE improved the conformal measurement at a reduced cost. However, the rim of the embedding by multilevel C-Isomap or multilevel C-SDE is prone to be bent, since the conformal refining strategies in Section 3.3.3 preserve local angles rather than the global structure.

For LLE, LE, and LTSA, the multilevel schemes did not necessarily improve the measurements with respect to the rank-based criteria trustworthiness (44) and continuity (45). However, in the multilevel framework the eigenvalue problem at the bottom level can be much smaller. The computational savings can be significant with an efficient C or Fortran

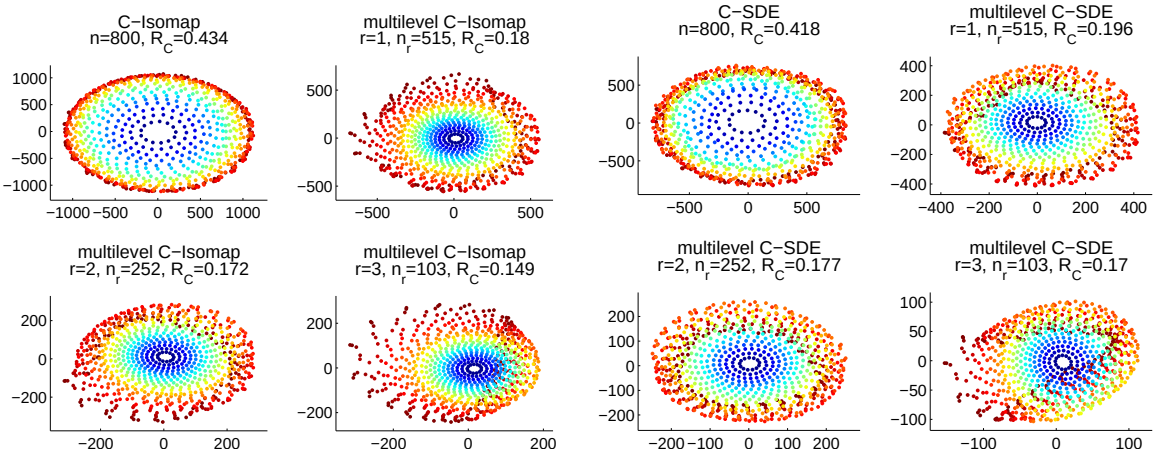


Figure 6: 2D mappings of `Uniform Fishbowl` by multilevel C-Isomap & multilevel C-SDE.

implementation of the multilevel techniques. Figure 7 and 8 show some two-dimensional mappings of `Swissroll` and `Conformal Fishbowl`, respectively.

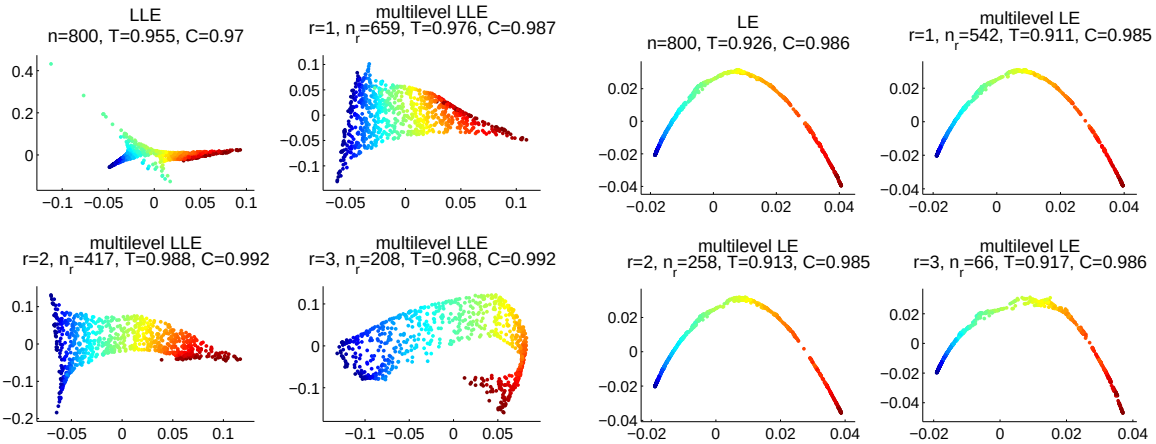


Figure 7: 2D mappings of `Swissroll` by multilevel LLE & multilevel LE.

In Section 3, we present two uncoarsening strategies for the geometric multilevel framework: greedy isometric or conformal refining and alternating isometric or conformal refining. The latter is more expensive but improves the embedding quality significantly. In Section 4, we present three refining strategies for the algebraic multilevel framework: prolongation refining, landmark-based refining, and regularized regression refining. Here we displayed only the results from the landmark-based refining. The prolongation refining and regularized regression refining also generated comparable results and there is no significant difference in the embedding quality. In the following experiments, we will give more details of comparison.

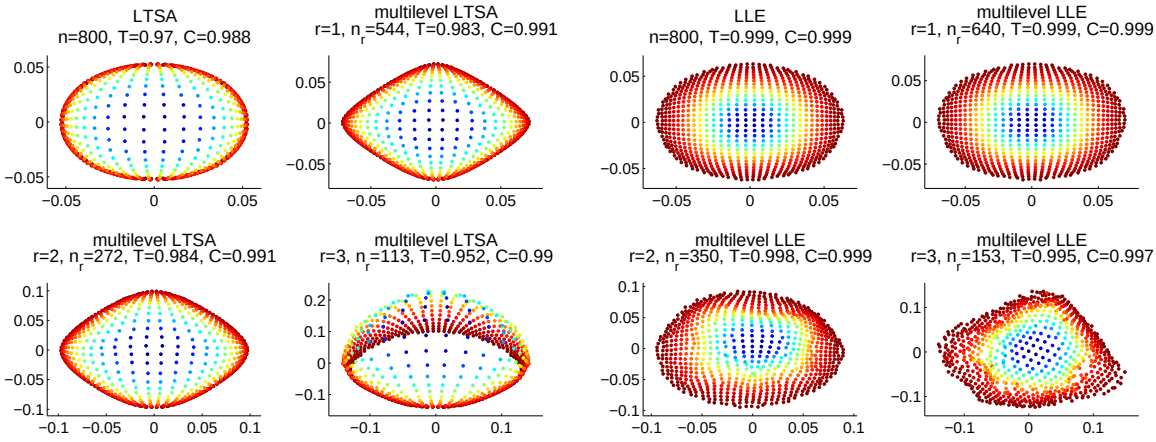


Figure 8: 2D mappings of `Conformal Fishbowl` by multilevel LTSA & multilevel LLE.

Note that in all experiments reported, we did not impose the self-repellent complement constraint. The self-repellent complement is a strong constraint which results in slow and damped coarsening. For example, if we impose the constraint for a self-repellent complement, using `Swissroll` with a directed kNN graph with $k = 7$ and degree of dependency $p = 1$, the numbers of vertices at level $l = 0, \ldots, 5$ are 800, 658, 578, 527, 493, 466, respectively. The numbers are unchanged for $p = 2, \ldots, 6$.

6.2 Sculpture Face Images

The `Sculpture Face` data set (Tenenbaum et al., 2000) contains 698 images of size 64-by-64 in grayscale of a sculpture face rendered with different poses and lighting directions[2]. Within the 4,096-dimensional input space, all of the images lie on an intrinsically three-dimensional manifold, that can be parameterized by three variables: left-right pose, up-down pose, and the lighting direction.

Table 2: Quality assessments of 2D mappings of `Sculpture` ($k = p = 6$), Part I.

# levels r	# points n_r	Isomap ($\mathcal{R}_N$) greedy refining	Isomap ($\mathcal{R}_N$) alter. refining	SDE ($\mathcal{R}_N$) greedy refining	SDE ($\mathcal{R}_N$) alter. refining	LE # pts. n_r	LE regress. trust.	LE refining conti.
0	698	0.695	0.695	**0.474**	**0.474**	698	0.977	0.988
1	501	**0.688**	0.516	0.546	0.516	501	0.978	0.987
2	298	0.704	**0.514**	0.568	0.514	248	0.978	0.987
3	131	0.739	0.514	0.651	0.514	50	**0.982**	**0.989**

Tables 2 and 3 report the quality measurements of the results using a kNN graph with $k = 6$ and embedding dimension $d = 3$. The best scores in each column are in boldface. In the multilevel frameworks, we set the degree of dependency to $p = 6$ and did not impose the

2. `http://isomap.stanford.edu/datasets.html`

Table 3: Quality assessments of 2D mappings of `Sculpture` ($k = p = 6$), Part II.

# levels r	# points n_r	eval.	LLE			LTSA		
			prolong. refining	landmark. refining	regress. refining	prolong. refining	landmark. refining	regress. refining
0	698	trust.	0.909	0.909	0.909	0.874	0.874	0.874
		conti.	0.971	0.971	0.971	0.968	0.968	0.968
1	555	trust.	0.920	0.920	0.920	0.951	0.950	0.949
		conti.	0.972	0.972	0.972	**0.987**	**0.987**	**0.987**
2	367	trust.	**0.948**	**0.948**	**0.948**	0.980	0.979	0.980
		conti.	0.984	0.984	0.984	0.983	0.983	0.983
3	200	trust.	0.932	0.933	0.935	**0.986**	**0.985**	0.986
		conti.	**0.985**	**0.985**	**0.985**	0.985	0.985	**0.986**

self-repellent complement constraint. See Definition 1. The number of levels $r = 0$ means that the original manifold learning method was applied without being multilevel.

Note that both multilevel Isomap and multilevel SDE use an undirected affinity graph and the same graph coarsening method; therefore the number of vertices at each level is the same in Table 2. both multilevel LLE and multilevel LTSA use an directed affinity graph and the same graph coarsening method; therefore the number of vertices at each level is the same in Table 3. As discussed in Section 4.1, multilevel LE uses the sparsity pattern of the coarse weight matrix to determine the coarse graph. This is different from the other methods.

We use the normalized isometric measure $\mathcal{R}_N$, defined in (41), for multilevel Isomap and multilevel SDE. Both methods rely on the geometric framework in Section 3 with two uncoarsening strategies: greedy refining and alternating refining. Both strategies were tested. From Tables 2 it is clear that the alternating refining improved the result with greedy refining. While multilevel Isomap with alternating isometric refining outperformed Isomap, the advantage of multilevel SDE, compared to SDE, was the computational savings.

The evaluation criteria we used for multilevel LE, multilevel LLE, and multilevel LTSA are trustworthiness and continuity, defined in (44) and (45), respectively. These methods utilize the algebraic framework with three refining strategies: prolongation refining, landmark-based refining, and regularized regression refining. As shown in Table 3, the embedding quality is insensitive to the refining strategy applied in multilevel LLE and multilevel LTSA, which improved the results of LLE and LTSA, respectively. This remark still holds for multilevel LE. Due to the limit of space, we report only the measurements with the regularized regression refining for multilevel LE in Table 2.

Figure 9 illustrates the two-dimensional mappings using Isomap and multilevel Isomap, where we set the number of neighbors per vertex in the kNN graph to $k = 6$ and the degree of dependency to $p = 6$ in the multilevel coarsening, and used the greedy isometric refining method in multilevel uncoarsening. Observe that in these plots, each coordinate axis of the embedding correlates highly with one degree of freedom underlying the original data: left-right pose is correlated with the x axis, and the up-down pose with the y axis.

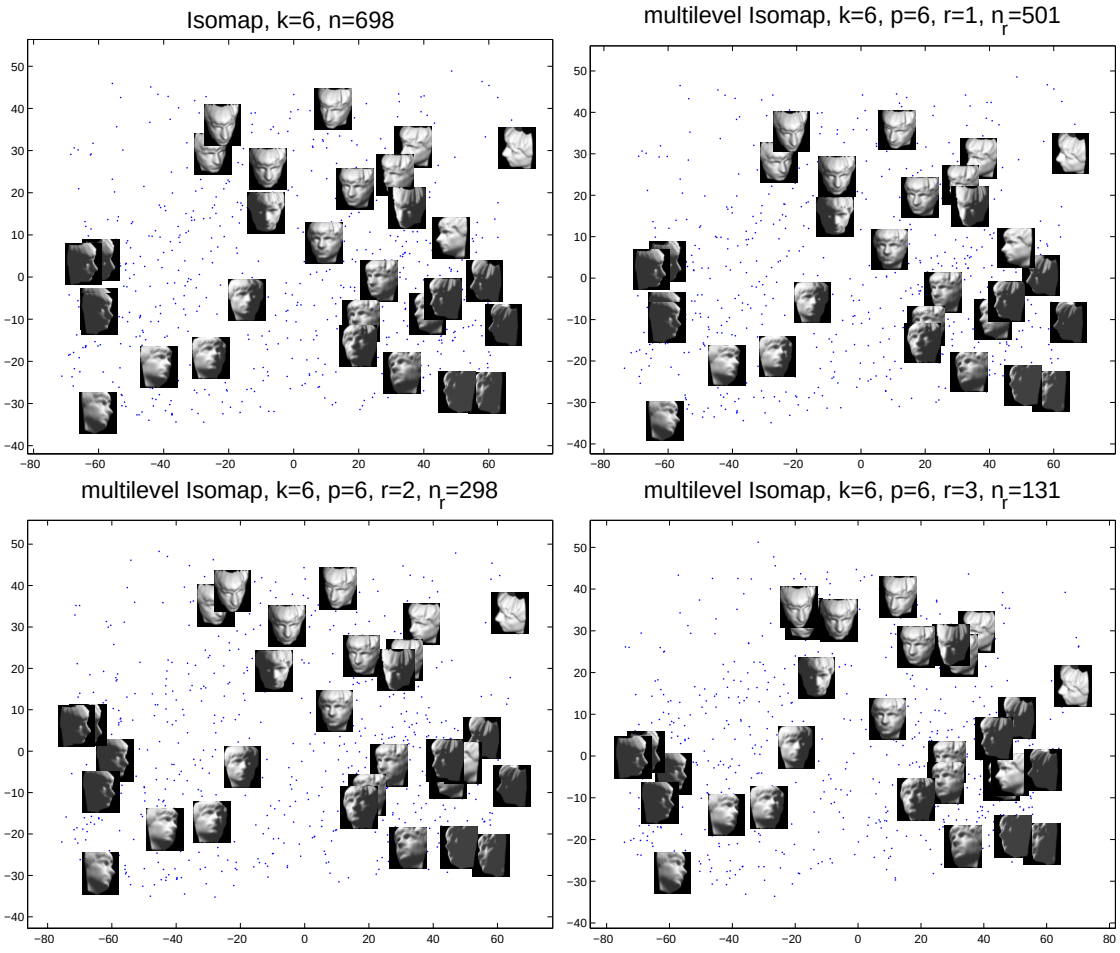


Figure 9: 2D mappings of `Sculpture Face` images by Isomap and multilevel Isomap.

6.3 Frey Face Video Frames

The `Frey Face` data set (Saul and Roweis, 2003) contains 1,965 face images of a single person, Brendan Frey, taken from sequential frames of a small video[3]. Each image is of size 20-by-28 in grayscale, and hence in 560-dimensional space after vectorization.

Table 4: Quality assessments of 2D mappings of `Frey Face` ($k = p = 6$), Part I.

# levels r	# points n_r	Isomap ($\mathcal{R}_N$) greedy refining	Isomap ($\mathcal{R}_N$) alter. refining	SDE ($\mathcal{R}_N$) greedy refining	SDE ($\mathcal{R}_N$) alter. refining	LE # pts. n_r	LE regress. trust.	LE refining conti.
0	1965	0.784	0.783	**0.666**	0.666	1965	0.946	0.981
1	1326	**0.782**	0.676	0.719	0.673	1326	0.948	0.981
2	653	0.796	0.669	0.748	0.671	514	0.951	0.981
3	231	0.875	**0.666**	0.828	**0.665**	83	**0.955**	**0.983**

Table 5: Quality assessments of 2D mappings of `Frey Face` ($k = p = 6$), Part II.

# levels r	# points n_r	eval.	LLE prolong. refining	LLE landmark. refining	LLE regress. refining	LTSA prolong. refining	LTSA landmark. refining	LTSA regress. refining
0	1965	trust.	0.899	0.899	0.899	0.796	0.796	0.796
		conti.	0.964	0.964	0.964	0.927	0.927	0.927
1	1517	trust.	0.900	**0.988**	0.899	0.901	0.896	0.899
		conti.	0.954	0.964	0.964	**0.964**	**0.960**	**0.961**
2	896	trust.	**0.948**	0.948	**0.949**	**0.925**	**0.907**	**0.912**
		conti.	**0.980**	**0.980**	**0.980**	0.964	0.953	0.957
3	441	trust.	0.944	0.945	0.947	0.883	0.868	0.866
		conti.	0.974	0.974	0.975	0.954	0.942	0.947

Our experimental setting is much the same as that for the `Sculpture Face` data set. We used a kNN graph with $k = 6$ and embedding dimension $d = 3$. The quality measurements are reported in Tables 4 and 5, in the same format of Tables 2 and 3, respectively.

The conclusion from Tables 4 and 5, summarized as follows, is also much similar to that of the experiments on the `Sculpture Face` images. For multilevel Isomap and multilevel SDE, the alternating refining method improved the greedy refining method at an extra cost. In addition, multilevel Isomap with alternating isometric refining outperformed Isomap. Compared to SDE, multilevel SDE can hardly generate a better embedding but the computational savings are significant. The multilevel techniques improved LLE, LE and LTSA, no matter whether the uncoarsening strategy prolongation refining, landmark-based refining, or regularized regression refining was applied.

Figure 10 illustrates the two-dimensional mappings of the these images obtained by LTSA and multilevel LTSA, where we set the number of neighbors per vertex in the kNN graph to $k = 20$ and the degree of dependency to $p = 12$ in the multilevel coarsening,

3. `http://cs.nyu.edu/~roweis/data.html`

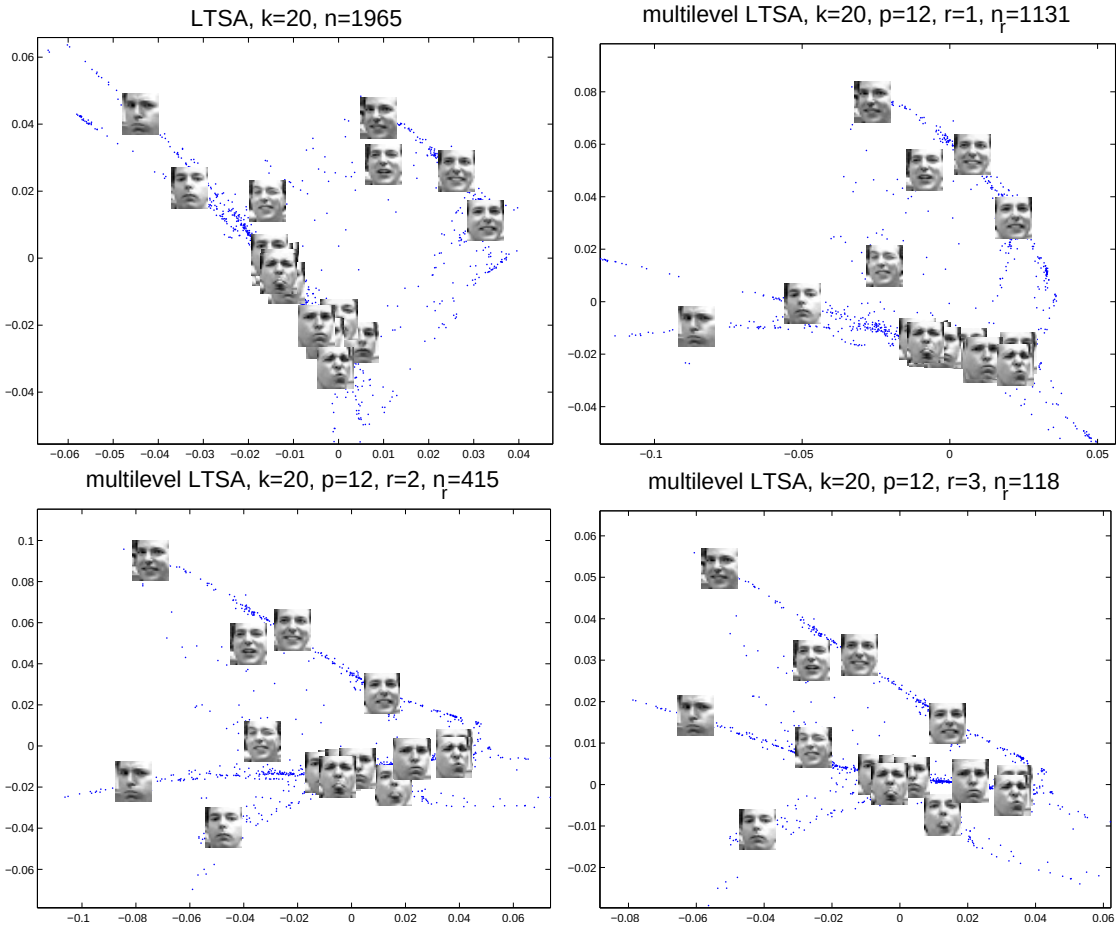


Figure 10: 2D mappings of `Frey Face` images by LTSA and multilevel LTSA.

and used the landmark-based refining method in multilevel uncoarsening. We can observe that all plots exhibits two intrinsic attributes, i.e., pose (left-right) and expression (serious-happy), which are correlated with the coordinate axes.

6.4 Application to Intrinsic Dimension Estimation

Both Isomap and SDE form a kernel matrix K and use a rank-d positive semidefinite matrix K_d to approximate K. The eigen-spectrum of K, or more precisely the 'significant' rank of K, provides a good estimate of the intrinsic dimension (Tenenbaum et al., 2000; Weinberger and Saul, 2006).

We plot $\|K-K_d\|_F/\|K\|_F$ as a function of d, and find the 'elbow' point as an indicator of the intrinsic dimension. When Isomap or SDE is incorporated into the multilevel framework, we have a coarse version of the kernel at the bottom level. By abuse of notation, we also denote this coarse kernel by K. The elbow point of $\|K-K_d\|_F/\|K\|_F$ with the coarse K can still be useful for intrinsic dimension estimation. The proposed method is very inexpensive, since at the bottom level the coarse kernel can be much smaller and therefore cheaper to obtain.

To verify whether this strategy works in practice, we used the `Sculpture Face` and `Frey Face` data sets for experiments. The results are shown in Figures 11 and 12, respectively. In both cases, we used a kNN graph with $k = 6$, and set the degree of dependency $p = 6$ in the multilevel framework.

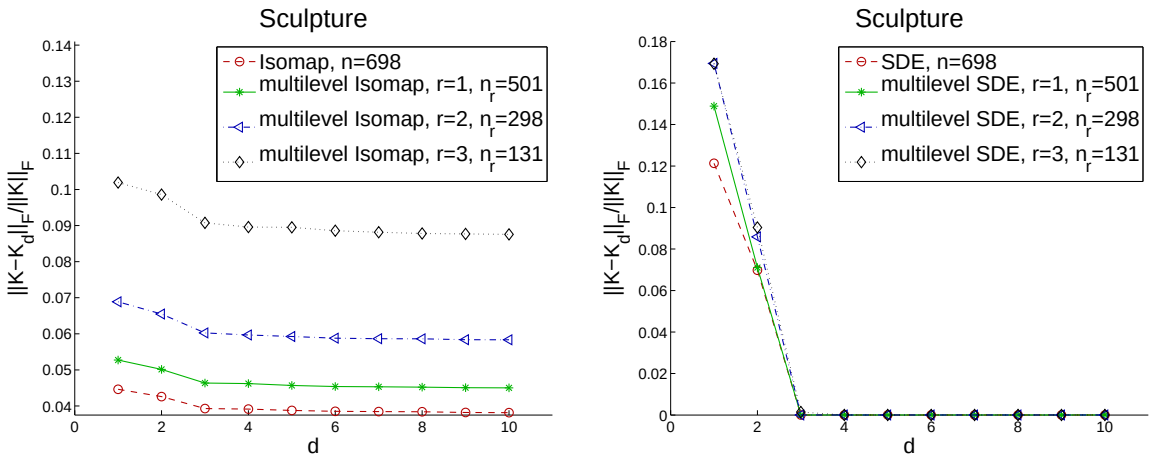


Figure 11: Intrinsic dimension estimation of `Sculpture Face` images.

In both plots of Figure 11, the elbow point is clearly at $d = 3$ in all cases, with the kernel K from Isomap, multilevel Isomap, SDE, or multilevel SDE. This confirms that the intrinsic dimension is 3. Note that the indicator from SDE or multilevel SDE is sharper than that from Isomap or multilevel Isomap. On the other hand, SDE is computationally more expensive because it involves semidefinite programming. The same conclusion can be drawn from Figure 12, except for that the elbow is getting ambiguous when the number of levels reaches $r = 3$.

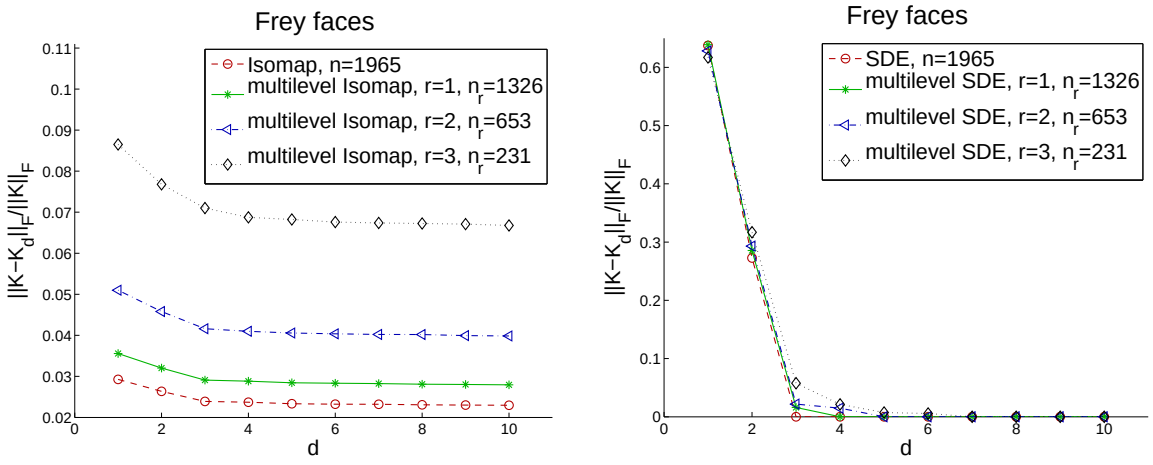


Figure 12: Intrinsic dimension estimation of `Frey Face` images.

7. Conclusion

We present two multilevel frameworks for nonlinear dimensionality reduction; one is geometric and the other is algebraic. The coarsening method can be applied to both directed graphs and undirected graphs. It relies on the dependency between vertices, a parameter that can be used to control the coarsening speed.

The geometric framework propagates the geodesic information in the coarsening phase and allows isometric refining or conformal refining level by level in the uncoarsening phase. It is especially suitable to incorporate the manifold learning algorithms which aim to find an isometric or conformal embedding, such as Isomap, C-Isomap, and SDE.

The uncoarsening phase minimizes isometric or conformal error. It relaxes the convexity assumption in Isomap and the assumption of uniformly distributed samples in C-Isomap to some extent. Our experiments exhibit remarkable improvements by multilevel Isomap and multilevel C-Isomap when these assumptions do not hold.

The algebraic framework is useful for manifold learning algorithms which minimizes a function in the form YMY^T for an embedding Y, subject to certain constraints. The embedding can be obtained from solving a symmetric eigenvalue problem. Examples include LLE, LE, and LTSA. In our experiments, multilevel LLE, multilevel LE, and multilevel LTSA often improved the embedding quality with appropriate parameters.

Acknowledgments

This work was supported by NSF grants DMS-0810938 and DMR-0940218 and by the Minnesota Supercomputing Institute.

Appendix A. Orthogonal Procrustes Problem, PCA, and MDS

We study the orthogonal Procrustes problem in a general form, establish a relation to the principal component analysis (PCA), and show the equivalence between the PCA and the

classical multidimensional scaling (MDS). Applications to isometric analysis and extensions for conformal analysis are also presented. Similar results can be found in the literature (Goldberg and Ritov, 2009; Golub and Van Loan, 1996; Schonemann, 1966; Sibson, 1978; Zhang and Zha, 2004). Here we give a unified view.

A.1 Orthogonal Procrustes Problem

The orthogonal Procrustes problem and its variants have been well studied with wide applications. We consider the general form, called the *Procrustes static* (Goldberg and Ritov, 2009; Sibson, 1978) as follows.

$$\begin{cases} \underset{Q,\gamma}{\text{minimize}} & \|X - QY - \gamma e_n^T\|_F^2 \\ \text{subject to} & Q^TQ = I_d,\ Q \in \mathbb{R}^{m\times d},\ \gamma \in \mathbb{R}^m, \end{cases} \tag{46}$$

where $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ and $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ $(d \le n)$ are input data and $e_n \in \mathbb{R}^n$ is a vector of ones. In practice, it happens often that $d = n$ and/or the term γe_n^T is dropped. At first glance, the goal is to find a matrix Q for rotation and a vector γ for translation such that $x_i \approx Qy_i + \gamma$ for $i = 1, \ldots, n$.

Let $Z = X - QY$. We treat Z as a constant and minimize $\|Z - \gamma e_n^T\|_F^2$ in terms of γ. The objective function can be written as $\mathcal{F}(\gamma) = \operatorname{trace}\left((Z - \gamma e_n^T)^T(Z - \gamma e_n^T)\right)$, which is a strictly convex function. The minimum of $\mathcal{F}$ is reached if and only if its gradient is zero. That is, $\nabla\mathcal{F} = -2(Ze_n - n\gamma) = 0$. Therefore, the minimizer is $\gamma = \frac{1}{n}Ze_n$. Substituting it back to (46), we obtain

$$\begin{cases} \underset{Q}{\text{minimize}} & \|\bar{X} - Q\bar{Y}\|_F^2 \\ \text{subject to} & Q^TQ = I_d,\ Q \in \mathbb{R}^{m\times d}, \end{cases} \tag{47}$$

where $\bar{X} = XJ_n$ and $\bar{Y} = YJ_n$, with $J_n = I_n - \frac{1}{n}e_ne_n^T$ the centering matrix, which rigidly translates the columns of a matrix such that the mean is at the origin, i.e., $\bar{X}e_n = 0$ and $\bar{Y}e_n = 0$.

We rewrite the objective function of program (47) as:

$$\|\bar{X} - Q\bar{Y}\|_F^2 = \operatorname{trace}(\bar{X}^T\bar{X}) + \operatorname{trace}(\bar{Y}^T\bar{Y}) - 2\operatorname{trace}(Q\bar{Y}\bar{X}^T).$$

Since $\operatorname{trace}(\bar{X}^T\bar{X})$ and $\operatorname{trace}(\bar{Y}^T\bar{Y})$ are constant, we maximize $\operatorname{trace}(Q\bar{Y}\bar{X}^T)$. Let $\bar{Y}\bar{X}^T = U\Sigma V^T$ be the singular value decomposition of $\bar{Y}\bar{X}^T$ in reduced form, where $U^TU = I_d$, $V^TV = I_d$, and $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)$. Then we have

$$\operatorname{trace}(Q\bar{Y}\bar{X}^T) = \operatorname{trace}(QU\Sigma V^T) = \operatorname{trace}(V^TQU\Sigma) \le \sum_{i=1}^{d}\sigma_i.$$

Let $W = QU$. Then $W^TW = I_d$, along with $V^TV = I_d$, gives the property that V^TW has all elements bounded by 1, which implies the last inequality.

The maximizer of $\operatorname{trace}(Q\bar{Y}\bar{X}^T)$ is

$$Q = VU^T, \qquad \bar{Y}\bar{X}^T = U\Sigma V^T, \tag{48}$$

which makes $\text{trace}(V^TQU\Sigma) = \sum_{i=1}^{d}\sigma_i$. This $Q = VU^T$ is also the minimizer of (47), no matter whether the columns of $\bar{X}$ and the columns of $\bar{Y}$ are centered at the origin or not. In other words, the solution still works for (46), if the term γe_n^T for translation is dropped.

The orthogonal projection matrix $Q \in \mathbb{R}^{d\times m}$ and the translation vector $\gamma \in \mathbb{R}^m$ which form the minimizer of (46) can be used to map the out-of-sample data. This property is utilized in RML (Lin and Zha, 2008) and GP (Goldberg and Ritov, 2009). To be specific, given an additional $y \in \mathbb{R}^d$, we can map it to $x = Qy + \gamma$. Given an additional $x \in \mathbb{R}^m$, we can map it to $y = Q^T(x-\gamma)$, which minimizes $\|x-Qy-\gamma\|$, where Q^T is the Moore-Penrose pseudo-inverse of Q because $Q^TQ = I_d$. Instead, one may use (47) to remove the freedom of translation.

A.2 Relation to PCA

A relation of (46) to the principal component analysis (PCA) can be established as follows. We relax $Y \in \mathbb{R}^{d\times n}$ in (46) and consider:

$$\begin{cases} \underset{Y,Q,\gamma}{\text{minimize}} & \|X - QY - \gamma e_n^T\|_F^2 \\ \text{subject to} & Y \in \mathbb{R}^{d\times n},\ Q^TQ = I_d,\ Q \in \mathbb{R}^{m\times d},\ \gamma \in \mathbb{R}^m. \end{cases} \tag{49}$$

We assume $d < n$ or the problem is trivial. In a similar vein to (47), we can get rid of the term γe_n^T in (49) and rewrite the objective function as $\|\bar{X} - QYJ_n\|_F^2$, where $\bar{X} = XJ_n$ and $J_n = I_n - \frac{1}{n}e_ne_n^T$. The difference is that now Y is a variable. Hence the only constraint to QY is that the rank is at most d. This is a standard low rank approximation problem except for the factor J_n in $\|\bar{X} - QYJ_n\|_F^2$. However, this factor J_n has no effect in the low rank approximation, since $\bar{X}e_n = 0$ and J_n is a projector which deflates only e_n.

To be precise, we minimize $\|\bar{X} - A\|$ subject to $\text{rank}(A) \le d$, where $\bar{X}e_n = 0$. The minimizer is $A = U_d\Sigma_dV_d^T$, where $U_d\Sigma_dV_d^T$ is the rank-d truncated SVD of A (Golub and Van Loan, 1996). This $A = U_d\Sigma_dV_d^T$ is also the minimizer of $\|\bar{X} - AJ_n\|$ under the same constraint $\text{rank}(A) \le d$, since $\bar{X}e_n = 0$ implies $U_d\Sigma_dV_d^TJ_n = U_d\Sigma_dV_d^T$. Note that here A plays the role of QY in (49). Hence $Q = U_d$ and $Y = \Sigma_dV_d^T$, along with $\gamma = \frac{1}{n}(X - U_d\Sigma_dV_d^T)e_n$, constitute a minimizer of (49), where $U_d\Sigma_dV_d^T$ is the rank-d truncated SVD of XJ_n.

The solution to (49) is equivalent to applying PCA to $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ for a low dimensional embedding $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$. To be specific, PCA projects $x_i \in \mathbb{R}^m$ to $y_i \in \mathbb{R}^d$ by $y_i = Q^Tx_i$ for $i = 1, \ldots, n$, where $Q^TQ = I_d$. The goal is to maximize the variance of $y_1, \ldots, y_n$:

$$\sum_{i=1}^{n}\|y_i - \frac{1}{n}\sum_{j=1}^{n}y_j\|^2 = \sum_{i=1}^{n}\|y_i - \frac{1}{n}Ye_n\|^2 = \|Y - \frac{1}{n}Ye_ne_n^T\|_F^2 = \|Q^TX(I_n - \frac{1}{n}e_ne_n^T)\|_F^2, \tag{50}$$

where $X(I_n - \frac{1}{n}e_ne_n^T) = XJ_n = \bar{X}$. Denote by $U_d\Sigma_dV_d^T$ the rank-d truncated SVD of $\bar{X}$. The minimizer of (50) is $Q = U_d$, yielding $Y = U_d^TX = \Sigma_dV_d^T$. The equivalence is clear.

A.3 Equivalence between PCA and MDS

PCA projects given $x_1, \ldots, x_n \in \mathbb{R}^m$ to $y_1, \ldots, y_n \in \mathbb{R}^d$. On the other hand, MDS takes a set of dissimilarities δ_{ij} and returns a set of points $y_1, \ldots, y_n \in \mathbb{R}^d$ such that $\|y_i - y_j\| \approx \delta_{ij}$ for

$i, j = 1, \ldots, n$. The two methods are seemingly different but essentially equivalent (Gower, 1966). The discussion is as follows.

Let $D \in \mathbb{R}^{n\times n}$ be the matrix whose (i, j) entry is δ_{ij}^2. Image that there are points $x_1, \ldots, x_n$ in a high dimensional space such that $\delta_{ij} = \|x_i - x_j\|$. Let $B = -\frac{1}{2}J_n D J_n$. Then with some algebra, it can be verified that the (i, j) entry of B is $(x_i - \bar{x})^T(x_j - \bar{x})$, where $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$ is the mean of $x_1, \ldots, x_n$. Using the notation in Appendix A.2, we have $B = \bar{X}^T\bar{X}$. The embedding $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ is obtained from minimizing $\|B - Y^T Y\|_F^2$. That is, $Y = \Lambda_d^{1/2} V_d^T$, where $\Lambda_d \in \mathbb{R}^{d\times d}$ is a diagonal matrix formed with the d largest eigenvalues of B, and the columns of $V_d \in \mathbb{R}^{n\times d}$ are the corresponding eigenvectors.

Since $B = \bar{X}^T\bar{X}$, the eigenvectors of B is the same as the right singular vectors of $\bar{X}$, and the eigenvalues of B is the squared singular values of $\bar{X}$. Therefore, MDS is equivalent to PCA, when each dissimilarity d_{ij} in MDS is the same as the distance $\|x_i - x_j\|$ between points x_i and x_j in PCA.

A.4 Application to Isometric Analysis

Consider a mapping $f : \Omega \to \mathcal{M}$ between two manifolds $\Omega \subset \mathbb{R}^d$ and $\mathcal{M} \subset \mathbb{R}^m$, where Ω is open. Assume that we are given a small neighborhood $\mathcal{N} \subset \Omega$ containing some point $\bar{y} \in \mathcal{N}$. Suppose there are sample points $y_1, \ldots, y_n \in \mathcal{N}$ and $x_1, \ldots, x_n \in f(\mathcal{N})$, such that $x_i = f(y_i)$ for $i = 1, \ldots, n$. The Taylor series gives $f(y_i) = f(\bar{y}) + J_f(\bar{y})(y_i - \bar{y}) + O(\|y_i - \bar{y}\|^2)$, where $J_f(\bar{y})$ is the Jacobian. Substituting x_i for $f(y_i)$, we have

$$\|x_i - J_f(\bar{y})y_i - \gamma\| = O(\|y_i - \bar{y}\|^2), \tag{51}$$

where $\gamma = f(\bar{y}) - J_f(\bar{y})\bar{y}$. Aggregating (51) into one matrix form, we obtain

$$\|X - J_f(\bar{y})Y - \gamma e_n^T\|_F / \|Y - \bar{y}e_n^T\|_F = O(\|Y - \bar{y}e_n^T\|_F), \tag{52}$$

where $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ and $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$. When $\int_{\mathcal{N}} dy \to 0$, the right-hand side of (52) is driven to 0, and therefore so is the left-hand side. This changes the goal into one of minimizing $\|X - J_f(\bar{y})Y - \gamma e_n^T\|_F$.

The squared factor $\|X - J_f(\bar{y})Y - \gamma e_n^T\|_F^2$ matches the objective function of (46) and (49), where $Q = J_f(\bar{y})$. Assuming that f is an isometry, $J_f(\bar{y})^T J_f(\bar{y}) = I_d$, which is the constraint of both (46) and (49). There are three scenarios. First, if both X and Y are known, then we can use (46) to estimate the Jacobian $J_f(\bar{y})$. Second, if only X is known, then we use (49) to estimate Y and $J_f(\bar{y})$. Computationally, this is equivalent to PCA as shown in Appendix A.2. Third, if we do not know any of X and Y, but instead we know $\|x_i - x_j\|$ for $i, j = 1, \ldots, n$, then we can still use MDS to estimate Y, via the equivalences discussed in Appendices A.2 and A.3. Even better, if $\mathcal{N}$ is convex and we have the approximate geodesic distance between x_i and x_j for $i, j = 1, \ldots, n$, then we can exploit (5) and use Isomap to estimate Y. Since the data points in the computation are in a neighborhood instead of the global manifold, we call them local PCA, local MDS, and local Isomap, respectively.

A remark concerning the second and third cases is as follows. Consider (52). Since

$$X - J_f(\bar{y})Y - \gamma e_n^T = X - (J_f(\bar{y})P^T)(PY + \xi e_n^T) - (\gamma - J_f(\bar{y})P^T\xi)e_n^T$$

for any orthogonal matrix $P \in \mathbb{R}^{d\times d}$ and any vector $\xi \in \mathbb{R}^d$. It means that the estimated $y_1, \ldots, y_n$ are subject to rotation and translation.

A.5 Extension for Conformal Analysis

To measure the satisfaction of angle preservation, we add a scalar $c \geq 0$ to (46) and have

$$\begin{cases} \underset{c,Q,\gamma}{\text{minimize}} & \|X - c\,QY - \gamma e_n^T\|_F^2 \\ \text{subject to} & c \geq 0,\ Q^TQ = I_d,\ Q \in \mathbb{R}^{m\times d},\ \gamma \in \mathbb{R}^m, \end{cases} \tag{53}$$

where the newly added $c \geq 0$ corresponds to the scale function $c(y)$ of a conformal mapping in (6). In a similar discussion leading to (47), we transform the program (53) into:

$$\begin{cases} \underset{c,Q}{\text{minimize}} & \|\bar{X} - c\,Q\bar{Y}\|_F^2 \\ \text{subject to} & c \geq 0,\ Q^TQ = I_d,\ Q \in \mathbb{R}^{m\times d}, \end{cases} \tag{54}$$

where $\bar{X} = XJ_n$ and $\bar{Y} = YJ_n$, with $J_n = I_n - \frac{1}{n}e_ne_n^T$ the centering matrix.

With a little thought, as long as $c \geq 0$, (48) still gives an optimal $Q = VU^T$ for (54), where U and V come from the singular value decomposition $\bar{Y}\bar{X}^T = U\Sigma V^T$, independent of the scale factor c. The objective function of (54) can be written as:

$$\operatorname{trace}((\bar{X} - c\,Q\bar{Y})^T(\bar{X} - c\,Q\bar{Y})) = \operatorname{trace}(\bar{X}^T\bar{X}) - 2c\operatorname{trace}(Q\bar{Y}\bar{X}^T) + c^2\operatorname{trace}(\bar{Y}^T\bar{Y}), \tag{55}$$

which is a quadratic function in terms of c. The minimum of (55) is reached when $c = \operatorname{trace}(Q\bar{Y}\bar{X}^T)/\operatorname{trace}(\bar{Y}^T\bar{Y})$. Substituting $\bar{Y}\bar{X}^T = U\Sigma V^T$ and the optimal $Q = VU^T$, we obtain

$$c = \operatorname{trace}(VU^TU\Sigma V^T)/\operatorname{trace}(\bar{Y}^T\bar{Y}) = \operatorname{trace}(\Sigma)/\|\bar{Y}\|_F^2, \tag{56}$$

which is naturally nonnegative. We conclude that (48) and (56) constitute the minimizer of (54). Similar discussions can be found in Goldberg and Ritov (2009) and Sibson (1978).

Note that like (47), the program (54) is it own place, and the minimizer does not rely on the property $\bar{X}e_n = 0$ and $\bar{Y}e_n = 0$. In other words, the solution still works for (53), if the last term of the objective function γe_n^T is dropped.

If we relax $Y \in \mathbb{R}^{d\times n}$ in (53), then the free factor $c \geq 0$ can be incorporated into Y, and hence the solution is equivalent to the PCA as discussed in Appendix A.2. The interpretation of how (53) and (54) are related to a conformal mapping is similar to that for an isometry discussed in Appendix A.4. The details are omitted.

Appendix B. Manifold Learning Algorithms

This appendix reviews 5 manifold learning algorithms, namely Isomap, LLE, LE, LTSA, and SDE. All these methods use an affinity graph to model the neighborhood of each sample point. In practice, it is typically (but not limited to) a kNN graph. See Section 2.2.1 for a discussion on affinity graphs.

In what follows, it is assumed that an affinity graph $G = (V, E)$ has been obtained, where the vertex set $V = \{1, \ldots, n\}$ consists of indices of points, and $(i, j) \in E$ if vertex j is a neighbor of vertex i. The graph is either directed or undirected, depending on the manifold learning method.

B.1 Isomap

Isomap (Tenenbaum et al., 2000) is a nonlinear generalization of the classical multidimensional scaling (MDS). It replaces the Euclidean distances in MDS by the *geodesic* distances approximated by an affinity graph. The length of the shortest path between two points in the graph is the approximate geodesic distance between them. The algorithm can be summarized in the following steps.

1. Construct an undirected affinity graph $G = (V, E)$ of the input data $x_1, \ldots, x_n \in \mathbb{R}^m$, where the edge length $\widetilde{\delta}_{ij} = \|x_i - x_j\|$ for $(i, j) \in E$. With this, the all-pair shortest path problem is solved and all the squared approximate geodesic distances $\widetilde{\delta}_{ij}^2$ are saved in a symmetric matrix $\widetilde{D} \in \mathbb{R}^{n\times n}$.

2. Compute the Grammian matrix $\widetilde{B} = -\frac{1}{2} J_n \widetilde{D} J_n \in \mathbb{R}^{n\times n}$, where $J_n = I - \frac{1}{n} e_n e_n^T \in \mathbb{R}^{n\times n}$ with $I_n \in \mathbb{R}^{n\times n}$ the identity matrix and $e_n \in \mathbb{R}^n$ a column vector of ones.

3. Then Isomap maps $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ nonlinearly to $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ by minimizing $\|\widetilde{B} - Y^T Y\|_F$. To be specific, we compute $Y = \Sigma_d^{1/2} V_d^T$, where $\Sigma_d \in \mathbb{R}^{d\times d}$ is the diagonal matrix consisting of the d largest eigenvalues of $\widetilde{B}$, and the columns of $V_d \in \mathbb{R}^{n\times d}$ are the corresponding eigenvectors. This $Y = \Sigma_d^{1/2} V_d^T \in \mathbb{R}^{d\times n}$ minimizes $\|\widetilde{B} - Y^T Y\|_F$.

The relation between the metric MDS and Isomap is worth noting. The metric MDS uses a distance matrix D whose (i, j) entry is $\|x_i - x_j\|^2$. Without loss of generality, we assume the inputs are translated so that the centroid is at origin, i.e., $\sum_{i=1}^n x_i = 0$. Then the (i, j) entry of the Grammian matrix $B = -\frac{1}{2} JDJ$ is $x_i^T x_j$. The linear mapping $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ is obtained from minimizing $\|B - Y^T Y\|_F$. On the other hand, Isomap minimizes $\|\widetilde{B} - Y^T Y\|_F$ for the low dimensional embedding $Y \in \mathbb{R}^{d\times n}$, where $\widetilde{B} = -\frac{1}{2} J\widetilde{D}J$, with $\widetilde{D}$ formed by the squared approximate geodesic distances rather than the squared Euclidean distances in MDS.

B.2 Locally Linear Embedding

Locally linear embedding (LLE) (Roweis and Saul, 2000; Saul and Roweis, 2003) maps the high dimensional input data $x_1, \ldots, x_n \in \mathbb{R}^m$ to $y_1, \ldots, y_n \in \mathbb{R}^d$ in a lower dimensional space (i.e., $d < n$) by three steps.

1. Construct an affinity graph of the input data $x_1, \ldots, x_n$. This graph can be directed.

2. The reconstruction weights in $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ are obtained by minimizing the cost function:
$$\mathcal{E}(W) = \sum_{i=1}^n \|x_i - \sum_{j=1}^n w_{ij} x_j\|^2, \tag{57}$$
subject to that $w_{ij} = 0$ if x_j is not one of k nearest neighbors of x_i, and $\sum_{j=1}^n w_{ij} = 1$ for $i = 1, \ldots, n$. Minimizing $\|x_i - \sum_{j=1}^n w_{ij} x_j\|^2$ in (57) requires solving a constrained least squares problem for each $i = 1, \ldots, n$.

3. The low dimensional data $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ is formed by the d right singular vectors $v_2, \ldots, v_{d+1}$ of $(I_n - W)$ corresponding to the second to the $(d+1)$st smallest singular values, i.e., $Y = [v_2, \ldots, v_{d+1}]^T$, where $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix.

The last step minimizes the embedding cost function:

$$\Phi(Y) = \sum_{i=1}^{n} \|y_i - \sum_{j=1}^{n} w_{ij} y_j\|^2 = \|Y - YW^T\|_F^2 = \mathrm{trace}[Y(I_n - W)^T(I_n - W)Y^T],$$

where two constraints are imposed, namely $\sum_{i=1}^n y_i y_i^T = YY^T = I_n$ and $\sum_{i=1}^n y_i = Ye_n = 0$, with $e_n \in \mathbb{R}^n$ the column vector of ones. The program can be written as:

$$\begin{cases} \underset{Y\in\mathbb{R}^{d\times n}}{\text{minimize}} & \mathrm{trace}(Y(I_n - W)^T(I_n - W)Y^T) \\ \text{subject to} & YY^T = I_n,\ Ye_n = 0. \end{cases} \tag{58}$$

Then the problem is transformed to computing the d eigenvectors $v_2, ..., v_{d+1}$ of $(I_n - W)^T(I_n - W)$ corresponding to the second to the $(d+1)$st smallest eigenvalues. The minimizer of (58) is $Y = [v_2, \ldots, v_{d+1}]^T$. The condition $Ye_n = 0$ drops the bottom eigenvector e_n which cannot be used to discriminate the embedded points $y_1, \ldots, y_n$. Note that the eigenvalues of $(I_n - W)^T(I_n - W)$ are the singular values of $(I_n - W)$, and the eigenvectors of $(I_n - W)^T(I_n - W)$ are the right singular vectors of $(I_n - W)$.

B.3 Laplacian Eigenmaps

In Laplacian Eigenmaps (Belkin and Niyogi, 2001, 2003), an affinity graph $G = (V, E)$ of the input data $x_1, \ldots, x_n \in \mathbb{R}^m$ is constructed. The graph is undirected, since the weighting scheme is a radical basis function. To obtain the low dimensional embedding $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$, we minimize the cost function:

$$\Psi(Y) = \sum_{i,j} w_{ij}\|y_i - y_j\|^2 = 2\,\mathrm{trace}(Y(D - W)Y^T) = 2\,\mathrm{trace}(YLY^T), \tag{59}$$

where $W = [w_{ij}]$ is a symmetric weight matrix, and D is a diagonal matrix with $d_{ii} = \sum_{j=1}^n w_{ij}$, and $L = D - W$ is the Laplacian matrix.

A popular weighting scheme is the Gaussian weights:

$$w_{ij} = \exp(-\|x_i - x_j\|^2/t) \tag{60}$$

for $(i, j) \in E$ and otherwise $w_{ij} = 0$, where $t > 0$ is a preset parameter. This weighting scheme is also called *heat kernel*. In our experiments we set t equal to the median of $\|x_i - x_j\|^2$ of all $(i, j) \in E$. Driving $\sigma \to \infty$ in (60), we obtain the induced binary weights.

To make the minimization of (59) well-posed, the constraints $YDY^T = I_n$ and $YDe_n = 0$ are imposed, where $e_n \in \mathbb{R}^n$ is the column vector of ones. The resulting program is:

$$\begin{cases} \underset{Y\in\mathbb{R}^{d\times n}}{\text{minimize}} & \mathrm{trace}(YLY^T) \\ \text{subject to} & YDY^T = I_n,\ YDe_n = 0. \end{cases} \tag{61}$$

The minimizer of (61) can be obtained from solving the generalized eigenvalue problem $Lz = \lambda Dz$. The low dimensional embedding is formed by the d generalized eigenvectors $v_2, ..., v_{d+1}$ corresponds to the second to the $(d+1)$st smallest eigenvalues, i.e., $Y = [v_2, \ldots, v_{d+1}]^T$, which minimizes (61). The condition $YDe_n = 0$ drops the bottom generalized eigenvector e_n, which has no discrimination power.

B.4 Local Tangent Space Alignment

The method of local tangent space alignment (LTSA) (Zhang and Zha, 2004) maps given $x_1, \ldots, x_n \in \mathbb{R}^m$ to $y_1, \ldots, y_n \in \mathbb{R}^d$ by the following steps.

1. Construct an affinity graph $G = (V, E)$ of the input data $x_1, \ldots, x_n$. This graph can be directed. Each vertex i is associated with a neighborhood $\mathcal{N}_i = \{j : (i, j) \in E\} \cup \{i\}$ including the vertex i itself. We denote $n_i = |\mathcal{N}_i|$, the size of the neighborhood $\mathcal{N}_i$.

2. For $i = 1, \ldots, n$, perform PCA on $X_i = [x_j]_{j \in \mathcal{N}_i}$ to obtain the low dimensional local coordinates $\Theta_i = [\theta_j^{(i)}]_{j \in \mathcal{N}_i}$. Here $\Theta_i \in \mathbb{R}^{d \times n_i}$ is the optimal estimate of the local isometric embedding. See Appendix A.4.

3. Let $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d \times n}$ be the low dimensional global coordinates to compute, and denote $Y_i = [y_j]_{j \in \mathcal{N}_i}$. It is assumed that for $i = 1, \ldots, n$, there is a $L_i \in \mathbb{R}^{d \times d}$ which is related to the Jacobian of the manifold mapping $x = f(y)$ at $y = y_i$, such that $Y_i J_{n_i} \approx L_i \Theta_i$, where $J_{n_i} = I_{n_i} - \frac{1}{n_i} e_{n_i} e_{n_i}^T$ is the centering matrix. The reason for centering is discussed in Appendix A.1. Note that we do not have to center Θ_i, since Θ_i comes from PCA and therefore $\Theta_i e_{n_i} = 0$ and $\Theta_i J_{n_i} = \Theta_i$.

4. The goal of LTSA is to minimize the sum of $\|Y_i J_{n_i} - L_i \Theta_i\|_F^2$ for $i = 1, \ldots, n$. When the minimum is reached, $L_i = Y_i J_{n_i} \Theta_i^T = Y_i \Theta_i^T$, where Θ_i^T is the Moore-Penrose pseudo-inverse of Θ_i because $\Theta_i^T \Theta_i = I_d$, and $J_{n_i} \Theta_i^T = \Theta_i^T$ comes from the property $\Theta_i e_{n_i} = 0$. Let $Y_i = Y S_i$, where $S_i \in \{0, 1\}^{n \times n_i}$ is the boolean selection matrix. Substituting $Y_i = Y S_i$ and $L_i = Y_i \Theta_i^T$, we obtain

$$\|Y_i J_{n_i} - L_i \Theta_i\|_F^2 = \|Y S_i J_{n_i} (I_{n_i} - \Theta_i^T \Theta_i)\|_F^2 = \|Y S_i H_i\|_F^2,$$

   where $H_i = J_{n_i}(I_{n_i} - \Theta_i^T \Theta_i) = I_{n_i} - [\frac{e_{n_i}}{\sqrt{n_i}}, \Theta_i]^T [\frac{e_{n_i}}{\sqrt{n_i}}, \Theta_i]$. Let $S = [S_1, \ldots, S_n]$ and $H = \mathrm{diag}(H_1, \ldots, H_n)$. The objective function to minimize is:

$$\sum_{i=1}^{n} \|Y_i J_{n_i} - L_i \Theta_i\|_F^2 = \sum_{i=1}^{n} \|Y S_i H_i\|_F^2 = \|YSH\|_F^2 = \mathrm{trace}(YSHH^T S^T Y^T).$$

   Hence we consider the program:

$$\begin{cases} \underset{Y \in \mathbb{R}^{d \times n}}{\text{minimize}} & \mathrm{trace}(Y(SHH^T S^T) Y^T) \\ \text{subject to} & YY^T = I_n,\ Ye_n = 0. \end{cases} \tag{62}$$

   The constraint $YY^T = I_n$ is to make the problem well-posed. The minimizer of (62) consists of bottom eigenvectors. It turns out that e_n is the bottom eigenvector of $SHH^T S^T$, since $e_n^T S_i H_i = e_{n_i}^T H_i = 0$ for $i = 1, \ldots, n$. Hence we add the

constraint $Ye_n = 0$ to remove e_n in the embedding. (62) is equivalent to a symmetric eigenvalue problem with the bottom eigenvector e_n, which is removed from the embedding by $Ye_n = 0$. The low dimensional embedding, the minimizer of (62), is $Y = [v_2, \ldots, v_{d+1}]^T$, where $v_2, ..., v_{d+1}$ are the d eigenvectors of SHH^TS^T, corresponding to the second to the $(d+1)$st smallest eigenvalues.

B.5 Semidefinite Embedding

Given input data $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$, semidefinite embedding (SDE) (Weinberger and Saul, 2004, 2006) consists of three steps for an embedding $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$:

1. Construct an undirected affinity graph $G = (V, E)$ of the input data $x_1, \ldots, x_n \in \mathbb{R}^m$.

2. Build a kernel $K = [k_{ij}] \in \mathbb{R}^{n\times n}$, such that $k_{ij} = z_i^T z_j$, where $z_1, \ldots, z_n$ is some 'conceptual' embedding which preserves the local distances $\|z_i - z_j\| = \|x_i - x_j\|$ for $(i, j) \in E$. At the same time, we maximize the variance $\sum_{i=1}^n \|z_i\|^2$, under the assumption that centroid of $z_1, \ldots, z_n$ is at origin, i.e., $\sum_{i=1}^n z_i = 0$. At the end, we solve the semidefinite program (Vandenberghe and Boyd, 1996):

$$\begin{cases} \underset{K\in\mathbb{R}^{n\times n}}{\text{maximize}} & \text{trace}(K) \\ \text{subject to} & K \succeq 0,\ e_n^T K e_n = 0; \\ & k_{ii} - 2k_{ij} + k_{jj} = \|x_i - x_j\|^2 \text{ for } (i, j) \in E. \end{cases} \tag{63}$$

3. With the minimizer K of (63), we obtain the embedding $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ from minimizing $\|K - Y^TY\|_F$. In this respect, $y_1, \ldots, y_n \in \mathbb{R}^d$ constitute the best d-dimensional approximation of the conceptual $z_1, \ldots, z_n$. The solution is a symmetric eigenvalue problem. More precisely, the embedding is $Y = \Lambda_d^{1/2} V_d^T$, where $\Lambda_d \in \mathbb{R}^{d\times d}$ is a diagonal matrix formed with the d largest eigenvalues of B, and the columns of $V_d \in \mathbb{R}^{n\times d}$ are the corresponding eigenvectors.

A few remarks on the semidefinite program (63) are as follows. The positive semidefiniteness $K \succeq 0$ is because $K \in \mathbb{R}^{n\times n}$ is the kernel matrix of the conceptual $z_1, \ldots, z_n$. Since it is assumed that $\sum_{i=1}^n z_i = 0$, we have $Ke_n = 0$, which is equivalent to $e_n^T K e_n = 0$ due to $K \succeq 0$. The constraints $k_{ii} - 2k_{ij} + k_{jj} = \|x_i - x_j\|^2$ for $(i, j) \in E$ come from the standard 'kernel trick' to preserve the local distances. The variance $\sum_{i=1}^n \|z_i\|^2$ can be presented with the kernel K, as trace(K). As a result, the program (63) can be written in terms of a semidefinite kernel $K \succeq 0$. An interesting property is that $\sum_{i,j=1}^n \|z_i - z_j\|^2 = 2n \sum_{i=1}^n \|z_i\|^2$. Hence maximizing $\sum_{i=1}^n \|z_i\|^2$ is equivalent to maximizing $\sum_{i,j=1}^n \|z_i - z_j\|^2$.

Note that in practice, if the distances are inaccurate, then it is unrealistic to expect that the program (63) is feasible. Therefore, we can replace the equality constraint $k_{ii} - 2k_{ij} + k_{jj} = \|x_i - x_j\|^2$ by the inequality constraint $k_{ii} - 2k_{ij} + k_{jj} \leq \|x_i - x_j\|^2$ for $(i, j) \in E$. It is important in the geometric multilevel framework presented in Section 3, where the propagated geodesic information contains approximation errors.

References

J. Bak and D. J. Newman. *Complex Analysis*. Springer-Verlag, 1st edition, 1982.

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, 2001.

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

Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1–4):613–623, 1999.

J. Chen, H.-r. Fang, and Y. Saad. Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection. *J. Mach. Learn. Res.*, 10:1989–2012, 2009.

R. R. Coifman and S. Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21:5–30, 2006.

V. de Silva and J. B. Tenenbaum. Unsupervised learning of curved manifolds. In *Nonlinear Estimation and Classification, Lecture Notes in Statistics*. Springer-Verlag, New York, NY, 2003a.

V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, Cambridge, MA, 2003b.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. URL `http://jmvidal.cse.sc.edu/library/dijkstra59a.pdf`.

M. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1st edition, 1976.

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

H.-r. Fang, S. Sakellaridi, and Y. Saad. Multilevel manifold learning with application to spectral clustering. In *The 19th ACM Conference on Information and Knowledge Management (CIKM)*, pages 419–428, 2010.

Y. Goldberg and Y. Ritov. Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms. *Mach. Learn.*, 77(1):1–25, 2009.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3/4):325–338, December 1966.

J. H. Ham, D. D. Lee, and L. K. Saul. Semisupervised alignment of manifolds. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 120–127, 2005.

G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48(1):96–129, 1998.

G. Karypis and V. Kumar. Multilevel k-way hypergraph partitioning. *VLSI Design*, 11(3): 285–300, 2000.

S. Lafon and A. B. Lee. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1393–1403, 2006.

J. A. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7–9):1431–1443, 2009.

T. Lin and H. Zha. Riemannian manifold learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):796–809, 2008.

B. O'Neill. *Elementary Differential Geometry*. Academic Press, revised 2nd edition, 2006.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Y. Saad. SPARSKIT: a basic tool kit for sparse matrix computations - version 2, 1994.

Y. Saad. *Iterative methods for sparse linear systems*. SIAM Publications, Philadelphia, PA, 2nd edition, 2003.

L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J. Mach. Learn. Res.*, 4:119–155, 2003.

L. K. Saul, K. Q. Weinberger, J. H. Ham, F. Sha, and D. D. Lee. Spectral methods for dimensionality reduction. In *Semisupervised Learning*. MIT Press, Cambridge, MA, 2006.

P. H. Schonemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

F. Sha and L. K. Saul. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *The 22nd international conference on Machine learning (ICML)*, pages 784–791, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5.

B. Shaw and T. Jebara. Minimum volume embedding. In *The 11th Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 2, pages 460–467, 2007.

R. Sibson. Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society*, 40(1):234–238, 1978.

A. Talwalkar, S. Kumar, and H. A. Rowley. Large-scale manifold learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

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

U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 1st edition, 2000.

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

J. Venna and S. Kaski. Local multidimensional scaling. *Neural Networks*, 19(6–7):889–899, 2006.

F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *ICML*, pages 985–992, 2006.

F. Wang and C. Zhang. Fast multilevel transduction on graphs. In *The 7th SIAM Conference on Data Mining (SDM)*, pages 157–168, 2007.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 988–995, 2004.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal on Computer Vision*, 70(1):77–90, 2006.

Q. Weinberger, B. D. Packer, and L. K. Saul. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *The 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 381–388, 2005.

H. Zha and Z. Zhang. Continuum isomap for manifold learnings. *Computational Statistics & Data Analysis*, 52:184–200, 2006.

Z. Zhang and H. Zha. Principal manifolds and nonlinear dimension reduction via tangent space alignment. *SIAM J. Sci. Comput.*, 26(1):313–338, 2004.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *The 20th International Conference on Machine Learning (ICML)*, pages 912–919, 2003.