Multilevel Nonlinear Dimensionality Reduction for Manifold Learning*

Haw-ren Fang† Sophia Sakellaridi‡ Yousef Saad‡

September 14, 2009

Abstract

Nonlinear dimensionality reduction techniques for manifold learning, e.g., Isomap, may become exceedingly expensive to carry out for large data sets. This paper explores a multilevel framework with the goal of reducing the cost of unsupervised manifold learning. In addition to savings in computational time, the proposed multilevel technique essentially preserves the geodesic information, and so it can potentially improve on some manifold learning methods which do not preserve isometry. An application to K-means clustering is also presented. Experimental results indicate that the multilevel approach can be an appealing alternative to standard techniques.

Keywords: Manifold learning, multilevel techniques, nonlinear dimensionality reduction, nearest-neighbor graph, eigenvalue problem

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 dimensions is the size of the number of pixels per image, or generally the number of numerical values per data item, and can be characterized by far fewer features.

Recently 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 represented by vectors $x_1, \ldots, x_n$ in $\mathbb{R}^m$, the task is to represent these with low dimensional vectors $y_1, \ldots, y_n \in \mathbb{R}^d$ with $d \ll m$, such that nearby points remain nearby, and distant points remain distant. Linear methods of dimensionality reduction, such as the classical Principal Component Analysis (PCA) and metric Multi-Dimensional Scaling (MDS), can become

*This work was supported by NSF grant DMS-0810938 and by the Minnesota Supercomputing Institute.

†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA. Email: hrfang@mcs.anl.gov

‡Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA. Email: {sakell,saad}@cs.umn.edu

inadequate because the meaningful low dimensional structure extracted from high dimensional data is often nonlinear. Therefore, considerable research effort has been devoted to the development of effective nonlinear methods to discover 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, e.g., [10, 11]. On the other hand, their incorporation into dimensionality reduction methods is currently under-explored. Inspired by their success in other applications, we presented a graph-based multilevel scheme for linear dimensionality reduction [15]. The goal of this paper is to expand this work by presenting a multilevel framework for *nonlinear dimensionality reduction*. The framework of these methods relies on an affinity graph and so it can be especially useful for affinity-graph-based manifold learning methods.

The multilevel framework proposed in this paper consists of three phases: data coarsening, nonlinear dimension reduction, and data refining. To coarsen the data, we employ a graph coarsening algorithm based on maximum independent sets. In practice, it is common to employ a k-nearest neighbor (kNN) graph at the highest level. After this, we project the coarsened data at the lowest level using one of several known (nonlinear) dimensionality reduction method for manifold learning. Finally, we recursively refine the data level by level, by solving a linear system to go from a given level to a higher level. The linear system comes from a least squares optimization which aims to preserve the closeness of data points between two adjacent levels.

Landmark versions of Isomap [5] and maximum variance unfolding (MVU) [26] by random sampling have been proposed to reduce the problem size and therefore the computational cost. The method proposed in this paper has three distinct advantages over the landmark approach. First, maximum independent sets provide a better representation of the original data than landmarks obtained from random sampling. Second, a kNN graph of the full data set rather than the sampled data points, is computed. By recursive coarsening we obtain a succession of graphs on which our refining scheme is based and this phase is independent of the dimensionality reduction method. Third, the multilevel structure propagates the geodesic information into the coarsened graphs and this may be beneficial to some manifold learning algorithms which do not preserve isometry.

In this paper we consider three well-known manifold learning algorithms: Isomap [22], Locally Linear Embedding (LLE) [14, 17], and Laplacian eigenmaps [2], which are representative in manifold learning [18]. Note that our multilevel framework is not limited to these methods. It can be applied to virtually all affinity-graph-based manifold learning methods, such as maximum variance unfolding (MVU) [27], Hessian LLE [6], conformal Isomap [5], incremental Isomap [13], diffusion maps [4, 12], conformal eigenmaps [19], and minimum volume embedding [20].

The rest of this paper is organized as follows. Section 2 reviews the three manifold learning algorithms, namely Isomap and LLE, and Laplacian eigenmaps. Section 3 presents our multilevel nonlinear dimensionality reduction framework. An application to clustering, the multilevel K-means algorithm, is discussed in Section 4. Sections 5 and 6 report on some results of manifold learning experiments and clustering experiments, respectively. All experiments were performed in sequential mode on a PC equipped with two dual-core AMD

Opteron(tm) 2214 @ 2.2GHz processors, using our Matlab implementation. A conclusion is given in Section 7.

2 Manifold Learning

We say that a given open set $\Psi \in \mathbb{R}^m$ in m-dimensional Euclidean space resides in a lower d-dimensional manifold (typically $d \ll m$), if there is a continuously differential function $f : \Omega \to \mathbb{R}^m$ on an open domain $\Omega \in \mathbb{R}^d$, such that $f(\Omega) = \Psi$. The parameterized manifold $\Psi = f(\Omega)$ is called *regular*, if the Jacobian matrix $J(y)$ of $f(y)$ has full rank for all $y \in \Omega$, and $f(y)$ does not self-intersect; i.e., $y_i \neq y_j$ implies $f(y_i) \neq f(y_j)$. We call the mapping f *isometric*, if it preserves the Euclidean distances between nearby points. In other words, $\|f(y+p) - f(y)\|_2 \approx \|p\|_2$ for $y, y+p \in \Omega$ and small $\|p\|_2$. Formally, $\|J(y)p\|_2 = \|p\|_2$ for $y \in \Omega$ and $p \in \mathbb{R}^d$, which means that the singular values of $J(y)$ are all one, or equivalently, $J(y)$ consists of orthogonal columns for $y \in \Omega$.

Manifold learning methods attempt to find a function f that maps points in $\Omega \in \mathbb{R}^m$ into points of a (much) lower dimension $\mathbb{R}^d$. In practice, we often have a discrete and possibly noisy sampled data $x_1, \ldots, x_n \in \mathbb{R}^m$ of Ψ, and the objective is to find the corresponding low dimensional embedding $y_1, \ldots, y_n \in \mathbb{R}^d$. The goal of the mapping is to preserve the closeness of nearby points, for which an affinity graph $G = (V, E)$, normally a kNN graph, is employed.

In this paper, we use matrices $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m \times n}$ and $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d \times n}$ $(d < n)$ to denote the original high dimensional data and the mapped low dimensional data, respectively. The column vector of ones is denoted by e. We also use integers $1, \ldots, n$ to denote the vertices of the affinity graph $G = (V, E)$, i.e., $V = \{1, \ldots, n\}$.

2.1 Isomap

Isomap [22] is a nonlinear generalization of the linear multidimensional scaling (MDS). It replaces the Euclidean distances in MDS by the *geodesic* distances approximated by an affinity graph $G = (V, E)$, whose vertices $1, \ldots, n$ in V correspond to the input data $x_1, \ldots, x_n \in \mathbb{R}^m$, and edges in E define the closeness of them. The length of the shortest path between vertices x_i and x_j, denoted by $\tilde{d}_{ij}$, is the approximate geodesic distance between them.

The algorithm can be summarized as follows. It starts by constructing an affinity graph, typically a kNN graph for the data. With this, the all-pair shortest path problem is solved and all the squared approximate geodesic distances $\tilde{d}_{ij}^2$ are saved in a symmetric matrix $\widetilde{D} \in \mathbb{R}^{n \times n}$. The next step is to compute the Grammian matrix $\widetilde{B} = -\frac{1}{2} J \widetilde{D} J \in \mathbb{R}^{n \times n}$, where $J = I - \frac{1}{n} e e^T \in \mathbb{R}^{n \times n}$ with $I \in \mathbb{R}^{n \times n}$ the identity matrix and $e \in \mathbb{R}^n$ a column vector of ones. 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 precise, denote by $\lambda_i \in \mathbb{R}$ and $v_i \in \mathbb{R}^n$ the ith eigenvalue and eigenvector of $\widetilde{B}$ in decreasing order. Let $\Sigma_d \in \mathbb{R}^{d \times d}$ be the diagonal matrix formed by $\lambda_1, \ldots, \lambda_d$, and the columns of $V_d \in \mathbb{R}^{n \times d}$ be $v_1, \ldots, v_d$. The mapped low dimensional data is $Y = \Sigma_d^{1/2} V_d^T \in \mathbb{R}^{d \times n}$.

The relation between the metric MDS and Isomap is worth noting. The metric MDS uses a distance matrix D whose (i, j) entry is $\|x_i - x_j\|_2^2$. 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 $x_i^T x_j$, i.e., $B = X^T X$. The linear mapping $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ is obtained from minimizing $\|B - Y^T Y\|_F$. Alternatively, Isomap minimizes $\|\widetilde{B} - Y^T Y\|_F$ to obtain the low dimensional data $Y \in \mathbb{R}^{d\times n}$, 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.

2.2 Locally Linear Embedding

Locally linear embedding (LLE) [14, 17] 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.

First, a kNN graph is constructed. Second, the reconstruction weights $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ are obtained by minimizing the cost function:

$$\mathcal{E}(W) = \sum_{i=1}^n \|x_i - \sum_{j=1}^n w_{ij} x_j\|_2^2, \tag{1}$$

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 $\|x_i - \sum_{j=1}^n w_{ij} x_j\|_2^2$ in (1) requires solving a constrained least squares problem for each $i = 1, \ldots, n$.

Finally, the high dimensional data $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ is mapped to the low dimensional data $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ by minimizing the embedding cost function:

$$\Phi(Y) = \sum_{i=1}^n \|y_i - \sum_{j=1}^n w_{ij} y_j\|_2^2 = \|Y - YW^T\|_F^2 = \mathrm{trace}[Y(I-W)^T(I-W)Y^T]. \tag{2}$$

Two constraints are added for the problem to be well-posed. First, it is required that the projected data be centered, i.e., $\sum_{i=1}^n y_i = 0$. Second, the mapped data, subject to scaling, must have unit covariance, i.e., $\sum_{i=1}^n y_i y_i^T = YY^T = I$.

Let $M = (I-W)^T(I-W)$. Then $Me = 0$, where e is the column vector of ones. Therefore, e is a eigenvector of M associated with the smallest eigenvector 0. Other eigenvectors v satisfy $v^T e = 0$. The embedding is formed by the d right singular vectors of $I - W$ corresponding to the second to the $(d+1)$st singular values in increasing order.

2.3 Laplacian Eigenmaps

In Laplacian eigenmaps, an affinity graph of $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$ is also constructed. The low dimensional embedding $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d\times n}$ is the minimizer of the cost function:

$$\Psi(Y) = \sum_{i,j} w_{ij}\|y_i - y_j\|_2^2 = 2\,\mathrm{trace}(Y(D-W)Y^T), \tag{3}$$

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

The weights can be *simple-minded:* $w_{ij} = 1$ if x_i and x_j are neighbors defined by the affinity graph, and otherwise $w_{ij} = 0$. Alternatively, we may use the *heat kernel*:

$$w_{ij} = \exp(-\|x_i - x_j\|_2^2/\sigma^2) \tag{4}$$

for each pair of neighboring points x_i, x_j, where $\sigma > 0$ is a preset parameter. Setting $\sigma = \infty$ in (4), we obtain the simple-minded weighting method.

To make the minimization of (3) well-posed, the constraint $YDY^T = I$ is imposed. The problem is transformed to solving the generalized eigenvalue problem $(D - W)z = \lambda Dz$, whose d generalized eigenvectors corresponding to the second to the $(d+1)$st eigenvalues form Y. The bottom generalized eigenvector e associated with eigenvalue 0 is ignored.

2.4 Discussion

Some characteristics of the methods just described are now summarized; see [18]. First, all algorithms first construct an affinity graph of the input data, typically a kNN graph. Isomap, and Laplacian eigenmaps require that the affinity graph be undirected, while LLE normally employs a kNN graph without symmetrization (i.e., still a directed graph). Second, Isomap which makes an implicit assumption of isometry of the manifold mapping, aims to preserve geodesic distances. On the other hand, LLE and Laplacian eigenmaps are designed to preserve the closeness of nearby points, and therefore not isometric. Third, LLE and Laplacian eigenmaps are relatively inexpensive. They compute the eigenvalues and generalized eigenvalues of sparse matrices, respectively. Isomap is more expensive since it begins by solving an all-pair shortest path problem and then computes eigenvalues of a dense Grammian matrix. These properties will be discussed again later in the framework of our multilevel technique.

3 Multilevel Nonlinear Dimensionality Reduction

This section presents our multilevel framework for nonlinear dimensionality reduction for manifold learning. This approach consists of three phases: data coarsening, nonlinear dimension reduction, and data refining. Figure 1 provides an illustration. 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 lowest level using a standard dimension reduction technique such as Isomap, LLE, or Laplacian eigenmaps. Then an 'uncoarsening' step of this low dimensional data is performed backing up to the highest level. Details are provided next.

3.1 The Coarsening Phase

Coarsening a graph $G = (V, E)$ means finding a 'coarse' approximation $\widehat{G} = (\widehat{V}, \widehat{E})$ that represents $G = (V, E)$, where $|\widehat{V}| < |V|$. By recursively coarsening we obtain a succession of smaller graphs which approximate the original graph G.

For graph coarsening steps we used maximum independent sets, which have been in use for multilevel graph partitioning [1, 3]. Connectivity of an affinity graph is important to many manifold learning algorithms but coarsening by maximum independent sets does not guarantee that the coarse graph is connected. However, Algorithm 1 visits the vertices in a special order to build the maximum independent set, so that it preserves the connectivity of the graph in the coarsening stage. This is now explained.

Consider the steps of Algorithm 1 to compute the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$. We claim that for each vertex k added to $\widehat{V}$, other than the very first element k_0 added to S, there

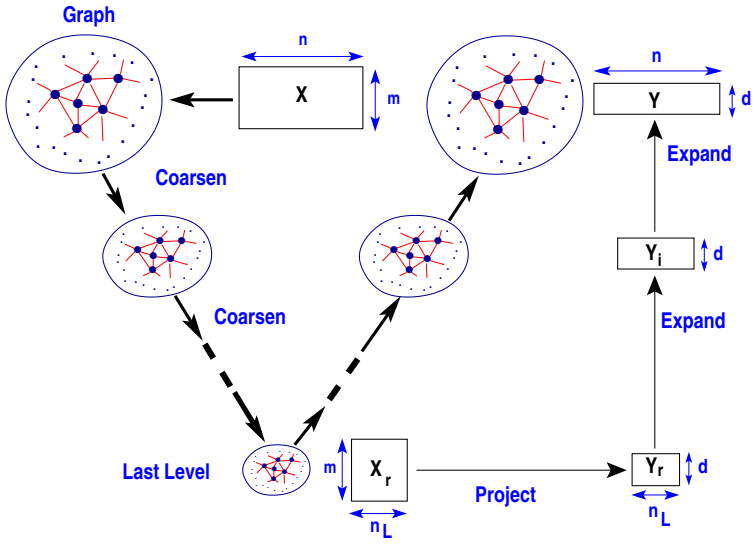


Figure 1: A sketch of the multilevel reduction.

exists a path consisting of edges in $\widehat{E}$ linking vertices k_0 and k. We now prove our claim by induction. All vertices in $\widehat{V}$ are from S in (*) and added in (**). Each element k ever in S, except the very first k_0 in (†), is added to S in (††), where there exist $(i,j),(j,k) \in E$ with i already in $\widehat{V}$. Since there is a path $i \to j \to k$ in the fine graph, if k is added into $\widehat{V}$ in some later iteration, then there will be an edge $(i,k) \in \widehat{E}$ in the coarse graph as instructed by the bottom part of the algorithm. Assuming that previous vertices added to $\widehat{V}$ satisfy our claim, there exists a path consisting of edges in $\widehat{E}$ linking k_0 and i, unless $i = k_0$. Since $(i,k) \in \bar{E}$, k_0 also links to k via a path in the coarse graph. This proves our claim by induction. Therefore, the coarse graph $\widehat{G} = (\widehat{V}, \widehat{E})$ is guaranteed to be connected under the condition that the original graph is.

Algorithm 1 provides an affinity graph $\widehat{G} = (\widehat{V}, \widehat{E})$ of the coarse level. Therefore, it is not necessary to compute a kNN graph for the graphs obtained at each level. In addition, we need the distances between nearby points in the coarse graph in the following two situations. First, some manifold learning algorithms, such as Isomap, need distances between nearby points to compute the mapping. Second, the multilevel refining stage, to be described later, will require the edge weights, and some weighting schemes, such as heat kernel, depend on the distances between nearby points.

We use δ and $\hat{\delta}$ to denote the distances at the fine and coarse levels, respectively. Given $(i,j) \in \widehat{E}$, one can simply use the actual distance $\hat{\delta}(x_i, x_j) = \|x_i - x_j\|_2$ for the coarse level. Alternatively, we can define

$$\hat{\delta}(x_i, x_j) = \min_{(i,k),(k,j) \in E} \delta(x_i, x_k) + \delta(x_k, x_j). \tag{5}$$

Then distance computations are avoided at the coarse level. More importantly, if we compute distances by (5) at all levels, the computed distances indeed approximate geodesic distances. Recall that the goal of manifold learning is to unfold the underlying structure of a given data set into a lower dimensional space. Therefore, geodesic distances appear more useful than

Input: A connected undirected graph $G = (V, E)$ with $V = \{1, \ldots, n\}$.
Output: The coarsened graph $\widehat{G} = (\widehat{V}, \widehat{E})$.
$\widehat{V} \leftarrow \emptyset$ ▷ maximum independent set
$\widehat{U} \leftarrow \emptyset$ ▷ complement set of $\widehat{V}$
Randomly pick $k_0 \in V$; $S \leftarrow \{k_0\}$. ▷ (†)
repeat
  Randomly pick $i \in S$; $S \leftarrow S \setminus \{i\}$. ▷ (*)
  if $i \notin \widehat{U} \cup \widehat{V}$ **then**
    $\widehat{V} \leftarrow \widehat{V} \cup \{i\}$ ▷ (**)
    for all $(i, j) \in E$, $j \notin \widehat{U}$ **do**
      $\widehat{U} \leftarrow \widehat{U} \cup \{j\}$
      for all $(j, k) \in E$ **do**
        if $k \notin \widehat{U} \cup \widehat{V}$ **then**
          $S \leftarrow S \cup \{k\}$ ▷ (††)
        end if
      end for
    end for
  end if
until $S = \emptyset$
$\widehat{E} \leftarrow \emptyset$ ▷ edge set of $\widehat{G}$
for all $i, k \in \widehat{V}$ **do**
  if $\exists j$ such that $(i, j), (j, k) \in E$ **then**
    $\widehat{E} \leftarrow \widehat{E} \cup \{(i, k)\}$
  end if
end for

Algorithm 1: Graph coarsening by a maximum independent set.

actual distances to discover the nonlinear manifold. This is especially important to Isomap which aims at preserving isometry.

By recursively coarsening the graph, we obtain a succession of graphs $G_1, G_2, \ldots, G_r$, where $G_i = (V_i, E_i)$ is the coarse graph of level i for $i = 1, \ldots, r$, and G_r is the lowest level graph. The corresponding data sets are denoted by matrices $X_i \in \mathbb{R}^{m \times |V_i|}$ for $i = 1, \ldots, r$.

3.2 The Dimension Reduction Phase

Given a data set $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$, a dimensionality reduction algorithm produces $Y = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^{d \times n}$ $(d < m)$ such that Y preserves certain features of X. In our multilevel framework, presented in Figure 1, we apply a dimensionality reduction method to the data set $X_r \in \mathbb{R}^{m \times |V_r|}$ of the lowest level (rth level), and obtain a set $Y_r \in \mathbb{R}^{d \times |V_r|}$ $(d < m)$. The dimensionality reduction methods considered for this task are affinity-graph-based, such as Isomap, LLE, and Laplacian eigenmaps, where the graph from the multilevel framework is used. Recall that it is not necessary to build a kNN graph at the lowest level.

Note that Isomap and Laplacian eigenmaps use an undirected affinity graph (i.e., applying symmetrization to a kNN graph), whereas LLE uses a directed affinity graph (i.e., a kNN

graph without symmetrization). In our multilevel framework the affinity graph is undirected, regardless of the dimensionality reduction method applied at the bottom level.

3.3 The Refining Phase

The objective of the refining phase is to obtain a reduced representation $Y \in \mathbb{R}^{d\times n}$ of the data $X \in \mathbb{R}^{m\times n}$, where $n = |V_1|$, at the topmost level, starting from the reduced representation $Y_r \in \mathbb{R}^{d\times|V_r|}$ of data $X_r \in \mathbb{R}^{m\times|V_r|}$ of the lowest level (rth level).

We refine 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 graphs of the kth and $(k+1)$st levels, respectively. For each level $k = r-1, r-2, \ldots, 1$, we recursively build the reduced representation Y of the kth level from $\widehat{Y}$ of the $(k+1)$st level in a low dimensional space, by solving a least squares problem which minimizes the sum of squared distances between data points in the low dimensional space:

$$E = \sum_{i,j\in V} w_{ij}\|y_i - y_j\|_2^2, \tag{6}$$

where $W = [w_{ij}]$ is a symmetric weight matrix; each entry w_{ij} is nonzero only if the vertices i, j are adjacent (i.e., connected by an edge). The closer the vertices, the heavier the weight.

Yet not specified are the weights between nearby data points. We adopt the two weighting schemes used in Laplacian eigenmaps [2]. One is the heat kernel, $w_{ij} = e^{-\delta(x_i,x_j)^2/\sigma^2}$ for some scalar $\sigma > 0$. The distance function $\delta(x_i, x_j)$ between x_i and x_j can be the Euclidean distance $\|x_i - x_j\|_2$ as that in (4). With our multilevel framework we use the approximate geodesic distance (5) across all levels, since this is a more faithful distance measure for the underlying manifold of the given data. The other scheme, obtained when $\sigma = \infty$, is the 'simple-minded' weighting, in which $w_{ij} = 1$ for all adjacent vertices.

We denote the vertex set of the coarse level by $\widehat{V} \subset V$, and its complement by $\widehat{U} = V \setminus \widehat{V}$. Therefore $\widehat{U} \cup \widehat{V} = V$ and $\widehat{U} \cap \widehat{V} = \emptyset$. Since the weights are symmetric, we can rewrite (6) as

$$E = \sum_{i\in\widehat{U}}\sum_{j\in\widehat{U}} w_{ij}\|y_i - y_j\|_2^2 + \sum_{i\in\widehat{V}}\sum_{j\in\widehat{V}} w_{ij}\|y_i - y_j\|_2^2 + 2\sum_{i\in\widehat{U}}\sum_{j\in\widehat{V}} w_{ij}\|y_i - y_j\|_2^2. \tag{7}$$

The first term of (7) can be written as

$$\sum_{i\in\widehat{U}}\sum_{j\in\widehat{U}} w_{ij}\|y_i - y_j\|_2^2 = 2\,\mathrm{trace}[Y_1(D_1 - W_1)Y_1^T], \tag{8}$$

where $Y_1 \in \mathbb{R}^{d\times|\widehat{U}|}$ includes the points to be determined in Y, $W_1 \in \mathbb{R}^{|\widehat{U}|\times|\widehat{U}|}$ is the weight matrix between all points in Y_1, and $D_1 \in \mathbb{R}^{|\widehat{U}|\times|\widehat{U}|}$ is the diagonal matrix whose entries are the row/column sums of W_1.

The second term of (7) is a constant,

$$\sum_{i\in\widehat{V}}\sum_{j\in\widehat{V}} w_{ij}\|y_i - y_j\|_2^2 = 2\,\mathrm{trace}[Y_2(D_2 - W_2)Y_2^T] = \mathrm{Const}_1, \tag{9}$$

since it depends only on points in $Y_2 \in \mathbb{R}^{d\times|\widehat{V}|}$ that have been already determined at the coarse level. The matrices D_2 and W_2 in (9) are defined similarly to D_1 and W_1 in (8).

The third term of (7), after some algebra, can be written as

$$\begin{aligned} 2\sum_{i\in\widehat{U}}\sum_{j\in\widehat{V}} w_{ij}\|y_i - y_j\|_2^2 &= 2\sum_{i\in\widehat{U}}\sum_{j\in\widehat{V}} w_{ij} y_i^T y_i - 4\sum_{i\in\widehat{U}}\sum_{j\in\widehat{V}} w_{ij} y_i^T y_j + 2\sum_{i\in\widehat{U}}\sum_{j\in\widehat{V}} w_{ij} y_j^T y_j \\ &= 2\,\mathrm{trace}[Y_1 D_{12} Y_1^T] - 4\,\mathrm{trace}[Y_1 W_{12} Y_2^T] + \mathrm{Const}_2, \end{aligned} \tag{10}$$

where $W_{12} \in \mathbb{R}^{|\widehat{U}|\times|\widehat{V}|}$ is the weight matrix between the points to be determined (i.e., indexed by $\widehat{U}$) and those already determined (i.e., indexed by $\widehat{V}$), $D_{12} \in \mathbb{R}^{|\widehat{U}|\times|\widehat{U}|}$ is the diagonal matrix whose entries are the column sums of W_{12}.

Putting these expressions (8), (9), and (10) together back into (7), we obtain a quadratic function:

$$E = 2\,\mathrm{trace}[Y_1(D_1 - W_1 + D_{12})Y_1^T] - 4\,\mathrm{trace}[Y_2 W_{12}^T Y_1^T] + \mathrm{Const.} \tag{11}$$

To minimize E, we set the partial derivatives of (11) to zero, and obtain

$$Y_1(L_1 + D_{12}) = Y_2 W_{12}^T, \tag{12}$$

where $L_1 = D_1 - W_1 \in \mathbb{R}^{|\widehat{U}|\times|\widehat{U}|}$ is the Laplacian matrix of the points to be determined.

Two observations deserve noting. First, L_1 is symmetric and diagonally dominant with a positive diagonal. By the Gershgorin circle theorem, L_1 is positive semidefinite. It also has the smallest eigenvalue 0 associated with eigenvector e, the column vector of ones. D_{12} is diagonal with nonnegative entries. Therefore, the objective function (11) is convex, and Y_1 is the minimizer if and only if (12) holds. Second, our data coarsening method is based on maximum independent sets, and we refine the mapping level by level. Hence each undetermined vertex $i \in \hat{U}$ has at least one determined neighbor $j \in \hat{V}$ associated with a positive weight $w_{ij} > 0$. So D_{12} has a positive diagonal. Recall that L_1 is symmetric positive semidefinite. By a theorem of Weyl [21, Corollary 4.9], stated below, $L_1 + D_{12}$ is positive definite and therefore nonsingular. Thus, the solution to the linear system (12) is unique, and so is the minimizer of (11).

Theorem 1 (Weyl) *Let A, B be two $n\times n$ Hermitian matrices and $\lambda_k(A)$, $\lambda_k(B)$, $\lambda_k(A+B)$ be the eigenvalues of A, B, and $A+B$ arranged in increasing order for $k = 1, \ldots, n$. Then for $k = 1, \ldots, n$, we have*

$$\lambda_k(A) + \lambda_1(B) \le \lambda_k(A+B) \le \lambda_k(A) + \lambda_n(B).$$

Putting the points as columns of Y_1 (i.e., indexed by $\widehat{U}$) and those already determined as columns of Y_2 (i.e., indexed by $\widehat{V}$) together, we obtain the reduced representation of the finer level (i.e., vertices indexed by $\widehat{U} \cup \widehat{V}$). By recursively refining the data this way, we obtain a reduced representation of the original data.

4 Application to Clustering

Given set of points $X = [x_1, x_2, \ldots, x_n]$ in Euclidean space, the objective of clustering is to partition it into a certain number of subsets, called clusters, which are as distinct as

possible. The K-means algorithm, as one of the best-known clustering methods available, (locally) minimizes the quantization error:

$$E(s, w) = \sum_{i}^{n} \|x_i - c(s(i))\|_2^2, \tag{13}$$

where $s(i)$ is the index of the cluster to which x_i belongs, and $c(j)$ is the prototype, e.g., the centroid of cluster j.

When the clustering s is fixed, the minimizer of (13) in terms of c is when $c(j)$ is the centroid of the data entries in cluster j. On the other hand, if c is fixed, the minimizer of (13) in terms of s is reached when $s(i)$ is the cluster index of the closest prototype to x_i. K-means iteratively minimizes $E(s, c)$ in terms of s and c, until the value of $E(s, c)$ cannot be further reduced. For details, see, e.g., [9, chapter 14].

The main drawback of the K-means clustering is that it is sensitive to initialization. Random initialization could yield poor results in extreme cases, which may be avoided by a structured initialization scheme utilizing our multilevel technique. The procedure is sketched next.

We first recursively apply the graph coarsening method in Algorithm 1, and obtain a succession of graphs $G_i = (V_i, E_i)$ for $i = 1, \ldots, r$, where $V_1 = \{1, \ldots, n\}$ is the set of indices of the given data $x_1, \ldots, x_n$. Then we cluster the data points at the bottom level by the K-means algorithm, which may be initialized randomly. For each level $i = r-1, r-2, \ldots, 1$, we still do the K-means clustering, initialized by the clustering centroids at $(i+1)$st level.

For high dimensional data, it is useful to remove redundancy and noise of the data by dimensionality reduction techniques. In our dimensionality reduction framework presented in Figure 1, we have a succession of sets of low dimensional points $Y_r, Y_{r-1}, \ldots, Y_1$, to which we apply the K-means algorithm sequentially. The cluster centroids at level $i+1$ are used to initialize the K-means clustering at level i for $i = r-1, r-2, \ldots, 1$. At the bottom level r we may use random initialization. The pseudocode is given in Algorithm 2.

{Given $X = [x_1, \cdots, x_n] \in \mathbb{R}^{m \times n}$, partition it into K clusters.}
Obtain $Y_1, Y_2, \ldots, Y_r$ by multilevel dimensionality reduction. ▷ Figure 1
Apply the K-means clustering, initialized randomly, to bottom level points Y_r.
for $i = r-1, \ldots, 1$ **do**
    Apply K-means clustering to points in Y_i, initialized by the K centroids at level $i+1$.
end for

Algorithm 2: Multilevel K-means clustering with dimensionality reduction.

5 Manifold Learning Experiments

In this section we illustrate the application of the proposed multilevel manifold learning scheme to various data sets. We use the three nonlinear dimensionality reduction methods, Isomap [22], LLE [17], and Laplacian eigenmaps [2], and the versions of the multilevel algorithms which incorporate these techniques at the lowest level as described earlier. The

Floyd-Warshall algorithm was utilized to solve the all-pairs shortest path problem [7, 25] which arises in Isomap and multilevel-Isomap.

Section 5.1 describes the embedding evaluation metrics used in our experiments. The outputs of two sampled synthetic data sets, `Swissroll` and `S-curve`, are displayed in Section 5.2. Sections 5.3, 5.4 and 5.5 present the results of experiments on three data sets, `Sculpture` images, `Frey Face` video frames, and `Teapot` images, respectively.

Since Algorithm 1 for coarsening the data is randomized, we report the average numbers from 100 random runs for each data set, each method, and each level $r = 2, 3, 4$ in Tables 1–3, which display the average number of images at each coarsening level, and the average CPU time used for graph coarsening, processing for dimensionality reduction, and data refining. For all methods, processing time includes the time used for eigen-computation. For Isomap and multilevel-Isomap, processing time also includes the time to compute the geodesic distances. For LLE, it includes the time to obtain the reconstruction weights. In the embedding quality measurement plots in Figures 7, 9, and 11, we used 50 random runs for each data set, each method, and each level $r = 2, 3, 4$ and took the average.

We may 'fix' Algorithm 1 (i.e., without randomization) by visiting the vertices in the order in which the data items are listed. By doing so the manifold mappings in Figures 3–6, 8, and 10 were obtained.

5.1 Embedding Evaluation

In order to compare the quality of the nonlinearly mapped data, we adopt the embedding evaluation metrics, the *trustworthiness* and *continuity* of the proximity relationships of data entries [23, 24].

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 $\hat{r}(i, j)$ the rank of y_j in the ordering according to the distance from y_i. The trustworthiness is defined by

$$T(p) = \frac{2}{np(2n - 3p - 1)} \sum_{i=1}^{n} \sum_{j \in U_p(i)} (r(i, j) - p),$$

where $U_p(i)$ contains the indices of p nearest neighbors of y_i in the low dimensional space. The continuity is defined by

$$C(p) = \frac{2}{np(2n - 3p - 1)} \sum_{i=1}^{n} \sum_{j \in V_p(i)} (\hat{r}(i, j) - p),$$

where $V_p(i)$ contains the indices of p nearest neighbors of x_i in the high dimensional space.

The higher the trustworthiness or continuity, the better the performance. Both $T(p)$ and $C(p)$ are bounded above by 1. The upper bound 1 is reached if and only if $U_p(i) = V_p(i)$ for $i = 1, \ldots, n$, which means that the p nearest neighbors for each data entry in the high dimensional space coincide with those in the low dimensional space.

We also measure the mapping quality by the harmonic mean of the trustworthiness and continuity, which we call *H-score*:

$$H(p) = \frac{2\,T(p)C(p)}{T(p) + C(p)}.$$

5.2 Synthetic Data

We used two synthetic data sets sampled in three-dimensional space: the `Swissroll` and the `S-curve`, each with 2,000 sample points, as shown in Figure 2. These data sets, though embedded in three-dimensional space, reside on two-dimensional manifolds.

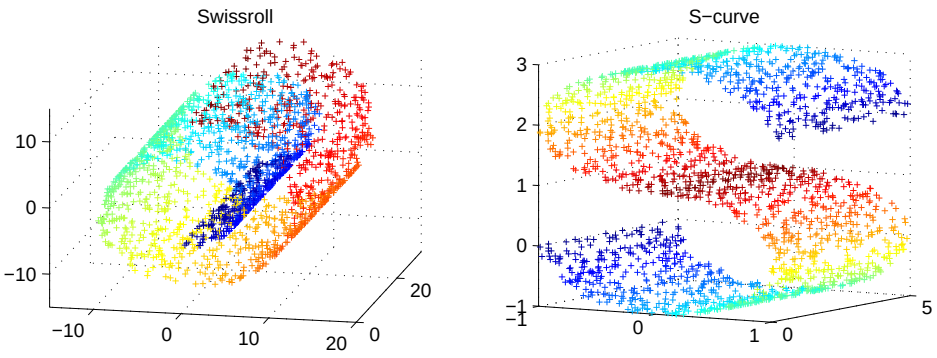


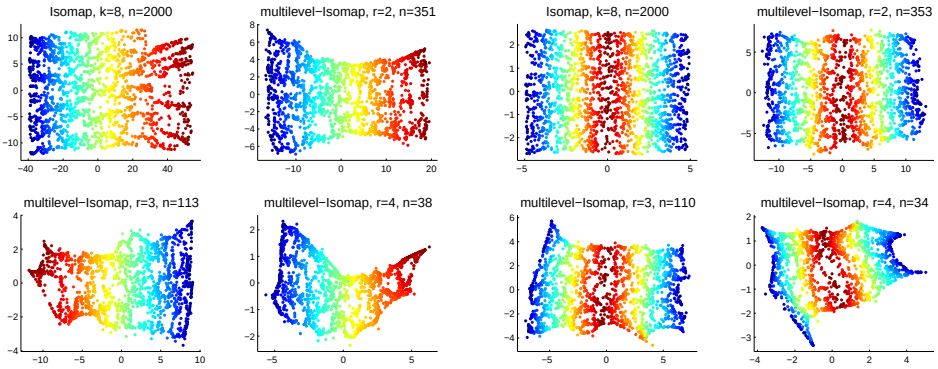
Figure 2: Two examples of data points sampled on 3-D manifolds.

Figures 3–5 illustrate the two-dimensional projections of `Swissroll` and `S-curve` data sets using the embedding methods Isomap, LLE, Laplacian eigenmaps, and those with multi-level techniques with the number of levels $r = 2, 3, 4$. In the kNN graph construction, we set to $k = 8$ the number of nearest neighbors per sampled point. In each graph coarsening step, the vertices were visited in the order in which the data items are listed. For data refining and also for Laplacian eigenmaps, we used the simple-minded weighting scheme. For `Swissroll`, the number of points at each of the four levels is 2,000, 351, 113, and 38, respectively. For `S-curve`, the numbers of points at four levels are 2,000, 353, 110, and 34, respectively. The result tends to indicate that the cohesiveness is pretty much kept while the number of levels is increased.

5.3 Sculpture Face Images

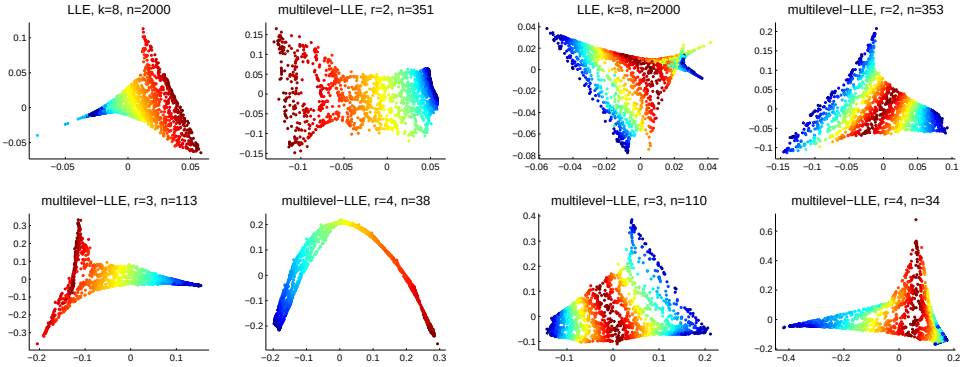
The `Sculpture Face` data set [22][1] includes 698 images of size 64-by-64 in grayscale of a sculpture face rendered with different poses and lighting directions. Within the 4,096-dimensional input space, all of the images lie on an intrinsically three-dimensional manifold, that can be parameterized by three variables: left-right pose, up-down pose, and the lighting direction.

[1] `http://isomap.stanford.edu/datasets.html`



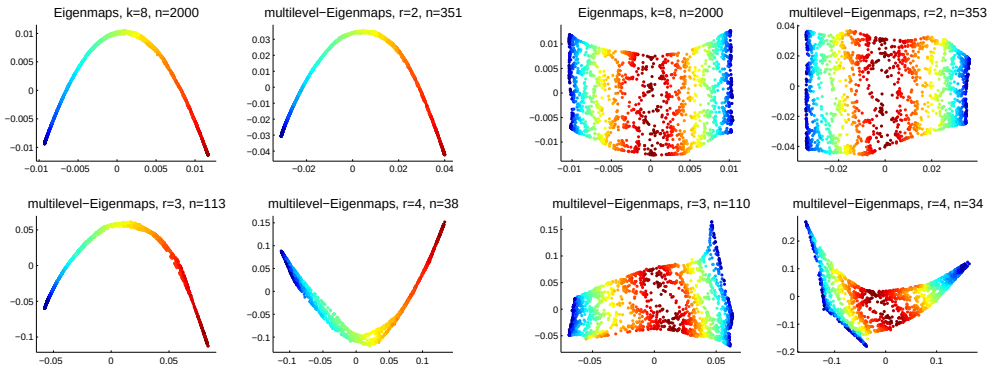
(a) `Swissroll` (b) `S-curve`

Figure 3: 2D projections using Isomap and multilevel-Isomap ($k = 8$).



(a) `Swissroll` (b) `S-curve`

Figure 4: 2D projections using LLE and multilevel-LLE ($k = 8$).



(a) `Swissroll` (b) `S-curve`

Figure 5: 2D projections using Eigenmaps and multilevel-Eigenmaps ($k = 8$).

We report the results of experiments using a kNN graph with $k = 6$ and embedding dimensions $d = 2$. In our multilevel framework we used the heat kernel weighting scheme when refining the data, and also when the Laplacian eigenmaps is employed.

Figure 6 illustrates the two-dimensional mappings using Isomap and multilevel-Isomap. 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. The plots by LLE, multilevel-LLE, and Eigenmaps and multilevel-Eigenmaps, not shown due to space limit, also have this characteristic.

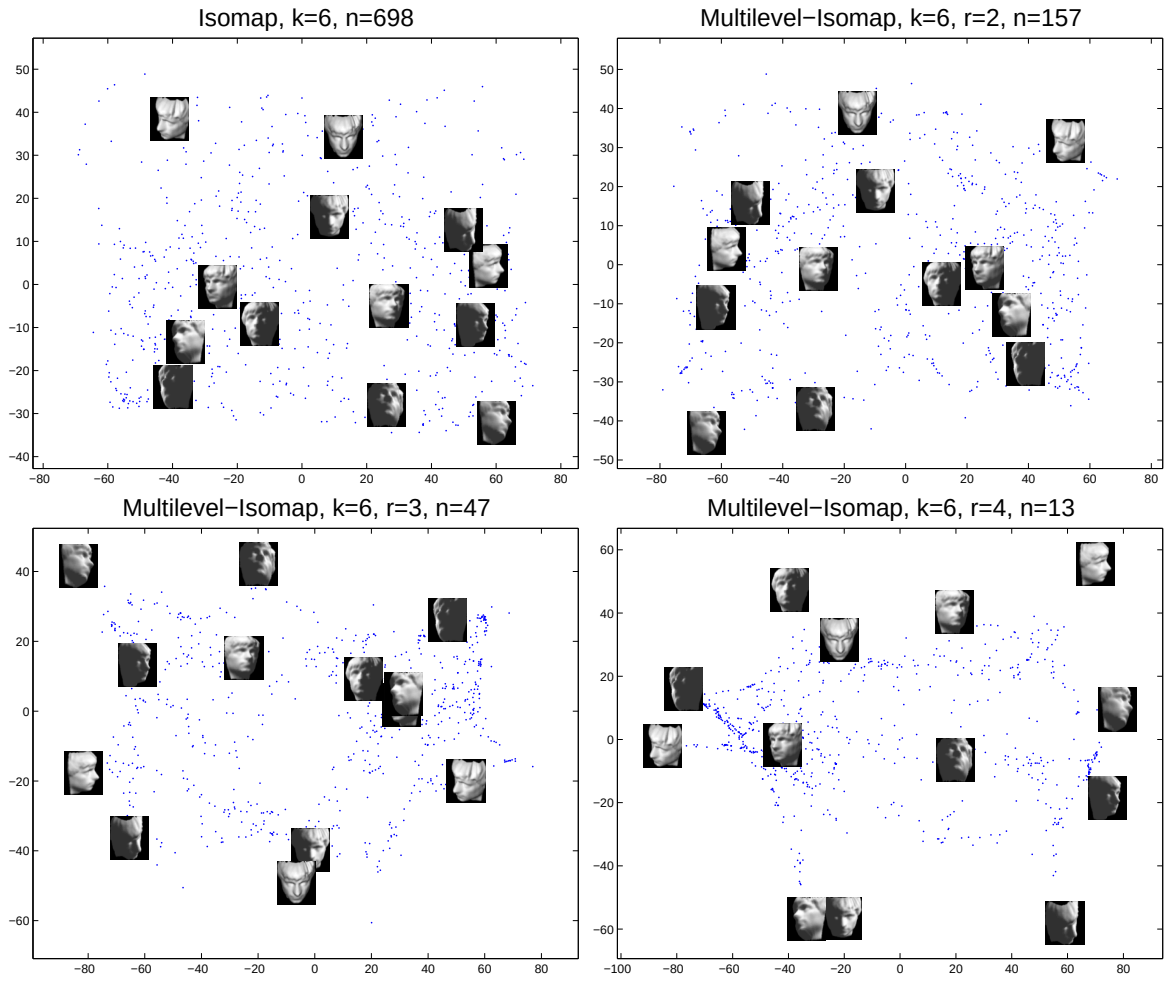


Figure 6: 2D mappings of `Sculpture Face` data set by Isomap and multilevel-Isomap.

Table 1 reports the average CPU time for kNN graph construction, graph coarsening, dimensionality reduction, and data refining. Note that our multilevel technique generally achieved significant savings in CPU time. For example, using $r = 2$ levels, our multilevel technique achieved about 83% savings in computation time for Isomap, 34% savings for LLE, and insignificant (2%) savings for Eigenmaps. Omitting the time for kNN graph construction, the savings were 99%, 84%, and 38% for Isomap, LLE, and Eigenmaps. Using more levels resulted in more time savings.

Table 1: Computation time for `Sculpture Face` data set.

kNN time (secs)	level	average # of images	coarsen. time (secs)	Isomap proc. time	Isomap ref. time	LLE proc. time	LLE ref. time	Eigenmaps proc. time	Eigenmaps ref. time
1.58	#1	698	N/A	12.180	N/A	1.0600	N/A	0.1000	N/A
	#2	142.69	0.0330	0.0744	0.0110	0.1065	0.0329	0.0192	0.0099
	#3	44.13	0.0058	0.0104	0.0018	0.0326	0.0062	0.0157	0.0020
	#4	15.54	0.0015	0.0062	0.0007	0.0137	0.0014	0.0131	0.0010

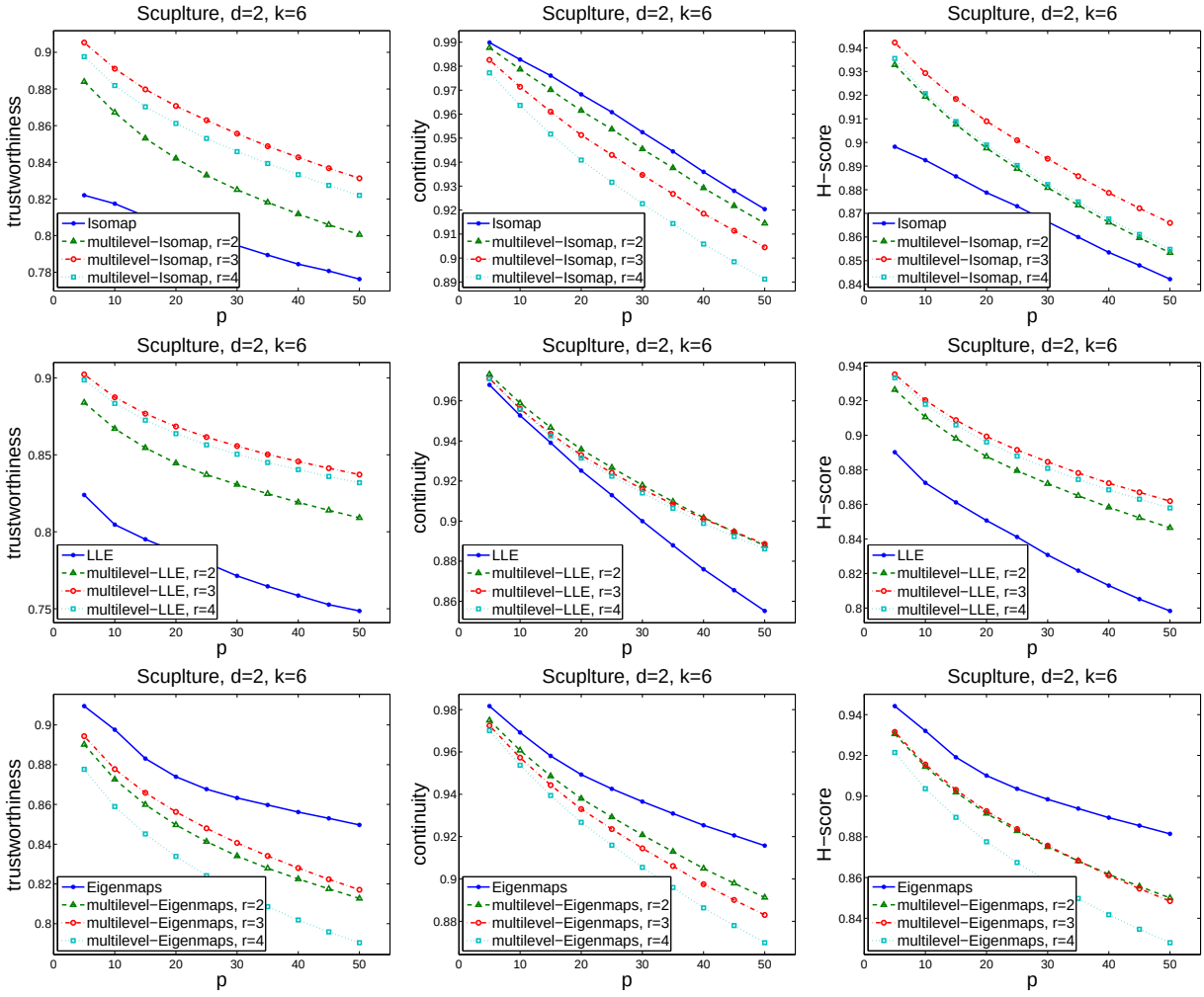


Figure 7: Trustworthiness, continuity, and H-score of `Sculpture Face` data set by Isomap and multilevel-Isomap, LLE and multilevel-LLE, Eigenmaps and multilevel-Eigenmaps.

Figure 7 displays the plots of trustworthiness, continuity, and H-score values as a function of p, the size of the neighborhood used in the measurement, where we set the number of levels up to four. In this experiment our multilevel technique improved Isomap and LLE, but multilevel-Eigenmaps performed less satisfactory than Eigenmaps.

5.4 Frey Face Video Frames

The `Frey Face` data set [17][2] contains 1,965 face images of a single person, Brendan Frey, taken from sequential frames of a small video. Each image is of size 20-by-28 in grayscale, and hence in 560-dimensional space after vectorization.

We report the result using a kNN graph with $k = 12$ and embedding dimensions $d = 2$. In our multilevel framework we used the heat kernel weighting scheme when refining the data, and also when the Laplacian eigenmaps is employed.

Figure 8 illustrates the two-dimensional mappings of the these images obtained by LLE and multilevel-LLE. 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. This property is also more or less reflected in the plots by Isomap, multilevel-Isomap, Eigenmaps, and multilevel-Eigenmaps, which are not shown to save space.

The computation time is displayed in Table 2. Our multilevel technique reduced the computation time significantly for Isomap and LLE. For example, with $r = 2$ levels the savings for Isomap and LLE are more than 99% and 87%. For Eigenmaps there was no computation savings, since the cost for multilevel graph coarsening and data refining is comparable to that for dimensionality reduction.

Table 2: Computation time for `Frey Face` data set.

kNN time (secs)	level	average # of images	coarsen. time (secs)	Isomap proc. time	Isomap ref. time	LLE proc. time	LLE ref. time	Eigenmaps proc. time	Eigenmaps ref. time
1.38	#1	1965	N/A	393.67	N/A	14.22	N/A	0.3500	N/A
	#2	252.15	0.2328	0.4771	0.1493	0.2025	0.1461	0.0254	0.1472
	#3	47.01	0.0212	0.0115	0.0053	0.0225	0.0056	0.0173	0.0050
	#4	12.59	0.0026	0.0073	0.0007	0.0069	0.0010	0.0117	0.0020

Figure 9 displays the plots of trustworthiness, continuity, and H-score values as a function of p, the size of the neighborhood used in measuring them, where we set the number of levels up to four. Clearly the multilevel technique improved Isomap and LLE in both computation time and embedding quality, while multilevel-Eigenmaps performed comparable to Eigenmaps using this data set.

[2] `http://www.cs.toronto.edu/~roweis/data.html`

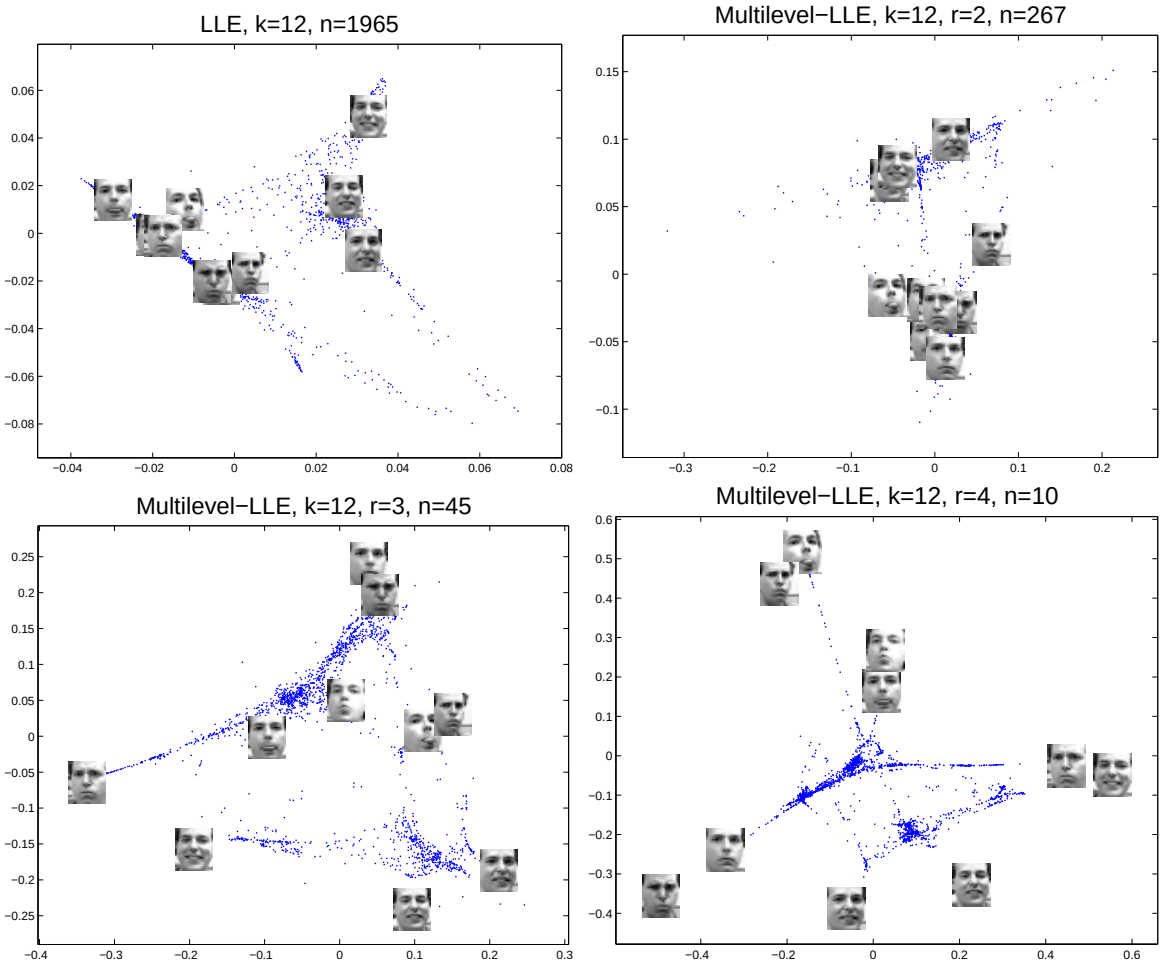


Figure 8: 2D mappings of `Frey Face` database using LLE and multilevel-LLE.

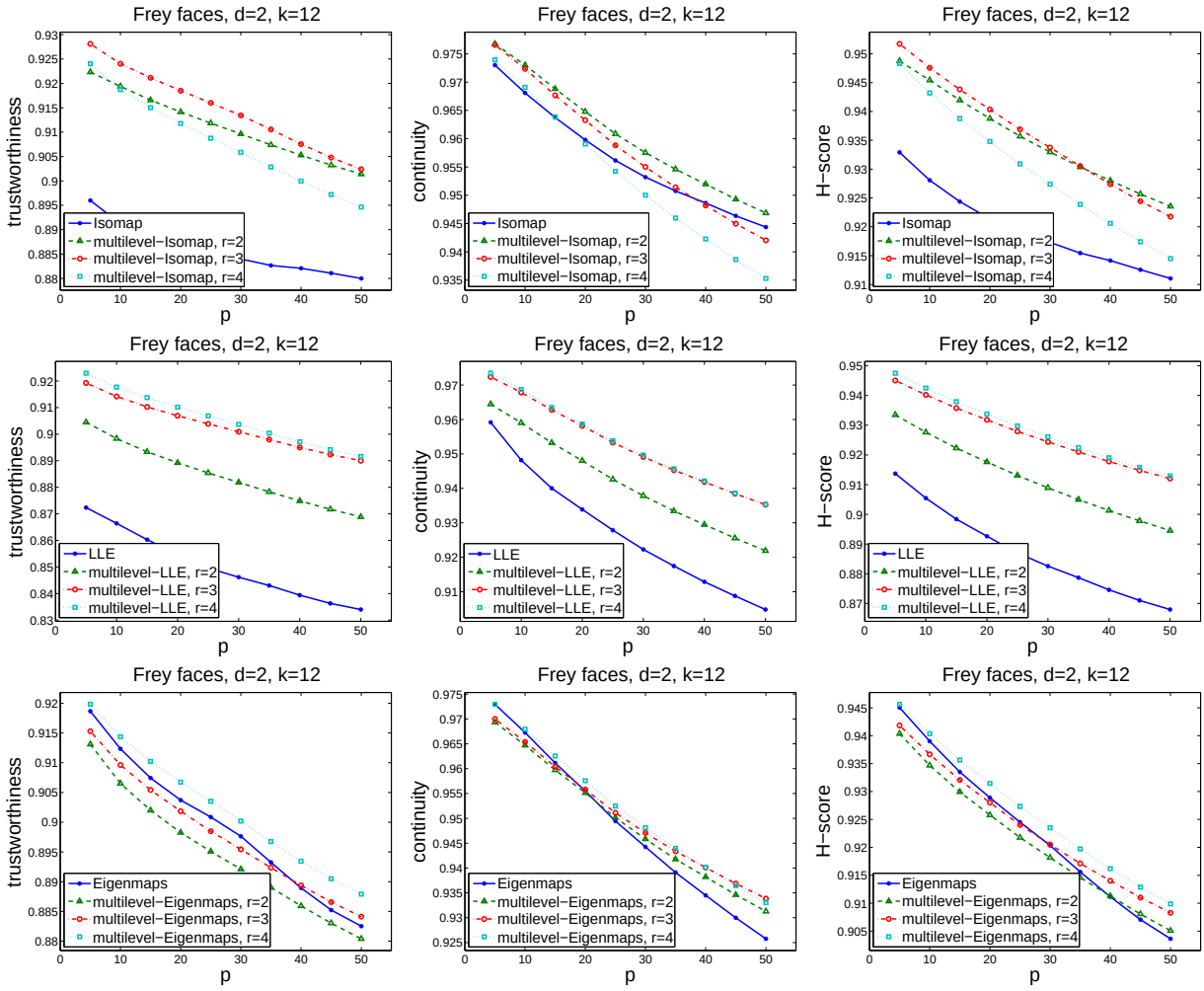


Figure 9: Trustworthiness, continuity, and H-score of `Frey Face` database by Isomap and multilevel-Isomap, LLE and multilevel-LLE, Eigenmaps and multilevel-Eigenmaps.

5.5 Rotating Teapot Images

The `Teapot` data set [27][3], generated by Jihun Ham, includes 400 images of size 76-by-101 pixels, with 3-byte color depth, giving rise to inputs of $p = 23,028$ dimensions. The images were created by viewing a teapot from different angles.

We report the results of experiments using a kNN graph with $k = 10$ and the embedding dimensions $d = 2$. In our multilevel framework we used the heat kernel weighting scheme when refining the data, and also when the Laplacian eigenmaps is employed.

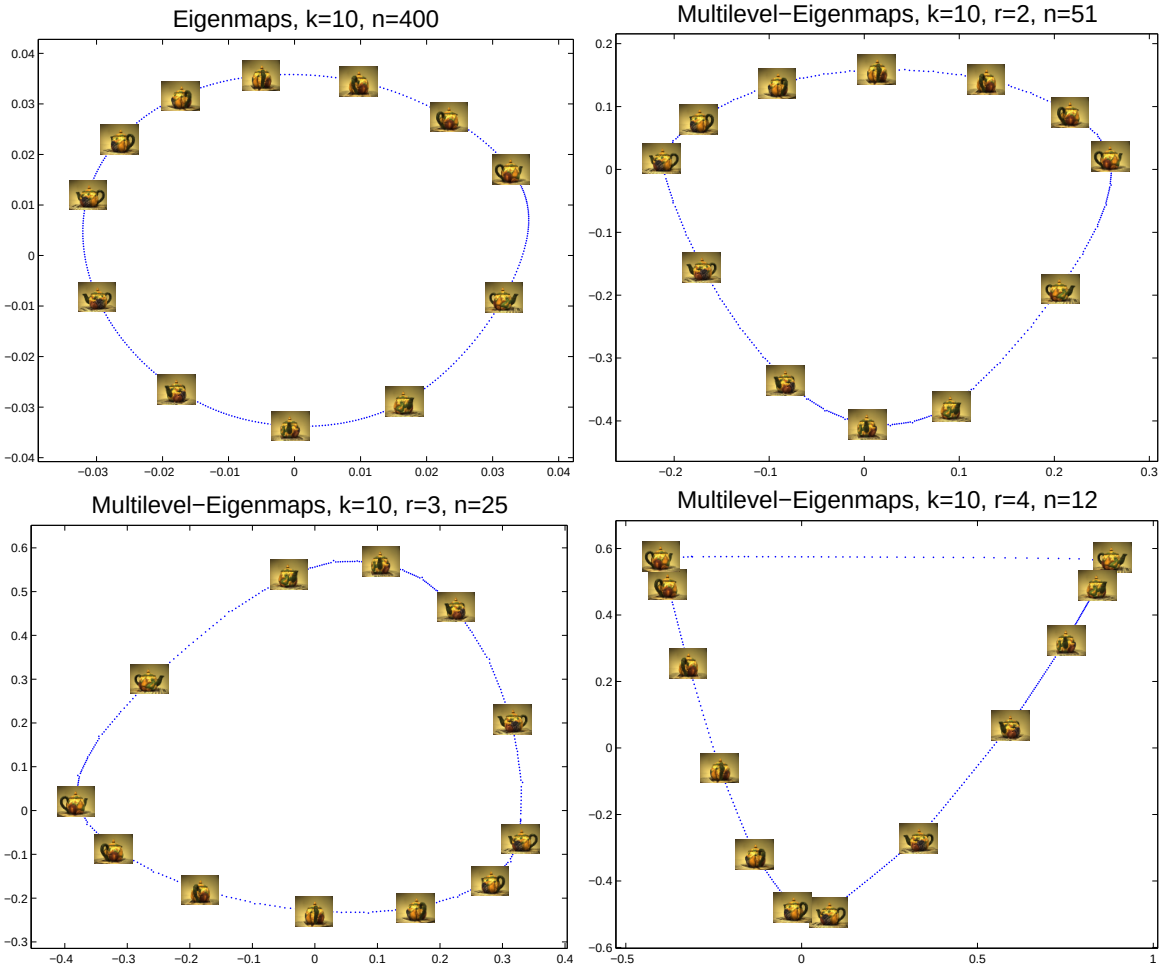


Figure 10: 2D mappings of `Teapot` data set using Eigenmaps and multilevel-Eigenmaps.

Figure 10 illustrates the two-dimensional mappings of the `Teapot` data set obtained by Eigenmaps and multilevel-Eigenmaps. Though very high dimensional, we observe that the images in this data set can be effectively parameterized by one degree of freedom, the angle of rotation. The mapped points in the two-dimensional space typically formed a round shape.

The computation time is displayed in Table 3. Using $r = 2$ levels, our multilevel technique achieved about 58% savings in computation time for Isomap, 66% savings for LLE, and

[3] `http://www.weinbergerweb.net/Downloads/Data.html`

insignificant (3%) savings for Eigenmaps.

Table 3: Computation time for `Teapot` data set.

kNN time (secs)	Level	average # of images	coarsen. time (secs)	Isomap proc. time	Isomap ref. time	LLE proc. time	LLE ref. time	Eigenmaps proc. time	Eigenmaps ref. time
1.67	#1	400	N/A	2.34	N/A	3.59	N/A	0.0800	N/A
	#2	50.35	0.0139	0.0124	0.0077	0.0916	0.0073	0.0155	0.0068
	#3	24.85	0.0014	0.0081	0.0008	0.0464	0.0011	0.0149	0.0013
	#4	12.30	0.0003	0.0053	0.0013	0.0239	0.0007	0.0109	0.0008

Figure 11 displays the plots of trustworthiness, continuity and H-score values as a function of p, the size of the neighborhood used in measuring them, where we set the number of levels up to four. Our multilevel-LLE and multilevel-Eigenmaps achieved comparable embedding quality with LLE and Eigenmaps, respectively. On the other hand, multilevel-Isomap, while achieve significant computational savings, performed less satisfactory than Isomap for this data set, in terms of the resulting embedding quality.

6 Clustering Experiments

We compare empirically the performance of three clustering algorithms discussed in Section 4.

1. K-means clustering with random initialization.
2. Nonlinear dimensionality reduction (e.g., Isomap, LLE, and Laplacian eigenmaps) and then K-means clustering with random initialization.
3. Multilevel dimensionality reduction joint with multilevel K-means clustering with random initialization at the bottom level.

Section 6.1 describes the methods to evaluate the quality of clusters. Sections 6.2 and 6.3 reports the results of experiments on `ORL` face database and `UMIST` face database, respectively.

6.1 Clustering Evaluation

We evaluate the quality of clusters by *purity* and *entropy* [28]:

$$\text{purity} = \sum_{i=1}^{K} \frac{n_i}{n} \text{purity}(i), \qquad \text{purity}(i) = \frac{1}{n_i} \max_j \left(n_i^j\right),$$

and

$$\text{entropy} = \sum_{i=1}^{K} \frac{n_i}{n} \text{entropy}(i), \qquad \text{entropy}(i) = -\sum_{j=1}^{K} \frac{n_i^j}{n_i} \log_K \frac{n_i^j}{n_i},$$

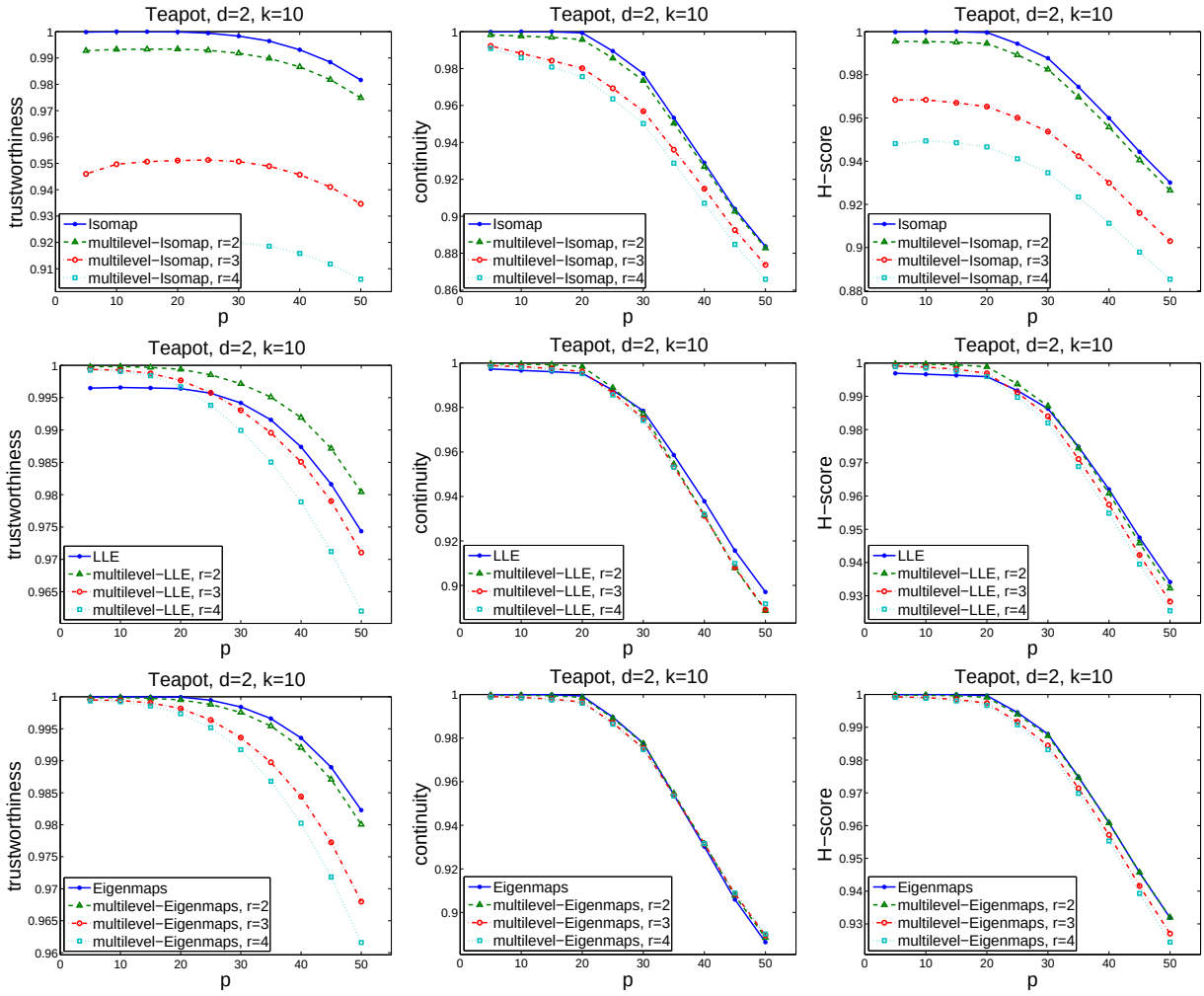


Figure 11: Trustworthiness, continuity and H-score of `Teapot` data set by Isomap and multilevel-Isomap, LLE and multilevel-LLE, Eigenmaps and multilevel-Eigenmaps.

where K is the number of clusters, n_i^j is the number of entries of class j in cluster i, and n_i is the number of data entries in cluster i. Note that we have assumed that each entry is associated with a label indicating the class to which it belongs.

Both purity and entropy are bounded between 0 and 1. The larger the purity, or the smaller the total entropy, the better the performance. The optimal value 0 of entropy and the optimal value 1 of purity are met, if and only if the clusters match exactly the classes.

6.2 ORL Face Database

We used `ORL` (Olivetti Research Laboratory) database [16] which contains 40 subjects each having 10 grayscale images of size 112-by-92 with various facial expressions (smiling/non-smiling, etc.), giving a total of 400 images. After vectorizing the images, we obtained a matrix X of size 10,304-by-400. Sample face images of the first two individuals are shown in Figure 12.

Figure 12: Sample ORL face images.

For the dimensionality reduction we constructed a kNN graph with $k = 5$ neighbors per data entry[4], and used the embedding dimensions $d = 10, 20, \ldots, 50$. For each method and each parameter setting, we report the average numbers of 100 random runs. The randomization applies the data coarsening, initialization for K-means clustering, and initialization at the bottom level of multilevel K-means clustering.

Figure 13 show the plots of purity and entropy values of the resulting clusters. Moreover, Table 4 reports the number of iterations, the CPU time, the purity and entropy for various clustering methods, where we set the embedding dimensions $d = 30$, and the number of iterations refers to the number of K-means iterations at topmost level. The CPU time includes the time for the kNN graph construction when a dimensionality reduction method was used, and includes graph coarsening and refining time when our multilevel technique was incorporated.

Table 4: Statistics of clustering results of `ORL` data set.

	K-means	K-means with dimensionality reduction ($d = 30$)					
		Isomap	m-level-Isomap	LLE	m-level-LLE	E-maps	m-level-E-maps
# iterations	7.46	8.57	6.70	10.63	6.52	9.34	6.43
CPU time	11.45	2.732	0.711	1.327	0.766	0.563	0.668
purity	0.641	0.700	0.729	0.680	0.727	0.694	0.725
entropy	0.206	0.169	0.151	0.182	0.150	0.168	0.155

As expected, a dimensionality reduction method may improve clustering quality in terms of both purity and entropy by removing redundancy and noise in the data. It may also reduce the computational cost by clustering data in a lower dimensional space. Further improvement can be made by structured initialization by our multilevel dimensionality reduction technique.

[4]This kNN graph with $k = 5$, after symmetrization, was disconnected with two components. We added the shortest edge between the vertices of the two components to make a connected graph.

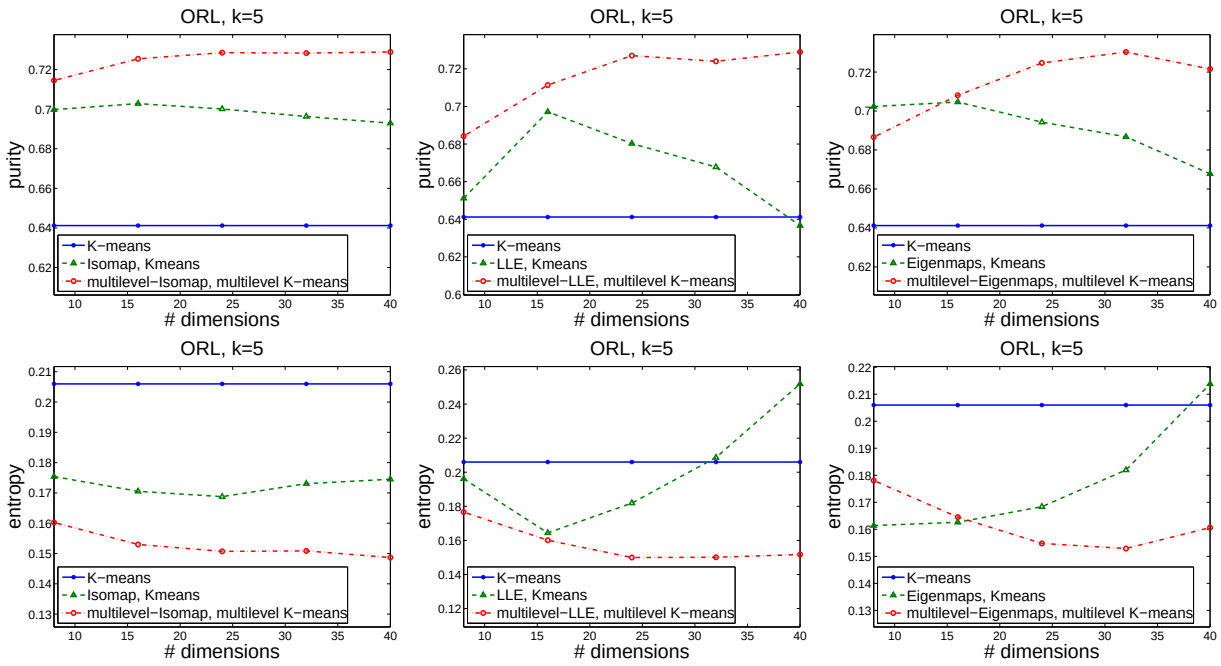


Figure 13: Purity and entropy values of clustering results of ORL data set.

6.3 UMIST Face Database

The UMIST database [8] contains 565 images in grayscale of 20 subjects with 19 to 48 images per subject. We used cropped images of size 112-by-92 in our experiments. Figure 14 shows sample images of the first individual.

Figure 14: Sample UMIST face images.

For the dimensionality reduction we constructed a kNN graph with $k = 7$ neighbors per data entry, and used the embedding dimensions $d = 5, 10, \ldots, 25$. For each method and each parameter setting, we report the average numbers of 100 random runs. The randomization applies the data coarsening, initialization for K-means clustering, and initialization at the bottom level of multilevel K-means clustering.

Figure 15 show the plots of purity and entropy values of the resulting clusters. Moreover, Table 5 reports the number of iterations, the CPU time, the purity and entropy for various

clustering methods, where we set the embedding dimensions $d = 15$, and the number of iterations refers to the number of K-means iterations at topmost level. The CPU time includes the time for kNN graph construction when a dimensionality reduction method was used, and includes graph coarsening and refining time when our multilevel technique was incorporated.

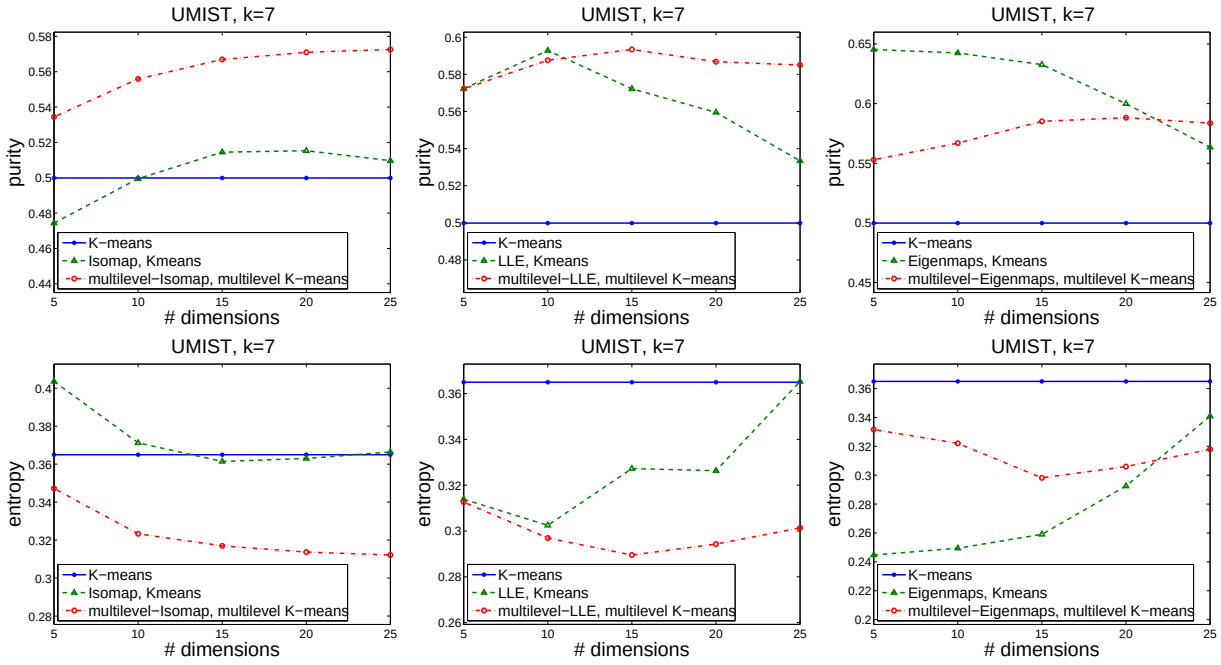


Figure 15: Purity and entropy values of clustering results of `UMIST` data set.

Table 5: Statistics of clustering results of `UMIST` data set.

	K-means	K-means with dimensionality reduction ($d = 30$)					
		Isomap	m-level-Isomap	LLE	m-level-LLE	E-maps	m-level-E-maps
# iterations	11.99	15.07	10.01	14.85	8.88	10.19	8.39
CPU time	12.13	10.976	0.637	2.149	0.744	0.451	0.569
purity	0.500	0.515	0.567	0.572	0.593	0.633	0.585
entropy	0.365	0.361	0.317	0.327	0.290	0.259	0.298

As can be observed, dimensionality reduction methods improved clustering quality by removing redundancy and noise in the data. The computational cost is also reduced because of clustering data in a lower dimensional space. Further improvement on the K-means clustering with Isomap and LLE dimensionality reduction methods was achieved by structured initialization by our multilevel dimensionality reduction technique.

7 Conclusion

The class of multilevel nonlinear dimension reduction techniques for manifold learning presented in this paper aim at reducing cost without sacrificing accuracy. As was observed a side benefit of performing a coarsening of the kNN graph is that it often leads to improved accuracy. This in effect shows that the combination of coarsening with standard manifold learning methods can be powerful. This remains to be explored further because the beneficial effect just mentioned is not seen for all methods. Experiments indicate that the proposed multilevel framework usually reduces the computational cost of some existing methods for manifold learning, while yielding comparable or better results. We have shown an application of the method to clustering, by incorporating the multilevel dimensionality reduction technique with the K-means algorithm for structured initialization. Experiments show that this often results in a significant improvement in clustering quality and in computational time.

References

[1] S. T. Barnard and H. D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and Experience*, 6:101–107, 1994.

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

[3] T. Chan, B. Smith, and J. Zou. Multigrid and domain decomposition methods for unstructured meshes. In *Third International Conference on Advances in Numerical Methods and Applications*, pages 53–62, Sofia, Bulgaria, 1994.

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

[5] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, Cambridge, MA, 2003.

[6] 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.

[7] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, 1962.

[8] D. B. Graham and N. M. Allinson. Face recognition: From theory to applications. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, editors, *NATO ASI Series F, Computer and Systems Sciences, Vol. 163*, pages 446–456, 1998.

[9] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2nd edition edition, 2009.

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

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

[12] 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 Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.

[13] M. H. C. Law and A. K. Jain. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(3):377–391, 2006.

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

[15] S. Sakellaridi, H.-r. Fang, and Y. Saad. Graph-based multilevel dimensionality reduction with applications to eigenfaces and latent semantic indexing. In *ICMLA '08: Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications*, pages 194–200, Washington, DC, USA, 2008. IEEE Computer Society.

[16] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[17] L. K. Saul and S. T. Rowies. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J. of Machine Learning Research*, 4:119–155, 2003.

[18] L. K. Saul, K. Q. Weinberger, J. H. Ham, F. Sha, and D. D. Lee. Spectral methods for dimensionality reduction. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semisupervised Learning*. MIT Press, Cambridge, MA, 2006.

[19] F. Sha and L. K. Saul. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the 22nd international conference on Machine learning (ICML'05)*, pages 784–791, New York, NY, USA, 2005. ACM.

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

[21] G. W. Stewart and J.-g. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

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

[23] J. Venna and S. Kaski. Neighborhood preservation in nonlinear production methods: An experimental study. In *ICANN, International Conference on Artificial Neural Networks*, 2001.

[24] J. Venna and S. Kaski. Local multidimensional scaling. *Neural Networks*, 19:889–899, 2006.

[25] S. Warshall. A theorem on boolean matrices. *J. ACM*, 9(1):11–12, 1962.

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

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

[28] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.