Isometric Projection

Deng Cai
Computer Science Department
University of Illinois at Urbana-Champaign
dengcai2@cs.uiuc.edu

Xiaofei He
Yahoo! Research Labs
hex@yahoo-inc.com

Jiawei Han
Computer Science Department
University of Illinois at Urbana-Champaign
hanj@cs.uiuc.edu

Abstract

Recently the problem of dimensionality reduction has received a lot of interests in many fields of information processing. We consider the case where data is sampled from a low dimensional manifold which is embedded in high dimensional Euclidean space. The most popular manifold learning algorithms include Locally Linear Embedding, ISOMAP, and Laplacian Eigenmap. However, these algorithms are nonlinear and only provide the embedding results of training samples. In this paper, we propose a novel linear dimensionality reduction algorithm, called **Isometric Projection**. Isometric Projection constructs a weighted data graph where the weights are discrete approximations of the geodesic distances on the data manifold. A linear subspace is then obtained by preserving the pairwise distances. In this way, Isometric Projection can be defined everywhere. Comparing to Principal Component Analysis (PCA) which is widely used in data processing, our algorithm is more capable of discovering the intrinsic geometrical structure. Specially, PCA is optimal only when the data space is linear, while our algorithm has no such assumption and therefore can handle more complex data space. Experimental results on two real life data sets illustrate the effectiveness of the proposed method.

Introduction

In many real world applications, such as information retrieval, face recognition, bioinformatics, etc., one is often confronted with high-dimensional data. However, there might be reason to suspect that the naturally generated high-dimensional data probably reside on a lower dimensional manifold. This leads one to consider methods of dimensionality reduction that allow one to represent the data in a lower dimensional space.

One of the most popular dimensionality reduction algorithms might be Principal Component Analysis (PCA) (Duda, Hart, & Stork 2000). PCA performs dimensionality reduction by projecting the original n-dimensional data onto the $d(\ll n)$-dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data so that the pairwise *Euclidean* distances can be best preserved. If the data is embedded in a linear subspace, PCA is guaranteed to discover the dimensionality of the subspace and produces a compact representation.

In many real world problems, however, there is no evidence that the data is sampled from a linear subspace. For example, it is always believed that the face images are sampled from a nonlinear low-dimensional manifold which is embedded in the high-dimensional ambient space (He *et al.* 2005b). This motivates us to consider manifold based techniques for dimensionality reduction. Recently, various manifold learning techniques, such as ISOMAP (Tenenbaum, de Silva, & Langford 2000), Locally Linear Embedding (LLE) (Roweis & Saul 2000) and Laplacian Eigenmap (Belkin & Niyogi 2001) have been proposed which reduce the dimensionality of a *fixed* training set in a way that maximally preserve certain inter-point relationships. LLE and Laplacian Eigenmap are local methods which attempt to preserve local geometry of the data; essentially, they seek to map nearby points on the manifold to nearby points in the low-dimensional representation. ISOMAP is a global method which attempts to preserve geometry at all scales, mapping nearby points on the manifold to nearby points in low-dimensional space, and faraway points to faraway points. One of the major limitations of these methods is that they do not generally provide a functional mapping between the high and low dimensional spaces that are valid both on and off the training data.

There are some approaches that try to address this issue by explicitly defining an embedding function either linear or in reproducing kernel Hilbert space (RKHS) when minimizing the objective function (He & Niyogi 2003), (He *et al.* 2005a). They provide natural out-of-sample extensions of Lapalcian Eigenmaps and LLE. However, when the number of features is larger than the number of samples, these algorithms need to apply Singular Value Decomposition (SVD) first to get the stable solution of the optimization problems. Due to the high computational cost of SVD, these algorithms may not be applied to very high dimensional data with large size.

In this paper, we propose a novel dimensionality reduction algorithm called **Isometric Projection** (IsoProjection), which explicitly takes into account the manifold structure. To model the manifold structure, we first construct a nearest neighbor graph of the observed data. We then compute shortest paths in the graph for all pairs of data points. The shortest-paths computation gives an estimate of the global metric structure. Using techniques from Multi-Dimensional Scaling (MDS) and requiring the mapping function to be linear, we obtain the objective function of Isometric Projection. Finally, the optimization problem can be efficiently solved by techniques from spectral graph analysis and regression, which leads to Isometric Projection.

The points below highlight several aspects of the paper:

1. IsoProjection provides an optimal linear approximation to the true isometric embedding of the underlying data manifold. It tends to give a more faithful representation of the data's global structure than PCA does.
2. IsoProjection is defined everywhere. Therefore, query points can also be mapped into the low-dimensional representation space in which retrieval, clustering and classification may be performed.
3. While the linear versions of Laplacian Eigenmaps (He & Niyogi 2003) and LLE (He *et al.* 2005a) need to apply SVD first which can be very computational expensive, IsoProjection is computed by using spectral graph analysis and regression which are very efficient even for high dimensional data of large size.
4. IsoProjection is fundamentally based on ISOMAP (Tenenbaum, de Silva, & Langford 2000), but ISOMAP does not have properties (2) above.

Background

In this section, we provide mathematical background of manifold based dimensionality reduction. For a detailed treatment of manifolds, please see (Lee 2002).

Data are generally represented as points in high-dimensional vector space. For example, a 32 × 32 image can be represented by a 1024-dimensional vector. Every element of the vector corresponds to a pixel. A text document can be represented by a term vector. In many cases of interests, the data may not fill the whole ambient space, but reside on or near a submanifold embedded in the ambient space. One hopes then to estimate geometrical and topological properties of the submanifold from random samples ("scattered data") lying on this unknown submanifold. The formal definition of manifold is as follows.

Definition An p-dimensional manifold (denoted by $\mathcal{M}^p$) is a topological space that is locally Euclidean. That is, around every point, there is a neighborhood that is topologically the same as the open unit ball in $\mathbb{R}^p$.

In order to compute distances on the manifold, one needs to equip a metric to the topological manifold. A manifold possessing a metric is called *Riemannian Manifold*, and the metric is commonly referred to as *Riemannian Metric*.

Definition Suppose for every point $\mathbf{x}$ in a manifold $\mathcal{M}$, an inner product $\langle\cdot,\cdot\rangle_x$ is defined on a tangent space $T_x\mathcal{M}$ of $\mathcal{M}$ at $\mathbf{x}$. Then the collection of all these inner products is called the Riemannian metric.

Once the Riemannian metric is defined, one is allowed to measure the lengths of the tangent vectors $\mathbf{v} \in T_x\mathcal{M}$:

$$\|\mathbf{v}\|^2 = \langle \mathbf{v}, \mathbf{v}\rangle$$

For every smooth curve $r : [a, b] \rightarrow \mathcal{M}$, we have tangent vectors:

$$r'(t) = \frac{dr}{dt} \in T_{r(t)}\mathcal{M}$$

and can therefore use the Riemannian metric (inner product of the tangent spaces) to define their lengths. We can then define the length of r from a to b:

$$length(r) = \int_a^b \|\frac{dr}{dt}\|dt = \int_a^b \|r'(t)\|dt$$

Note that, a Riemannian metric is not a distance metric on $\mathcal{M}$. However, for a connected manifold, it is the case that every Riemannian metric induces a distance matric on $\mathcal{M}$, i.e. *Geodesic Distance*.

Definition The geodesic distance $d_{\mathcal{M}}(a, b)$ is defined as the length of the shortest curve connecting a and b.

In the plane, the geodesics are straight lines. On the sphere, the geodesics are great circles (like the equator). Suppose $\mathcal{M}^p$ is embedded in a n-dimensional Euclidean space $\mathbb{R}^n$ $(p \leq n)$. Let us consider a low dimensional map, $f : \mathbb{R}^n \rightarrow \mathbb{R}^d (d \leq n)$, and the f has a support on a submanifold $\mathcal{M}^p$, i.e. $supp(f) = \mathcal{M}^p$. Note that, $p \leq d \leq n$, and p is generally unknown. Let d_{R^d} denote the standard Euclidean distance measure in $\mathbb{R}^d$. In order to preserve the intrinsic (invariant) geometrical structure of the data manifold, we seek a function f such that:

$$d_{\mathcal{M}^p}(\mathbf{x}, \mathbf{y}) = d_{R^d}(f(\mathbf{x}), f(\mathbf{y})) \qquad (1)$$

In this paper, we are particularly interested in linear mappings, i.e. projections. The reason is for its simplicity. And more crucially, the same derivation can be performed in reproducing kernel Hilbert space (RKHS) which naturally leads to its nonlinear extension (Cai, He, & Han 2006).

Isometric Projection

In this section, we introduce a novel dimensionality reduction algorithm, called Isometric Projection. We begin with a formal definition of the problem of dimensionality reduction.

The Problem

The generic problem of dimensionality reduction is the following. Given a set of points $\mathbf{x}_1, \cdots, \mathbf{x}_m$ in $\mathbb{R}^n$, find a mapping function that maps these m points to a set of points $\mathbf{y}_1, \cdots, \mathbf{y}_m$ in $\mathbb{R}^d$ $(d << n)$, such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$, where $\mathbf{y}_i = f(\mathbf{x}_i)$. Our method is of particular applicability in the special case where $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m \in \mathcal{M}$ and $\mathcal{M}$ is a nonlinear manifold embedded in $\mathbb{R}^n$.

The Objective Function

We define $X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m)$. Let $d_{\mathcal{M}}$ be the geodesic distance measure on $\mathcal{M}$ and d the standard Euclidean distance measure in $\mathbb{R}^d$. Isometric Projection aims to find a embedding function f such that Euclidean distances in $\mathbb{R}^d$

can provide a good approximation to the geodesic distances on $\mathcal{M}$. That is,

$$f^{opt} = \arg\min_{f} \sum_{i,j} \Big(d_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j) - d\big(f(\mathbf{x}_i), f(\mathbf{x}_j)\big) \Big)^2 \quad (2)$$

In real life data set, the underlying manifold $\mathcal{M}$ is often unknown and hence the geodesic distance measure is also unknown. In order to discover the intrinsic geometrical structure of $\mathcal{M}$, we first construct a graph G over all data points to model the local geometry. There are two choices:

1. ϵ-graph: we put an edge between i and j if $d(\mathbf{x}_i, \mathbf{x}_j) < \epsilon$.
2. kNN-graph: we put an edge between i and j if $\mathbf{x}_i$ is among k nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among k nearest neighbors of $\mathbf{x}_i$.

Once the graph is constructed, the geodesic distances $d_{\mathcal{M}}(i, j)$ between all pairs of points on the manifold $\mathcal{M}$ can be estimated by computing their shortest path distances $d_G(i, j)$ on the graph G. The procedure is as follows: initialize $d_G(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked by an edge; $d_G(\mathbf{x}_i, \mathbf{x}_j) = \infty$ otherwise. Then for each value of $p = 1, 2, \cdots, m$ in turn, replace all entries $d_G(\mathbf{x}_i, \mathbf{x}_j)$ by

$$\min\big\{ d_G(\mathbf{x}_i, \mathbf{x}_j),\ d_G(\mathbf{x}_i, \mathbf{x}_p) + d_G(\mathbf{x}_p, \mathbf{x}_j) \big\}.$$

The matrix of final values $D_G = \{d_G(\mathbf{x}_i, \mathbf{x}_j)\}$ will contain the shortest path distances between all pairs of points in G. This procedure is named Floyd-Warshall algorithm (Cormen *et al.* 2001).

In the following, we apply techniques from Multi-Dimensional Scaling (MDS) to convert distances to inner products, which uniquely characterize the geometry of the data in a form that supports efficient optimization (Mardia, Kent, & Bibby 1980),(Tenenbaum, de Silva, & Langford 2000).

Specifically, let D be the distance matrix such that D_{ij} is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Define matrix $S_{ij} = D_{ij}^2$ and $H = I - \frac{1}{m}\mathbf{e}\mathbf{e}^T$ where I is the identity matrix and $\mathbf{e}$ is the vector of all ones. It can be shown that $\tau(D) = -HSH/2$ is the inner product matrix. That is, $D_{ij}^2 = \tau(D)_{ii} + \tau(D)_{jj} - 2\tau(D)_{ij}, \forall\ i, j$ (Mardia, Kent, & Bibby 1980).

The matrix H is often called "centering matrix". Let D_Y denote the Euclidean distance matrix in the reduced subspace, and $\tau(D_Y)$ be the corresponding inner product matrix. Thus, the objective function (2) becomes minimizing the following:

$$\|\tau(D_G) - \tau(D_Y)\|_{L^2} \quad (3)$$

where $\|A\|_{L^2}$ is the L^2 matrix norm $\sqrt{\sum_{i,j} A_{i,j}^2}$.

Learning Isometric Projections

Consider a linear function $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$. Let $y_i = f(\mathbf{x}_i)$ and $Y = (y_1, \cdots, y_m) = \mathbf{a}^T X$. Thus, we have

$$\tau(D_Y) = Y^T Y = X^T \mathbf{a}\mathbf{a}^T X$$

The optimal projection is given by solving the following minimization problem:

$$\mathbf{a}^* = \min_{\mathbf{a}} \|\tau(D_G) - X^T\mathbf{a}\mathbf{a}^T X\|^2 \quad (4)$$

Following some algebraic steps and noting $tr(A) = tr(A^T)$, we see that:

$$\begin{aligned} & \|\tau(D_G) - X^T\mathbf{a}\mathbf{a}^T X\|^2 \\ = \;& tr\Big(\Big(\tau(D_G) - X^T\mathbf{a}\mathbf{a}^T X\Big)\Big(\tau(D_G) - X^T\mathbf{a}\mathbf{a}^T X\Big)^T \Big) \\ = \;& tr\Big(\tau(D_G)\tau(D_G)^T - X^T\mathbf{a}\mathbf{a}^T X\tau(D_G)^T - \\ & \quad \tau(D_G)X^T\mathbf{a}\mathbf{a}^T X + X^T\mathbf{a}\mathbf{a}^T X X^T\mathbf{a}\mathbf{a}^T X\Big) \end{aligned}$$

Note that, the magnitude of $\mathbf{a}$ is of no real significance because it merely scales y_i. Therefore, we can impose a constraint as follows:

$$\mathbf{a}^T X X^T \mathbf{a} = 1$$

Thus, we have

$$tr\Big(X^T\mathbf{a}\mathbf{a}^T X X^T\mathbf{a}\mathbf{a}^T X\Big) = tr\Big(\mathbf{a}^T X X^T\mathbf{a}\mathbf{a}^T X X^T\mathbf{a}\Big) = 1$$

And,

$$\begin{aligned} & \|\tau(D_G) - X^T\mathbf{a}\mathbf{a}^T X\|^2 \\ = \;& tr\Big(\tau(D_G)\tau(D_G)^T\Big) - 2tr\Big(\mathbf{a}^T X\tau(D_G)X^T\mathbf{a}\Big) + 1 \end{aligned}$$

Now, the minimization problem (4) can be written as follows:

$$\arg\max_{\mathbf{a}^T X X^T\mathbf{a}=1} \ \mathbf{a}^T X\tau(D_G)X^T\mathbf{a}. \quad (5)$$

The vectors $\mathbf{a}_i (i = 1, 2, \cdots, l)$ that minimize the above objective function are given by the eigenvectors corresponding to the maximum eigenvalues of the generalized eigen-problem:

$$X[\tau(D_G)]X^T\mathbf{a} = \lambda X X^T\mathbf{a} \quad (6)$$

Let $A = [\mathbf{a}_1, \cdots, \mathbf{a}_l]$, the linear embedding is as follows:

$$\mathbf{x} \rightarrow \mathbf{y} = A^T\mathbf{x}$$

where $\mathbf{y}$ is a l-dimensional representation of the high dimensional data point $\mathbf{x}$. A is the transformation matrix.

To get a stable solution of eigen-problem (6), the matrix XX^T is required to be non-singular (Stewart 2001) which is not true when the number of features is larger than the number of samples. The Singular Value Decomposition (SVD) can be used to solve this problem. Suppose $rank(X) = r$, the SVD decomposition of X is

$$X = U\Sigma V^T$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_r)$ and $\sigma_1 \geq \cdots \geq \sigma_r > 0$ are the singular values of X, $U \in \mathbb{R}^{n\times r}$, $V = \in \mathbb{R}^{m\times r}$ and $U^T U = V^T V = I$. Let $\widetilde{X} = U^T X = \Sigma V^T$ and $\mathbf{b} = U^T\mathbf{a}$, we have

$$\mathbf{a}^T X\tau(D_G)X^T\mathbf{a} = \mathbf{a}^T U\Sigma V^T\tau(D_G)V\Sigma U^T\mathbf{a} = \mathbf{b}^T\widetilde{X}\tau(D_G)\widetilde{X}^T\mathbf{b}$$

and

$$\mathbf{a}^T X X^T\mathbf{a} = \mathbf{a}^T U\Sigma V^T V\Sigma U^T\mathbf{a} = \mathbf{b}^T\widetilde{X}\widetilde{X}^T\mathbf{b}.$$

Now, the objective function of IsoProjection in (5) can be rewritten as:

$$\arg\max_{\mathbf{b}^T\widetilde{X}\widetilde{X}^T\mathbf{b}=1} \ \mathbf{b}^T\widetilde{X}\tau(D_G)\widetilde{X}^T\mathbf{b}.$$

and the optimal **b**'s are the maximum eigenvectors of eigen-problem:

$$\widetilde{X}\tau(D_G)\widetilde{X}^T\mathbf{b} = \lambda\widetilde{X}\widetilde{X}^T\mathbf{b}. \tag{7}$$

It is easy to check that $\widetilde{X}\widetilde{X}^T = \Sigma^2$ is nonsingular and the eigen-problem can be stably solved. After we get **b**, the **a** can be obtained by

$$\mathbf{a} = U\mathbf{b}.$$

Efficient Computation with Regression

The above linear extension approach has also been applied on Laplacian Eigenmaps and LLE which leads to Locality Preserving Projection (LPP) (He & Niyogi 2003) and Neighborhood Preserving Embedding (NPE) (He *et al.* 2005a). However, when the number of features (n) is larger than the number of samples (m), the high computational cost of SVD makes this approach unlikely to be applied on large scale high dimensional data.

Let us analyze the computational complexity of this linear extension approach. We use the term *flam* (Stewart 1998), a compound operation consisting of one addition and one multiplication, to present operation counts. The most efficient algorithm to calculate the SVD decomposition requires $\frac{3}{2}m^2n + \frac{9}{2}m^3$ flam (Stewart 2001). When $n > m$, the rank of X is usually of m. Thus, $\widetilde{X}$ is a square matrix of size $m \times m$. The calculation of matrices $\widetilde{X}\tau(D_G)\widetilde{X}^T$ requires $2m^3$ flam. The top l eigenvectors of eigen-problem (7) can be calculated within plm^2 flam by Lanczos algorithm, where p is the number of iterations in Lanczos (usually less than 20) (Stewart 2001). Thus, the time complexity of the the linear extension approach measured by flam is

$$\frac{3}{2}m^2n + \frac{13}{2}m^3 + plm^2, \tag{8}$$

which is cubic-time complexity with respect to m. For large scale high dimensional data, this approach is unlikely to be applied.

In order to solve the eigen-problem (6) efficiently, we use the following theorem:

Theorem 1 *Let **y** be the eigenvector of $\tau(D_G)$ with eigenvalue λ. If X^T**a** = **y**, then **a** is the eigenvector of eigen-problem in Eqn. (6) with the same eigenvalue λ.*

Proof We have $\tau(D_G)\mathbf{y} = \lambda\mathbf{y}$. At the left side of Eqn. (6), replace $X^T\mathbf{a}$ by **y**, we have

$$X\tau(D_G)X^T\mathbf{a} = X\tau(D_G)\mathbf{y} = X\lambda\mathbf{y} = \lambda X\mathbf{y} = \lambda XX^T\mathbf{a}$$

Thus, **a** is the eigenvector of eigen-problem (6) with the same eigenvalue λ.

Theorem (1) shows that instead of solving the eigen-problem in Eqn. (6), the linear projective functions can be obtained through two steps:

1. Compute the eigenvector of $\tau(D_G)$, **y**.
2. Find **a** which satisfies $X^T\mathbf{a} = \mathbf{y}$. In reality, such **a** might not exist. A possible way is to find **a** which can best fit the equation in the least squares sense:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \sum_{i=1}^{m} (\mathbf{a}^T\mathbf{x}_i - y_i)^2 \tag{9}$$

where y_i is the i-th element of **y**.

In the situation that the number of samples is smaller than the number of features, the minimization problem (9) is *ill posed*. We may have infinitely many solutions to the linear equations system $X^T\mathbf{a} = \mathbf{y}$. The most popular way to solve this problem is to impose a penalty on the norm of **a**:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \left(\sum_{i=1}^{m} \left(\mathbf{a}^T\mathbf{x}_i - y_i\right)^2 + \alpha\|\mathbf{a}\|^2 \right) \tag{10}$$

This is so called regularization and is well studied in statistics and the $\alpha \geq 0$ is a parameter to control the amounts of shrinkage. The regularized least squares in Eqn. (10) can be rewritten in the matrix form as:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \left((X^T\mathbf{a} - \mathbf{y})^T(X^T\mathbf{a} - \mathbf{y}) + \alpha\mathbf{a}^T\mathbf{a} \right).$$

Requiring the derivative of right side with respect to **a** vanish, we get

$$\begin{aligned} &(XX^T + \alpha I)\mathbf{a} = X\mathbf{y} \\ \Rightarrow\ &\mathbf{a} = (XX^T + \alpha I)^{-1}X\mathbf{y} \end{aligned} \tag{11}$$

When $\alpha > 0$, this regularized solution will not satisfy the linear equations system $X^T\mathbf{a} = \mathbf{y}$ and **a** will not be the eigenvector of eigen-problem in Eqn. (6). It is interesting and important to see when (11) gives the exact solutions of eigen-problem (6). Specifically, we have the following theorem:

Theorem 2 *Let **y** be the eigenvector of $\tau(D_G)$, if **y** is in the space spanned by row vectors of X, the corresponding projective function **a** calculated in Eqn. (11) will be the eigenvector of eigen-problem in Eqn. (6) as α deceases to zero.*

Proof The proof is omitted due to the space limitation. A complete proof will be provided in a technical report.

When the the number of features is larger than the number of samples, the sample vectors are usually linearly independent, *i.e.*, $rank(X) = m$. In this case, any vector **y** is in the space spanned by row vectors of X. Thus, all the projective functions calculated in Eqn. (11) are the eigenvectors of eigen-problem in Eqn. (6) as α deceases to zero.

The top l eigenvectors of $\tau(D_G)$ can be calculated within plm^2 flam by Lanczos algorithm. The regularized least squares in Eqn. (10) can be efficiently solved by the iterative algorithm LSQR within $sl(2mn + 3m + 5n)$ flam, where s is the number of iterations in LSQR (usually less than 30) (Paige & Saunders 1982). The total time complexity measured by flam is

$$sl(2mn + 3m + 5n) + plm^2, \tag{12}$$

which is a significant improvement over the cost in Eqn. (8). It would be important to note that the cost in Eqn. (8) is exactly the cost of LPP (He & Niyogi 2003) and NPE (He *et al.* 2005a).

Table 1: Clustering accuracy (%) and learning time (s) on Reuters-21578

k	Baseline	LSI		LPP		NPE		IsoProjection	
	AC	AC	Time	AC	Time	AC	Time	AC	Time
2	87.6	88.2	0.044	91.9	0.21	83.1	0.23	**92.8**	**0.022**
3	77.8	78.5	0.066	79.5	0.41	75.7	0.45	**81.6**	**0.041**
4	72.0	72.6	0.100	75.5	0.85	70.1	0.89	**77.5**	**0.074**
5	69.8	70.6	0.175	**73.2**	1.76	65.0	1.81	72.4	**0.123**
6	66.6	67.2	0.256	69.5	3.06	54.6	3.12	**70.2**	**0.199**
7	64.2	64.5	0.327	**69.6**	4.02	52.9	4.10	68.7	**0.250**
8	56.4	57.4	0.426	**61.7**	5.23	44.7	5.33	60.9	**0.337**
9	55.6	55.8	0.544	**63.7**	6.78	46.2	6.88	61.6	**0.420**
10	53.9	54.6	0.677	56.9	9.41	41.1	9.52	**57.9**	**0.535**
Ave.	67.1	67.7	0.291	71.3	3.53	59.3	3.59	**71.5**	**0.222**

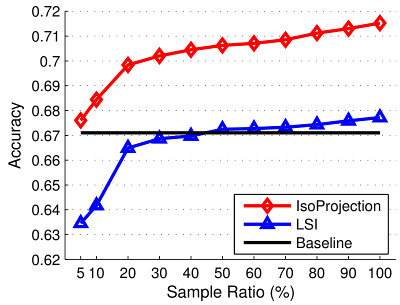

Figure 1: Generalization capability of IsoProjection.

Experimental Results

In this subsection, we investigate the use of IsoProjection for clustering and classification. Several popular linear dimensionality reduction algorithms are compared, which include Latent Semantic Indexing (LSI) (Deerwester *et al.* 1990), PCA (Duda, Hart, & Stork 2000), LDA (Belhumeur, Hepanha, & Kriegman 1997), LPP (He & Niyogi 2003) and NPE (He *et al.* 2005a). In the reduced subspace, the ordinary clustering and classification algorithms can then be used. In our experiments, we choose k-means for clustering and nearest neighbor classifier for classification for their simplicity.

Clustering on Reuters-21578

Reuters-21578 corpus contains 21578 documents in 135 categories. In our experiments, we discarded those documents with multiple category labels, and selected the largest 30 categories. It left us with 8,067 documents. Each document is represented as a term-frequency vector and each vector is normalized to 1.

The evaluations were conducted with different numbers of clusters. For each given class number $k(= 2 \sim 10)$, k classes were randomly selected from the document corpus. The documents and the cluster number k are provided to the clustering algorithms. The clustering result is evaluated by comparing the obtained label of each document with that provided by the document corpus. The accuracy (AC) is used to measure the clustering performance. Given a document $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}$$

where n is the total number of documents and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(r_i) is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm (Lovasz & Plummer 1986).

This process were repeated 25 times, and the average performance was computed. For each single test (given k classes of documents), the k-means step was repeated 10 times with different initializations and the best result in terms of the objective function of k-means was recorded. Table 1 shows the best performance as well as the time on learning the subspace for each algorithm.

As can be seen, our algorithm achieves similar performance to LPP, both of which consistently outperformed LSI, NPE and the baseline. For the baseline method, the clustering is simply performed in the original document space without any dimensionality reduction. Moreover, IsoProjection is the most efficient algorithm. This makes IsoProjection can be applied on large scale high dimensional data.

Generalization capability The advantage of IsoProjection over Isomap is that it has an explicit mapping function defined everywhere. For all these dimensionality reduction algorithms, learning the low dimensional representation is time consuming and the computational complexity scales with the number of data points. Since IsoProjection has explicit mapping functions, we can choose part of the data to learn a mapping function and use this mapping function to map the rest of data points to the reduced space. In this way, the computational complexity can be significantly reduced. It is hard for Isomap to adopt such technique since Isomap does not have the mapping function.

To demonstrate the generalization capability of IsoProjection, we designed the following experiment: For each test in the previous experiments, we only chose part of the data points (training set) to learn a mapping function. This mapping function is then used to map the rest of data points to the reduced space in which clustering is performed. The size of the training set ranged from 5% to 90% of the data set.

The average accuracy (averaged over 2∼10 classes) is shown in Fig. (1). It is clear that the performance improves with the number of training samples. Both IsoProjection and LSI have good generalization capability, however, there is no significant performance improvement of LSI over baseline which makes LSI less practical. For IsoProjection, it achieved similar performance to that using all the samples when only 30% of training samples were used. This makes it practical for clustering large sets of documents.

Face Recognition on Yale-B

The Extended Yale-B face database contains 16128 images of 38 human subjects under 9 poses and 64 illumination conditions (Lee, Ho, & Kriegman 2005). In this experiment, we

Table 2: Recognition accuracy (%) and learning time (s) on the extended Yale-B database

Train/Test	Baseline	PCA		LDA		LPP		NPE		IsoProjection	
	AC	AC	Time	AC	Time	AC	Time	AC	Time	AC	Time
G5/P59	36.6	36.6 (189)	0.095	76.0 (37)	0.111	75.6 (37)	0.164	**76.2** (37)	0.151	75.2 (37)	**0.084**
G10/P54	53.4	53.4 (379)	0.480	86.9 (37)	0.512	86.7 (37)	0.910	87.0 (37)	0.837	**87.7** (37)	**0.105**
G20/P44	69.2	69.2 (759)	3.048	89.6 (37)	3.116	91.2 (198)	5.960	89.6 (37)	5.777	**94.9** (37)	**0.234**
G30/P34	77.4	77.4 (900)	6.787	87.0 (37)	7.339	88.5 (215)	13.98	85.6 (477)	14.50	**97.9** (37)	**0.423**
G40/P24	81.9	81.9 (900)	6.974	95.5 (37)	7.697	96.6 (173)	14.77	91.9 (500)	15.41	**98.7** (37)	**0.677**
G50/P14	84.2	84.2 (1000)	7.124	97.8 (37)	8.036	98.4 (277)	15.46	91.7 (493)	16.14	**99.4** (37)	**1.042**

choose the frontal pose and use all the images under different illumination, thus we get 64 images for each person. All the face images are manually aligned and cropped. The cropped images are 32×32 pixels, with 256 gray levels per pixel. The image set is then partitioned into the gallery and probe set with different numbers. For ease of representation, Gp/Pq means p images per person are randomly selected for training and the remaining q images are for testing.

In general, the performance of all these methods varies with the number of dimensions. We show the best results and the optimal dimensionality obtained by PCA, LDA, LPP, NPE, IsoProjection and baseline methods in Table 2. For each Gp/Pq, we average the results over 20 random splits. In IsoProjection, the regularization parameter α is set to be 0.01 empirically.

As can be seen, our algorithm performed the best in almost all the cases. This is because IsoProjection uses the regression as a building block and incorporates the regularization technique. When there exist a large number of features, IsoProjection with regularization can produce more stable and meaningful solutions (Hastie, Tibshirani, & Friedman 2001).

Conclusion

In this paper, we propose a new linear dimensionality reduction algorithm called Isometric Projection. Isometric Projection is based on the same variational principle that gives rise to the Isomap (Tenenbaum, de Silva, & Langford 2000). As a result it is capable of discovering the nonlinear degree of freedom that underlie complex natural observations. Our approach has a major advantage over recent nonparametric techniques for global nonlinear dimensionality reduction such as (Roweis & Saul 2000), (Tenenbaum, de Silva, & Langford 2000), (Belkin & Niyogi 2001) that the functional mapping between the high and low dimensional spaces are valid both on and off the training data. Comparing to LPP and NPE, which are the linear version of Laplacian Eigenmap and LLE, our approach has the computational advantage. Thus our algorithm can be applied on large scale high dimensional data. Performance improvement of this method over PCA, LDA, LPP and NPE is demonstrated through several experiments.

Acknowledgments

We thank the reviewers for helpful comments. The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678/06-42771 and NSF BDI-05-15813.

References

Belhumeur, P. N.; Hepanha, J. P.; and Kriegman, D. J. 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on PAMI* 19(7).

Belkin, M., and Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS 14*.

Cai, D.; He, X.; and Han, J. 2006. Isometric projection. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2006-2747.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2001. *Introduction to algorithms*. MIT Press, 2nd edition.

Deerwester, S. C.; Dumais, S. T.; Landauer, T. K.; Furnas, G. W.; and harshman, R. A. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41(6):391–407.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification*. Hoboken, NJ: Wiley-Interscience, 2nd edition.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

He, X., and Niyogi, P. 2003. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*.

He, X.; Cai, D.; Yan, S.; and Zhang, H.-J. 2005a. Neighborhood preserving embedding. In *Proc. ICCV'05*.

He, X.; Yan, S.; Hu, Y.; Niyogi, P.; and Zhang, H.-J. 2005b. Face recognition using laplacianfaces. *IEEE Transactions on PAMI* 27(3):328–340.

Lee, K.; Ho, J.; and Kriegman, D. 2005. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on PAMI* 27(5):684–698.

Lee, J. M. 2002. *Introduction to Smooth Manifolds*. Springer-Verlag New York.

Lovasz, L., and Plummer, M. 1986. *Matching Theory*. North Holland, Budapest: Akadémiai Kiadó.

Mardia, K. V.; Kent, J. T.; and Bibby, J. M. 1980. *Multivariate Analysis*. Academic Press.

Paige, C. C., and Saunders, M. A. 1982. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software* 8(1):43–71.

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

Stewart, G. W. 1998. *Matrix Algorithms Volume I: Basic Decompositions*. SIAM.

Stewart, G. W. 2001. *Matrix Algorithms Volume II: Eigensystems*. SIAM.

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