NONLINEAR DIMENSIONALITY REDUCTION

The goal of dimensionality reduction is to map a high-dimensional dataset to a low dimensional one in such

(A) a way that _____ and/or _____ geometric & topological properties are preserved.

The most common method for DR,

(B) PCA, is _____.

However, many high dimensional datasets have _____ structure that is not captured by this method.

Isomap

Isometric feature mapping is the application of MDS to dissimilarities derived from shortest path lengths based on a local proximity graph such as a k-nearest neighbor graph.

This path length is viewed as an approximation to _____ distance on the underlying data _____.

Example | Swiss roll data

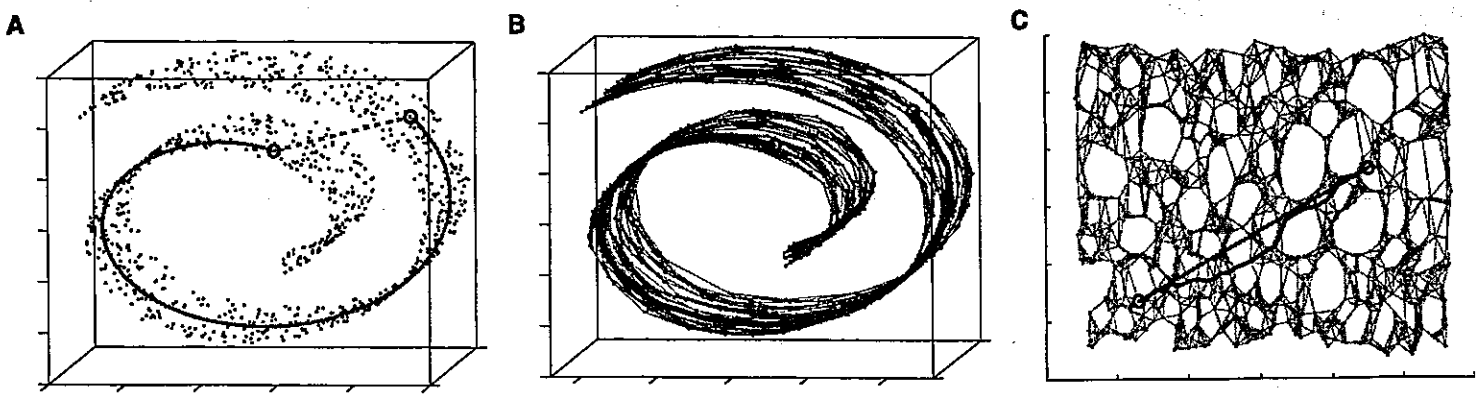
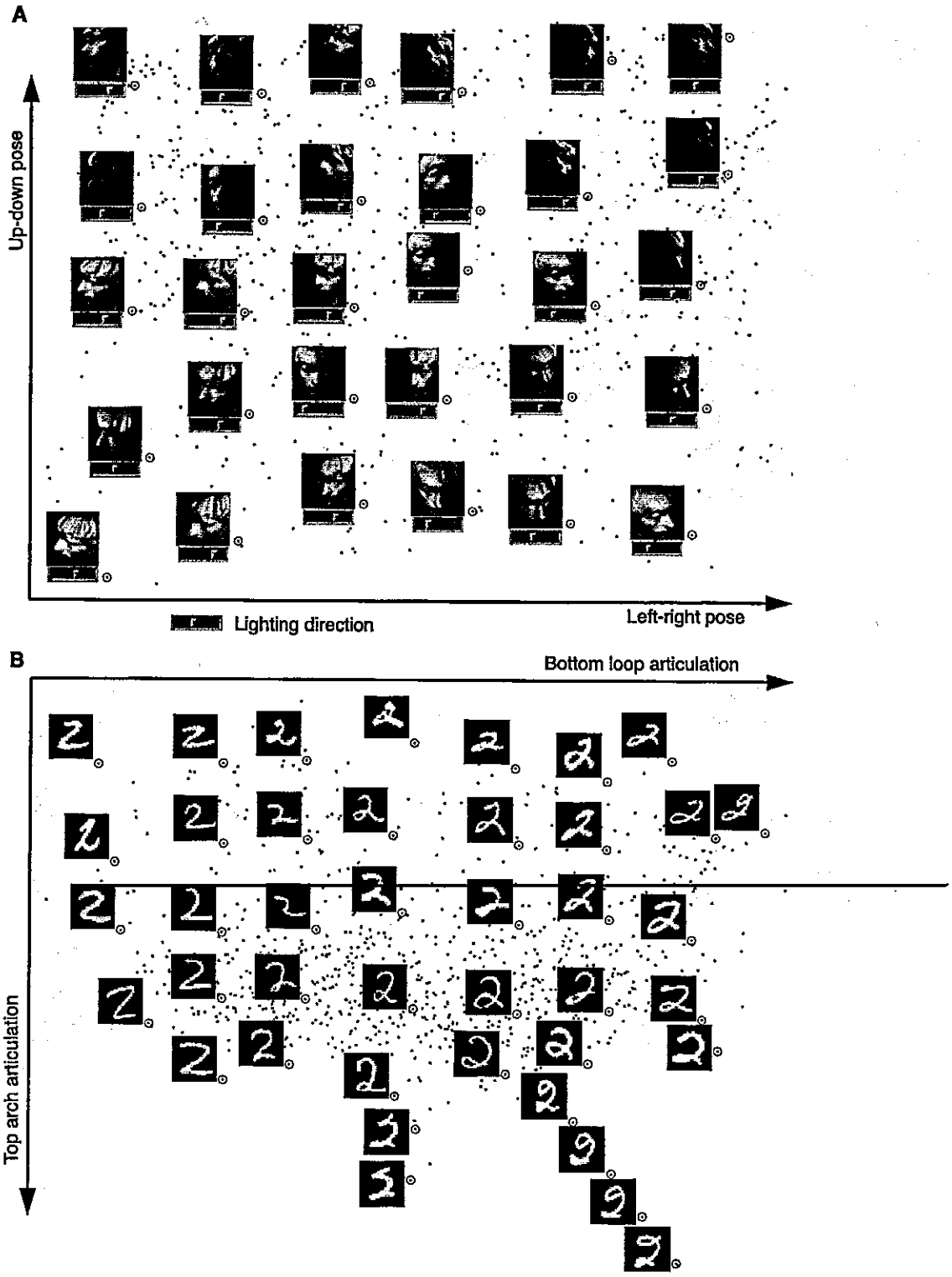


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N =$ 1000 data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).



Reference | Tennenbaum, de Silva, and Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science, 290, 2319-2323 (2000)

Laplacian eigenmaps

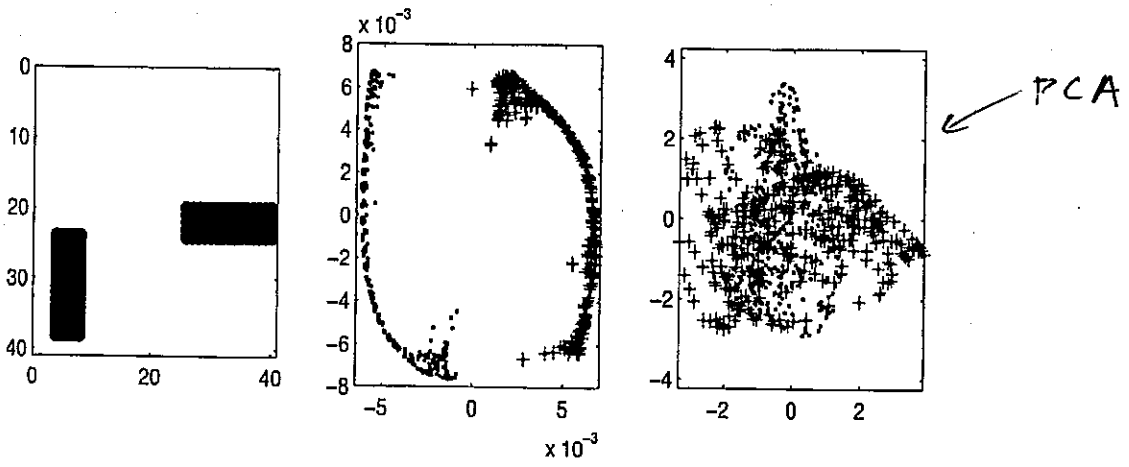
Given: $x_1, \ldots, x_n$

desired embedding dimension p

- compute similarity graph
- form graph Laplacian
- compute eigenvectors $u_2, \ldots, u_{p+1}$
- set

$$y_i = (u_{2i}, \ldots, u_{(p+1)i}) \in \mathbb{R}^p$$

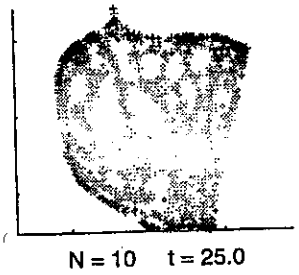
Example | Randomly positioned horizontal or vertical bar



Reference | M. Belkin + P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, Neural Computation 15, 1373-1396 (2003)

Unlike Isomap, Laplacian eigenmaps do not preserve global geodesic distances, just local neighborhood relationships.

Example | Swiss roll



Locally Linear Embedding

LLE capitalizes on the intuition that a data manifold that is globally nonlinear will appear linear locally.

Let high-dimensional data $x_1, \ldots, x_n \in \mathbb{R}^q$ be given. Let $y_1, \ldots, y_n \in \mathbb{R}^p$, $p < q$, be the desired embedding. LLE has 3 steps.

1. For each x_i, define a local neighborhood N_i, e.g., the k-nearest neighbors

(D) 2. Solve

$$\min_{\{w_{ij}\}} \quad \text{s.t.}$$

3. Now fix $\{w_{ij}\}$ and solve

$$\min_{\{y_i\}}$$

Computationally,

Step 2 $\Rightarrow$

Step 3 $\Rightarrow$

Even though LLE does not explicitly model global geodesic distances, it can still (hopefully) reconstruct the entire manifold by "patching together" local linear pieces.

Isomap vs. LLE

- ISOMAP: emphasizes global distance preservation ⇒ can distort local geometry
- LLE: emphasizes local neighborhood preservation ⇒ can lose global information, e.g. far away points mapped to nearby points.

Reference | Roweis & Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, 290, 2323–2326 (2000)

Kernel PCA

Kernel principal component analysis (KPCA) extends PCA in the same way that SVMs extend soft-margin hyperplane classifiers:

1. Map data $x_1, \ldots, x_n$ into a high-dimensional feature space $\mathcal{H}$

$$x = \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(d)} \end{bmatrix} \longmapsto \Phi(x) = \begin{bmatrix} \varphi^{(1)}(x) \\ \vdots \\ \varphi^{(D)}(x) \end{bmatrix}$$

(E) where $\varphi^{(j)}$ are _____ and $d \ll D$

2. Apply PCA to $\Phi(x_1), \ldots, \Phi(x_n)$

The principal components in $\mathcal{H}$ are linear in $\Phi(x_1), \ldots, \Phi(x_n)$, but nonlinear in $x_1, \ldots, x_n$. To make the procedure computationally tractable, we will rely

(F) on _____ _____ _____.

Recall | A function $k: \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ is an inner product kernel iff there exists an inner product space $\mathcal{H}$ and a mapping $\Phi: \mathbb{R}^d \longrightarrow \mathcal{H}$ such that

$$k(u, v) = \langle \Phi(u), \Phi(v) \rangle$$

for all $u, v \in \mathbb{R}^d$.

Derivation

We need to show that all operations can be represented in terms of inner products.

For simplicity, assume $\frac{1}{n}\sum_{i=1}^{n} \Phi(x_i) = 0$

(This will not be the case typically, but the general case can be reduced to this one)

The sample covariance matrix is

(G) $$C =$$

We need to find all $\lambda \geq 0$ and $v \in \mathcal{H}$, $v \neq 0$, such that

In fact, we only care about $\lambda > 0$.

Suppose v is an eigenvector of C with eigenvalue λ. Consider the following:

1) For each $i = 1, \dots, n$

$$\lambda \langle \Phi(x_i), v \rangle = \langle \Phi(x_i), Cv \rangle$$

2) If $\lambda > 0$, then

$$v = \frac{1}{\lambda} Cv$$

$$= \frac{1}{\lambda} \sum_{j=1}^{n} \Phi(x_j) \cdot \Phi(x_j)^T v$$

$$\in \text{span} \{ \Phi(x_1), \dots, \Phi(x_n) \}$$

which means there exist $a_1, \dots, a_n \in \mathbb{R}$ such that

(H) $$v =$$

Combining 1) and 2) we conclude: If $v \in \mathcal{H}$ is an eigenvector of C with eigenvalue $\lambda > 0$, then there exists $\alpha = (\alpha_1, \dots, \alpha_n)^T \in \mathbb{R}^n$ such that

$$v = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$$

and for each $k = 1, \dots, n$

$$\lambda \left\langle \Phi(x_k), \sum_{i=1}^{n} \alpha_i \Phi(x_i) \right\rangle = \frac{1}{n} \left\langle \Phi(x_k), \sum_{j=1}^{n} \Phi(x_j) \Phi(x_j)^T \sum \alpha_i \Phi(x_i) \right\rangle$$

$$\Updownarrow$$

(I)

$$\Updownarrow$$

where

$$K := \left[k(x_i, x_j) \right]_{i,j=1}^{n}$$

Claim Let $\lambda > 0$. Then

$$n\lambda K\alpha = K^2\alpha \iff n\lambda\alpha = K\alpha \quad \text{or} \quad K\alpha = 0$$

Proof ($\Leftarrow$) obvious ($\Rightarrow$) Assume $n\lambda K\alpha = K^2\alpha$.

We need to show α is an e-vec of K with e-val $n\lambda$, or 0.

Since K is symmetric & PSD, we can write $K = UDU^T$

where $U^TU = UU^T = I$, and D is diagonal with elements ≥ 0.

$$n\lambda K\alpha = K^2\alpha \iff n\lambda UDU^T\alpha = UD^2U^T\alpha$$

Set $w = U^T\alpha$. (change of coordinates), and left multiply by U^T, to yield $n\lambda Dw = D^2w$

$$n\lambda \begin{bmatrix} d_1 & & & & & \\ & d_2 & & & & \\ & & \ddots & & & \\ & & & d_m & & \\ & & & & 0 & \\ & & & & & \ddots \\ & & & & & & 0 \end{bmatrix} = \begin{bmatrix} d_1^2 & & & & & \\ & d_2^2 & & & & \\ & & \ddots & & & \\ & & & d_m^2 & & \\ & & & & 0 & \\ & & & & & \ddots \\ & & & & & & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

Assuming (for simplicity) $d_1 > d_2 > \cdots > d_m > 0$, we have either

(1) $w_1 = w_2 = \cdots = w_m = 0 \implies \alpha = Uw \in 0$-eigenspace of K

(2) $w = e_j$ (j^{th} standard basis vector) for some j, $1 \leq j \leq m$, in which case $n\lambda = d_j$ and $\alpha = Uw \in n\lambda$-eigenspace of K

If some eigenvalues have multiplicity > 1, the above conclusions can be easily seen to hold (if you understand eigenspaces) ∎

If $K\alpha = 0$, and $v = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$, then

$$K\alpha = \begin{bmatrix} \langle \Phi(x_1), v \rangle \\ \vdots \\ \langle \Phi(x_n), v \rangle \end{bmatrix} = 0$$

$\Rightarrow v = 0$.

Thus, we only need to find solutions of

$$n\lambda\alpha = K\alpha$$

with $\lambda > 0$. That is, if $n\lambda_1 \geq \cdots \geq n\lambda_m$ are the nonzero eigenvalues of K, with corresponding eigenvectors $\underline{\alpha}_1, \ldots, \underline{\alpha}_m$, then $\lambda_1 \geq \cdots \geq \lambda_m$ are the nonzero eigenvalues of $C = \frac{1}{n} \sum \Phi(x_i)\Phi(x_i)^T$, with corresponding

$$v_j = \sum_i \alpha_{ji} \Phi(x_i).$$

To ensure $\|v_j\| = 1$ we normalize $\underline{\alpha}_j$ so that

If $x \in \mathbb{R}^d$ is a test point, then the j^{th} component of x is

(R) $$\langle \underline{\Phi}(x), v_j \rangle =$$

Recall we assumed $\sum_{i=1}^{n} \underline{\Phi}(x_i) = 0$. To eliminate this assumption, just apply the above steps to

This leads to needing to diagonalize $\tilde{K}$ where

$$\tilde{K}_{ij} = \left(K_{ij} - \underline{1}_n K - K \underline{1}_n + \underline{1}_n K \underline{1}_n \right)_{ij}$$

and where $\underline{1}_n$ is $n \times n$ with $(\underline{1}_n)_{ij} = \frac{1}{n} \quad \forall_{ij}$.

KPCA

Input: $x_1, \ldots, x_n \in \mathbb{R}^d$, kernel k, dimension $p \leq m$

- Form $\tilde{K}$ and compute $\tilde{K} = U \cdot D \cdot U^T$ where

$$U = [u_1 \cdots u_n], \quad D = \text{diag}(d_1, \ldots, d_m, 0, \ldots, 0)$$

- Set $\alpha_j = \frac{1}{\sqrt{d_j}} u_j, \quad 1 \leq j \leq p$

Output: mapping $x \longmapsto y = (y^{(1)}, \ldots, y^{(p)})^T \in \mathbb{R}^p$ where

$$y^{(j)} = \sum_{i=1}^{n} \alpha_{ji} k(x, x_i)$$

Remarks: ① Unlike PCA, KPCA can return potentially more than d principal components.

② Unlike MDS, Isomap, Laplacian eigenmaps, and LLE, KPCA defines a mapping that applies to arbitrary test points, not just the training data.

③ MDS and LLE can be related to KPCA

④ See Smola and Schölkopf, Learning with Kernels, for more information.

Key A. global, local B. linear, nonlinear

C. geodesic, manifold

D. $\min_{\{w_{ij}\}} \sum_{i=1}^{n} \left\| x_i - \sum_{j \in N_i} w_{ij} x_j \right\|^2$ quadratic program

s.t. $\sum_{j \in N_i} w_{ij} = 1$, $w_{ij} \geq 0$

$\min_{\{y_i\}} \sum_{i=1}^{n} \left\| y_i - \sum_{j \in N_i} w_{ij} y_j \right\|^2$ eigenvalue problem

E. nonlinear F. inner product kernels

G. $C = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^T$, $Cv = \lambda v$

H. $v = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$

I. $\lambda (K\alpha)_\ell = \frac{1}{n} (K^2 \alpha)_\ell \iff n\lambda K\alpha = K^2 \alpha$

$K = [k(x_i, x_j)]_{ij}$

J. $1 = \|v_j\| = \sum_i \sum_\ell \alpha_{ji} \alpha_{j\ell} k(x_i, x_\ell) = \alpha_j^T K \alpha_j = n\lambda_j \|\alpha_j\|^2$

K. $\langle \Phi(x), v_j \rangle = \langle \Phi(x), \sum_i \alpha_{ji} \Phi(x_i) \rangle = \sum_i \alpha_{ji} k(x, x_i)$

L. $\hat{\Phi}(x_i) := \Phi(x_i) - \frac{1}{n} \sum_\ell \Phi(x_\ell)$