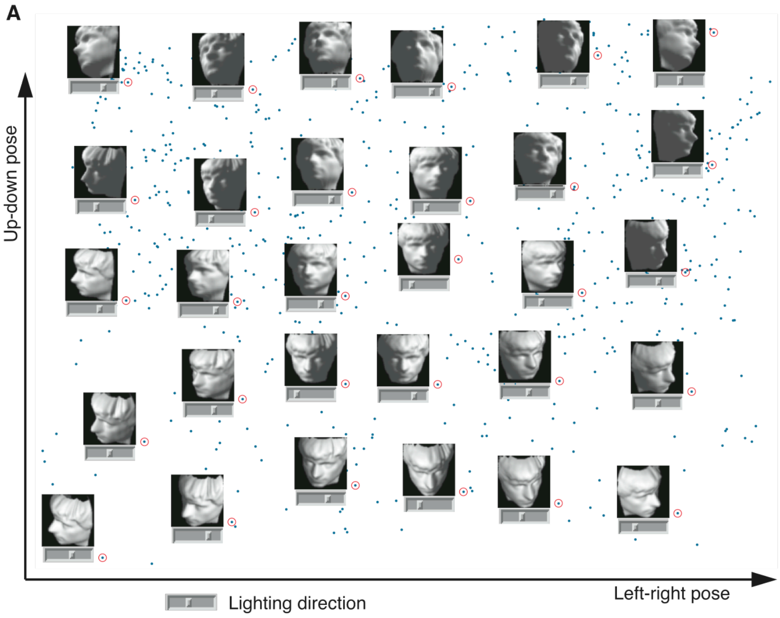
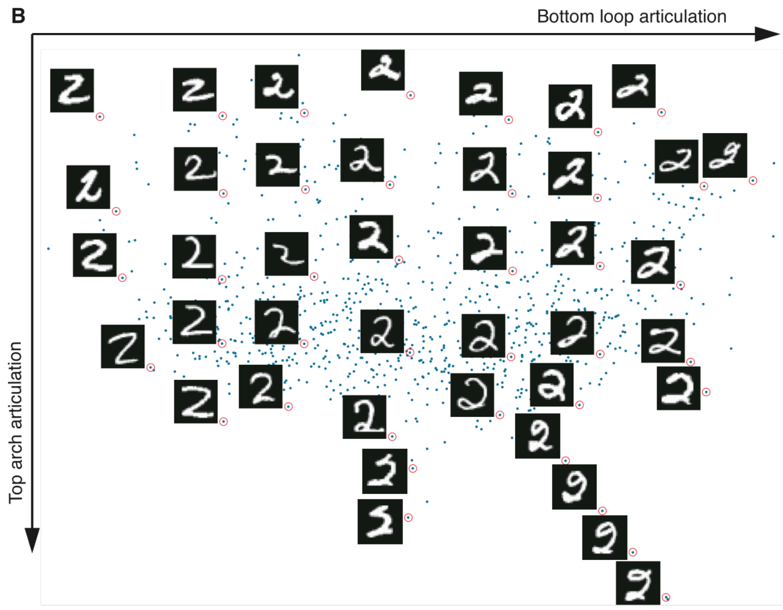
# Manifold learning:

# Isomap
# and
# Locally Linear Embedding

## Motivation



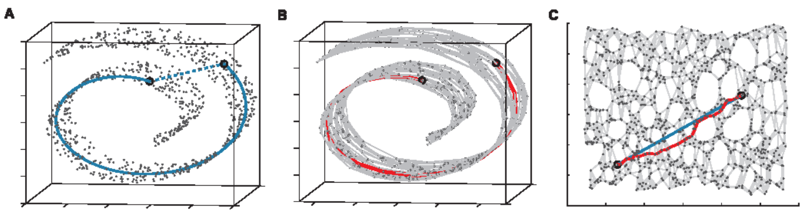
## Another example




## Isomap: Goals

- Discover the intrinsic degrees of freedom in the data
- Learn the underlying global geometry of the data using local metric information
- It's a **nonlinear** dimensionality reduction technique (unlike PCA)
- Like PCA: noniterative, polynomial time, global optimality
- Applications in vision, speech, motor control, biological sciences (and more)

## Isomap: overall approach




- Construct neighbourhood graph G
- For each pair of points in G, compute shortest path distances ---- geodesic distances.
- Use Classical MDS with geodesic distances.
  - Euclidean distance→ Geodesic distance

## Isomap: the algorithm

| Step | | |
|---|---|---|
| 1 | Construct neighborhood graph | Define the graph $G$ over all data points by connecting points $i$ and $j$ if [as measured by $d_X(i,j)$] they are closer than $\epsilon$ ($\epsilon$-Isomap), or if $i$ is one of the $K$ nearest neighbors of $j$ ($K$-Isomap). Set edge lengths equal to $d_X(i,j)$. |
| 2 | Compute shortest paths | Initialize $d_G(i,j) = d_X(i,j)$ if $i,j$ are linked by an edge; $d_G(i,j) = \infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. The matrix of final values $D_G = \{d_G(i,j)\}$ will contain the shortest path distances between all pairs of points in $G$ (*16*, *19*). |
| 3 | Construct $d$-dimensional embedding | Let $\lambda_p$ be the $p$-th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$ (*17*), and $v_p^i$ be the $i$-th component of the $p$-th eigenvector. Then set the $p$-th component of the $d$-dimensional coordinate vector $\mathbf{y}_i$ equal to $\sqrt{\lambda_p} v_p^i$. |

# The algorithm: Step 3

- The cost function minimized in Step 3 is:

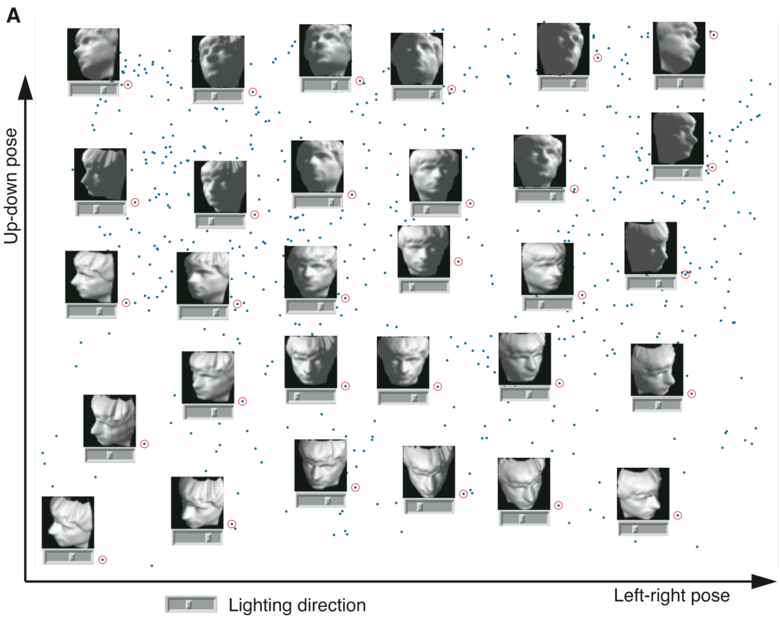
$$E = \left\| \tau(D_G) - \tau(D_Y) \right\|_2$$

- where $D_Y$ is the matrix of Euclidean distances

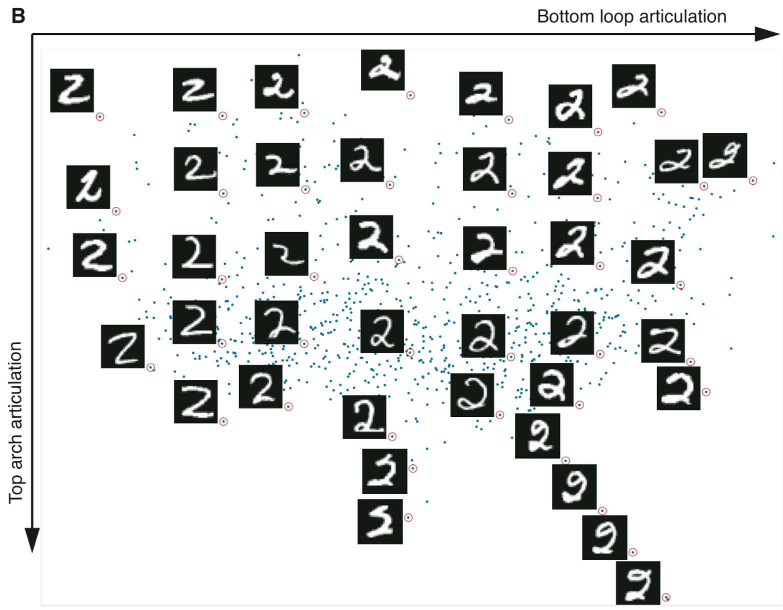
$$d_Y(i,j) = \left\| y_i - y_j \right\|$$

and $\tau$ is an operator that converts distances into dot products

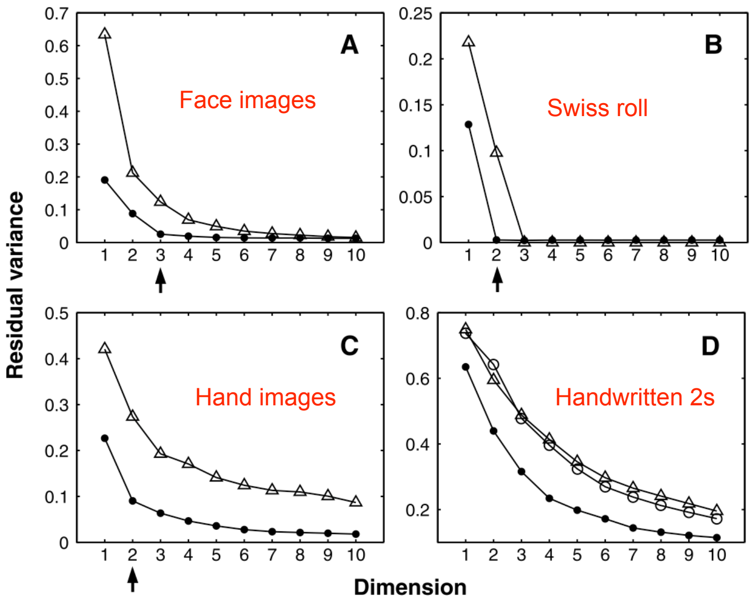
# Isomap: Results



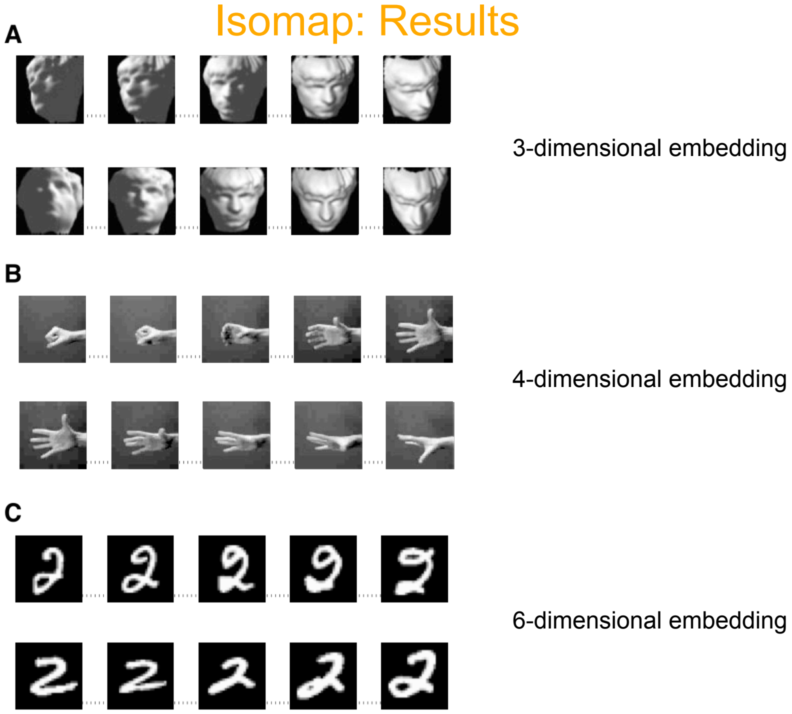
# Isomap: Results




# Isomap vs. PCA vs. MDS



## Isomap: Results




## Review: Isomap

- Isomap is still a global approach
  - All pairwise distances considered
- The resulting distance matrix is dense
  - Isomap does not scale to large datasets
  - Landmark Isomap proposed to overcome this problem
- LLE (Local Linear Embedding)
  - local approach
  - The resulting matrix is sparse
    - Apply efficient sparse matrix solvers

## Local Linear Embedding (LLE)

- Local Linear Embedding (LLE)
  - Intuition
  - Least squares problem
  - Eigenvalue problem

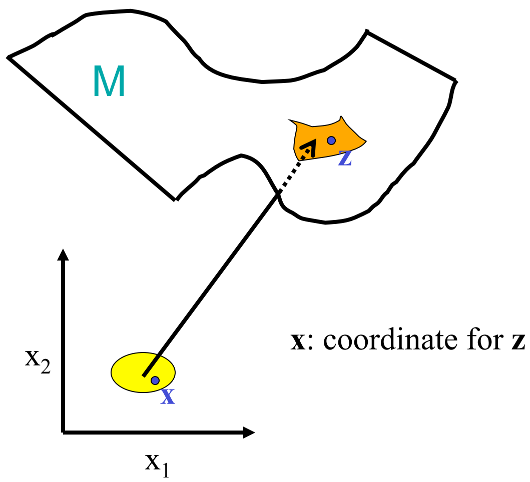
## Characterictics of a Manifold

$R^n$

Locally it is a linear patch

Key: how to combine all local patches together?

$R^2$



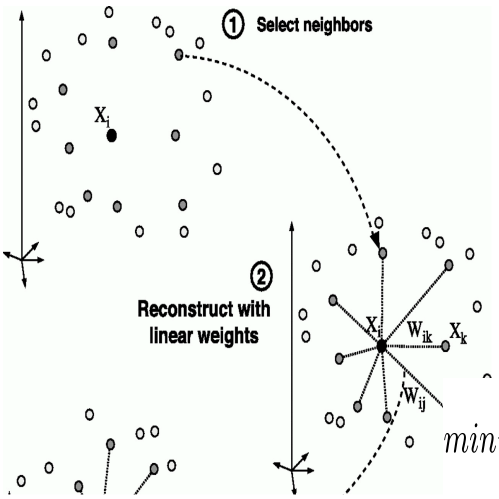
## LLE: Intuition

- Assumption: a manifold is approximately "linear" when viewed locally, that is, within a small neighborhood
  - Approximation error, e(W), can be made small:

$$min_W \parallel X_i - \sum_{j=1}^{K} W_{ij} X_j \parallel^2 \qquad (1)$$

- Locality is enforced by the constraint $W_{ij}=0$ if $z_j$ is not a neighbor of $z_i$
- A good projection should preserve this local geometric property as much as possible

## LLE: Intuition




$$min_W \parallel X_i - \sum_{j=1}^{K} W_{ij} X_j \parallel^2 \qquad (1)$$

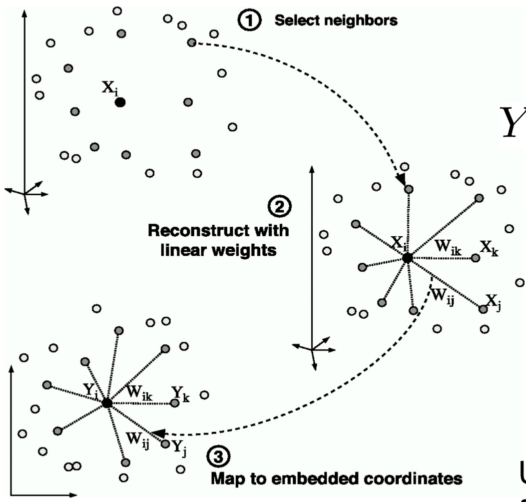
We expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold.

Each point can be written as a linear combination of its neighbors.
The weights chosen to minimize the reconstruction Error.

## LLE: Intuition

- The weights that minimize the reconstruction errors are invariant to rotation, rescaling and translation of the data points.
  - Invariance to translation is enforced by adding the constraint that the weights sum to one.
  - The weights characterize the intrinsic geometric properties of each neighborhood.
- **The same weights that reconstruct the data points in D dimensions should reconstruct it in the manifold in d dimensions.**
  - **Local geometry is preserved**

## LLE: Intuition




Low-dimensional embedding

$$Y_{d\times N} = [Y_1|Y_2|...|Y_N]$$

$$min_Y \sum_{i=1}^{N} \parallel Y_i - YW_i \parallel^2$$

the i-th row of W

Use the same weights from the original space

## Local Linear Embedding (LLE)

- Assumption: manifold is approximately "linear" when viewed locally, that is, in a small neighborhood
- Approximation error, $\varepsilon(\mathbf{W})$, can be made small

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2$$

- Meaning of **W**: a linear representation of every data point by its neighbors
  - This is an intrinsic geometrical property of the manifold
- A good projection should preserve this geometric property as much as possible

## Constrained Least Square problem

Compute the optimal weight for each point individually:

$$\varepsilon = \left| \vec{x} - \Sigma_j w_j \vec{\eta}_j \right|^2 = \left| \Sigma_j w_j \left( \vec{x} - \vec{\eta}_j \right) \right|^2 = \Sigma_{jk} w_j w_k C_{jk},$$

Neighbors of x

$$C_{jk} = (\vec{x} - \vec{\eta}_j) \cdot (\vec{x} - \vec{\eta}_k).$$

This error can be minimized in closed form, using a Lagrange multiplier to enforce the constraint that $\sum_j w_j = 1$. In terms of the inverse local covariance matrix, the optimal weights are given by:

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{lm} C_{lm}^{-1}}. \qquad (5)$$

Zero for all non-neighbors of x

## Finding a Map to a Lower Dimensional Space

Derivation for mapping Y:

$$\Phi(Y) = \sum_i \left\| Y_i - \sum_j W_{ij} Y_j \right\|^2 = \sum_i \left\| Y_i - [Y_1, Y_2, \cdots, Y_n] W_i^T \right\|^2$$

$$= \left\| [Y_1, Y_2, \cdots, Y_n] - [Y_1, Y_2, \cdots, Y_n][W_1^T, W_2^T, \cdots, W_n^T] \right\|_F^2$$

$$= \left\| Y - YW^T \right\|_F^2 = \left\| Y(I - W^T) \right\|_F^2 = \operatorname{trace}(Y(I-W)^T(I-W)Y^T)$$

$$= \operatorname{trace}(YMY^T)$$

where $Y = [Y_1, Y_2, \cdots, Y_n]$.

## Eigenvalue problem

Add the following constraint: $\sum_i \vec{Y}_i = \vec{0}.$

Also, to avoid degenerate solutions, we constrain the embedding vectors to have unit covariance, with outer products that satisfy

$$\frac{1}{N} \sum_i \vec{Y}_i \vec{Y}_i^\top = I, \qquad (10)$$

The optimal d-dimensional embedding is given by the bottom 2nd to (d+1)-th eigenvectors of the following matrix: (Note that 0 is its smallest eigenvalue.)

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$$

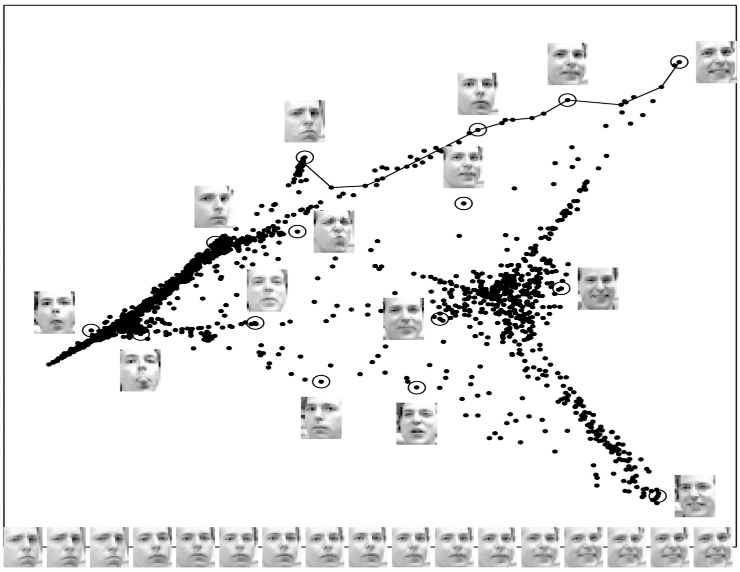
## The LLE algorithm

LLE ALGORITHM

1. Compute the neighbors of each data point, $\vec{X}_i$.
2. Compute the weights $W_{ij}$ that best reconstruct each data point $\vec{X}_i$ from its neighbors, minimizing the cost in eq. (1) by constrained linear fits.
3. Compute the vectors $\vec{Y}_i$ best reconstructed by the weights $W_{ij}$, minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.

Figure 2: Summary of the LLE algorithm, mapping high dimensional data points, $\vec{X}_i$, to low dimensional embedding vectors, $\vec{Y}_i$.

## Examples



Images of faces mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points. The bottom images correspond to points along the top-right path (linked by solid line) illustrating one particular mode of variability in pose and expression.

## Some Limitations of LLE

- Require dense data points on the manifold for good estimation
- A good neighborhood seems essential to their success
  - How to choose k?

## Experiment on LLE

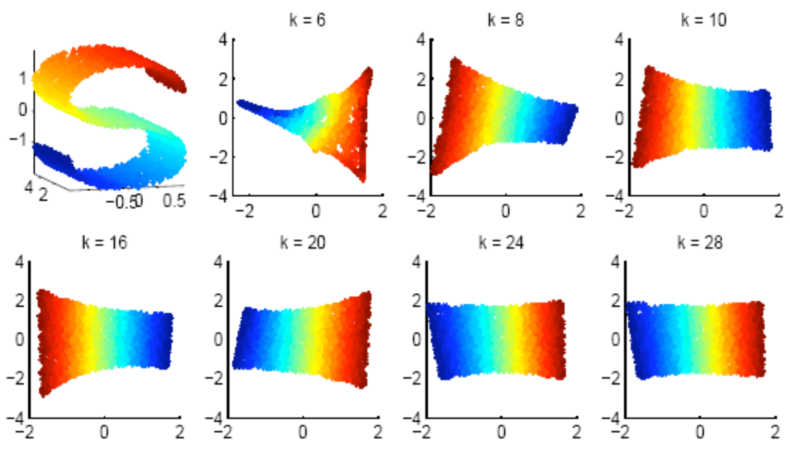



FIG. 5. *S-curve (top left) and computed 2D coordinates by LLE with various neighborhood size k.*

## Applications

- **Image Processing Using Locally Linear Embedding**
- **Feature Dimension Reduction For Microarray Data Analysis Using Locally Linear Embedding**
- **Locally linear embedding algorithm. Extensions and applications**
  - **http://herkules.oulu.fi/isbn9514280415/isbn9514280415.pdf**