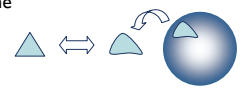
Subspace and Manifold Learning

Slides created by Prof. Tyng-Luh Liu

Manifold

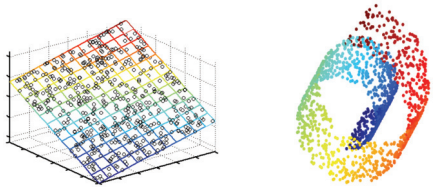
- In the neighborhood of any point on a manifold, the space behaves just like it would in the neighborhood of any point in some n-dimensional Euclidean space
- A sphere in 3-dimensional Euclidean space
  - A very small patch from the sphere looks just like a piece of 2-dimensional plane
  - 2-manifold



Framework

- Dimensionality reduction and manifold learning
- Data representation
  - Inputs are real-valued vectors in a high-dimensional space
- Linear structure
  - Does the data live in a low-dimensional subspace?
- Nonlinear structure
  - Does the data live on a low-dimensional submanifold?

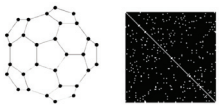
Linear vs. Nonlinear



What computational price must we pay for nonlinear dimensionality reduction?

Spectral Methods

- Matrix analysis
  - The low-dimensional structure is revealed by eigenvalues and eigenvectors
- Links to spectral graph theory
  - Matrices are derived from sparse weighted graphs



- Usefulness
  - Tractable methods can reveal nonlinear structures

Notation

- Inputs (high dimensional)

  $\mathbf{x}_i \in \mathbb{R}^D$ with $i = 1, 2, \ldots, m$
- Projections (high dimensional)

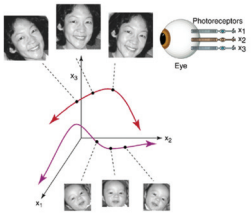
  $\mathbf{z}_i \in \mathbb{R}^D$ with $i = 1, 2, \ldots, m$
- Outputs (low dimensional)

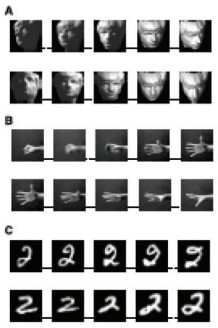
  $\mathbf{y}_i \in \mathbb{R}^d$ where $d \ll D$
- Goals, e.g.,
  - Nearby points remain nearby
  - Distant points remain distant
  - (Estimate d)

Manifold Learning

- Given high-dimensional data sampled from a low-dimensional submanifold, how to compute a faithful embedding?

Image Manifolds





Seung & Lee, 2000
Tenebaum et al., 2000

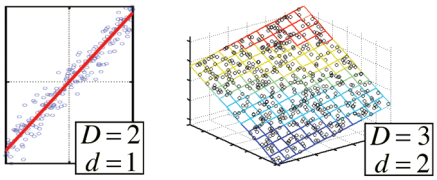
Outline

- Linear methods
  - PCA
  - MDS
- Graph-based methods
  - Isomap
  - LLE

Linear Methods #1

Principal Component Analysis (PCA)

Principal Component Analysis



Does the data mostly lie in a subspace?
If so, what is its dimensionality?

Maximum Variance Subspace

- Assume inputs are centered on the origin

$$\sum_i \mathbf{x}_i = 0$$

- Project into subspace

$$\mathbf{z}_i = P\mathbf{x}_i \text{ with } P^2 = P$$

- Maximize projected variance

$$\text{var}(\mathbf{z}) = \frac{1}{m}\sum_i \|P\mathbf{x}_i\|^2$$

Matrix Diagonalization

- Covariance matrix

$$\text{var}(\mathbf{z}) = \text{tr}(PCP^T) \text{ with } C = \frac{1}{m}\sum_i \mathbf{x}_i \mathbf{x}_i^T \in \mathbb{R}^{D\times D}$$

- Spectral decomposition

$$C = \sum_{\alpha=1}^{D} \lambda_\alpha \mathbf{e}_\alpha \mathbf{e}_\alpha^T \text{ with } \lambda_1 \geq \cdots \geq \lambda_D \geq 0$$

- Maximum variance projection

$$P = \sum_{\alpha=1}^{d} \mathbf{e}_\alpha \mathbf{e}_\alpha^T$$

Projects onto subspace spanned by top d eigenvectors

Linear Algebra

$$\begin{aligned}\text{var}(\mathbf{z}) &= \frac{1}{m}\sum_i \|P\mathbf{x}_i\|^2 = \frac{1}{m}\sum_i (P\mathbf{x}_i)^T(P\mathbf{x}_i) \\ &= \frac{1}{m}\sum_i \text{tr}\left((P\mathbf{x}_i)(P\mathbf{x}_i)^T\right) = \frac{1}{m}\sum_i \text{tr}\left(P\mathbf{x}_i\mathbf{x}_i^T P^T\right) \\ &= \text{tr}\left(P\frac{1}{m}\sum_i \mathbf{x}_i\mathbf{x}_i^T P^T\right) = \text{tr}(PCP^T) \qquad (\text{with } C = \tfrac{1}{m}\textstyle\sum_i \mathbf{x}_i\mathbf{x}_i^T)\end{aligned}$$

$$C = \sum_{\alpha=1}^{D} \lambda_\alpha \mathbf{e}_\alpha \mathbf{e}_\alpha^T \text{ with } \lambda_1 \geq \cdots \geq \lambda_D \geq 0 \quad \Rightarrow \quad \text{tr}(C) = \sum_{\alpha=1}^{D} \lambda_\alpha$$

$$C\mathbf{e}_\alpha = \lambda_\alpha \mathbf{e}_\alpha$$

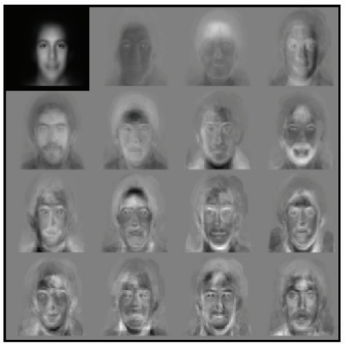
$$P = \sum_{\alpha=1}^{d} \mathbf{e}_\alpha \mathbf{e}_\alpha^T$$

$$\text{var}(\mathbf{z}) = \text{tr}(PCP^T) = \text{tr}\left(\sum_{\alpha=1}^{d} \lambda_\alpha \mathbf{e}_\alpha \mathbf{e}_\alpha^T\right) = \sum_{\alpha=1}^{d} \lambda_\alpha$$

Interpreting PCA

- Eigenvectors
  - Principal axes of maximum variance subspace
- Eigenvalues
  - Projected variance of inputs along principle axes
- Estimated dimensionality
  - Number of significant (nonnegative) eigenvalues

Example: Faces



Eigenfaces from 7562 images

top left image is linear combination of rest

Sirovich & Kirby (1987)
Turk & Pentland (1991)

Another Interpretation

- Assume inputs are centered on the origin

$$\sum_i \mathbf{x}_i = 0$$

- Project into subspace

$$\mathbf{z}_i = P\mathbf{x}_i \text{ with } P^2 = P$$

- Minimize reconstruction error

$$\text{err}(\mathbf{z}) = \frac{1}{m}\sum_i \|\mathbf{x}_i - P\mathbf{x}_i\|^2$$

Equivalence

- Minimum reconstruction error

$$\text{err}(\mathbf{z}) = \frac{1}{m}\sum_i \|\mathbf{x}_i - P\mathbf{x}_i\|^2$$

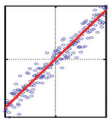
- Maximum variance subspace

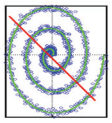
$$\text{var}(\mathbf{z}) = \frac{1}{m}\sum_i \|P\mathbf{x}_i\|^2$$

Both models for linear dimensionality reduction yield the same solution

Properties of PCA

- Strengths
  - Eigenvector method
  - No tuning parameters
  - Non-iterative
  - No local optima





- Weaknesses
  - Limited to second order statistics
  - Limited to linear projections

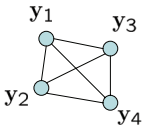
Linear Methods #2

Multidimensional Scaling (MDS)

Multidimensional Scaling

input

$$\begin{bmatrix} 0 & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{12} & 0 & \Delta_{23} & \Delta_{24} \\ \Delta_{13} & \Delta_{23} & 0 & \Delta_{34} \\ \Delta_{14} & \Delta_{24} & \Delta_{34} & 0 \end{bmatrix}$$



e.g. $\Delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$

Given $m(m-1)/2$ pairwise distances Δ_{ij}
find vectors $\mathbf{y}_i$ such that $\|\mathbf{y}_i - \mathbf{y}_j\| \approx \Delta_{ij}$

Metric Multidimensional Scaling

- Lemma
  If Δ_{ij} denote the Euclidean distances of zero mean vectors, then the inner products are

$$G_{ij} = \frac{1}{2}\left[\frac{1}{m}\sum_k \left(\Delta_{ik}^2 + \Delta_{kj}^2\right) - \Delta_{ij}^2 - \frac{1}{m^2}\sum_{k,l}\Delta_{kl}^2\right]$$

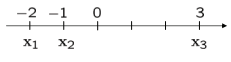
Gram matrix:
$G_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$

- Optimization
  Preserve dot products (proxy for distances)
  Choose vectors $\mathbf{y}_i$ to minimize

$$\text{err}(\mathbf{y}) = \sum_{i,j}\left(G_{ij} - \mathbf{y}_i \cdot \mathbf{y}_j\right)^2$$

Distance Matrix
→ Gram Matrix $G_{ij} = \frac{1}{2}\left[\frac{1}{m}\sum_k \left(\Delta_{ik}^2 + \Delta_{kj}^2\right) - \Delta_{ij}^2 - \frac{1}{m^2}\sum_{k,l}\Delta_{kl}^2\right]$

Example: three 1-d points



$$[\Delta_{ij}] = \begin{bmatrix} 0 & 1 & 5 \\ 1 & 0 & 4 \\ 5 & 4 & 0 \end{bmatrix} \Rightarrow [\Delta_{ij}^2] = \begin{bmatrix} 0 & 1 & 25 \\ 1 & 0 & 16 \\ 25 & 16 & 0 \end{bmatrix}$$

$$\begin{aligned} G_{12} &= \frac{1}{2}\left[\frac{1}{3}\sum_k \left(\Delta_{1k}^2 + \Delta_{k2}^2\right) - \Delta_{12}^2 - \frac{1}{9}\sum_{k,l}\Delta_{kl}^2\right] \\ &= \frac{1}{2}\left[\frac{1}{3}(0+1+25+1+0+16) - 1\right. \\ &\quad \left. - \frac{1}{9}(0+1+25+1+0+16+25+16+0)\right] \\ &= 2 \end{aligned}$$

Matrix Diagonalization

- Gram matrix "matching"

$$\text{err}(\mathbf{y}) = \sum_{i,j}\left(G_{ij} - \mathbf{y}_i \cdot \mathbf{y}_j\right)^2$$

- Spectral decomposition

$$G = \sum_{\alpha=1}^{m} \lambda_\alpha \mathbf{v}_\alpha \mathbf{v}_\alpha^T \text{ with } \lambda_1 \geq \cdots \geq \lambda_m \geq 0$$

- Optimal (low-rank) approximation

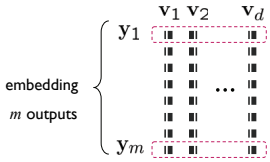
$$y_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i} \text{ for } \alpha = 1, 2, \ldots, d \text{ with } d \ll m$$

(scaled truncated eigenvectors)

Embedding

$G = \sum_{\alpha=1}^{m} \lambda_\alpha \mathbf{v}_\alpha \mathbf{v}_\alpha^T$ with $\lambda_1 \geq \cdots \geq \lambda_m \geq 0$

top d eigenvectors $\mathbf{v}_\alpha$ for $\alpha = 1, 2, \ldots, d$ with $d \ll m$



Interpreting MDS

$y_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i}$ for $\alpha = 1, 2, \ldots, d$ with $d \ll m$

- Eigenvectors
  - Ordered, scaled, and truncated to yield low-dimensional embedding
- Eigenvalues
  - Measure how each dimension contributes to dot products
- Estimated dimensionality
  - Number of significant (nonnegative) eigenvalues

Relation to PCA

- Dual matrices

$C_{\alpha\beta} = \frac{1}{n}\sum_i x_{i\alpha} x_{i\beta}$	covariance matrix $(D \times D)$	$\leftarrow XX^T$
$G_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$	Gram matrix $(m \times m)$	$\leftarrow X^T X$

$X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_m] \in \mathbb{R}^{D \times m}$

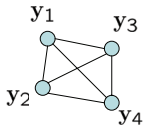
- Same eigenvalues
  - Matrices share nonzero eigenvalues up to constant factor
    - PCA: projection matrix
    - MDS: embedding
- Same results, different computation

  PCA scales as $O((m+d)D^2)$

  MDS scales as $O((D+d)m^2)$

Non-Metric MDS

$$\begin{bmatrix} 0 & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{12} & 0 & \Delta_{23} & \Delta_{24} \\ \Delta_{13} & \Delta_{23} & 0 & \Delta_{34} \\ \Delta_{14} & \Delta_{24} & \Delta_{34} & 0 \end{bmatrix}$$



Transform pairwise distances $\Delta_{ij} \rightarrow g(\Delta_{ij})$

Find vectors $\mathbf{y}_i$ such that $\|\mathbf{y}_i - \mathbf{y}_j\| \approx g(\Delta_{ij})$

Non-Metric MDS

- Distance transformation
  - Nonlinear, but monotonic
  - Preserves rank order of distances
- Optimization
  - Preserve transformed distances

  Choose vectors $\mathbf{y}_i$ to minimize

$$\text{err}(\mathbf{y}) = \sum_{i,j} \left(g(\Delta_{ij}) - \|\mathbf{y}_i - \mathbf{y}_j\| \right)^2$$

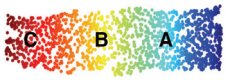
Properties of Non-Metric MDS

- Strengths
  - Relaxes distance constraints
  - Yields nonlinear embeddings
- Weaknesses
  - Highly nonlinear, iterative optimization with local minima
  - Unclear how to choose distance transformation

Non-Metric MDS for Manifolds?

Rank ordering of Euclidean distances is NOT preserved in "manifold learning"

$d(A, C) < d(A, B)$



$d(A, C) > d(A, B)$

Graph-Based Methods #1

Isometric Mapping of Data Manifolds (ISOMAP)

(Tenenbaum, de Silva, and Langford, 2000)

Dimensionality Reduction

- Inputs (high dimensional)

  $\mathbf{x}_i \in \mathbb{R}^D$ with $i = 1, 2, \ldots, m$

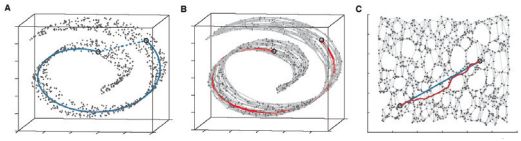
- Outputs (low dimensional)

  $\mathbf{y}_i \in \mathbb{R}^d$ where $d \ll D$

- Goals
  - Nearby points remain nearby
  - Distant points remain distant
  - (Estimate d)

Isomap

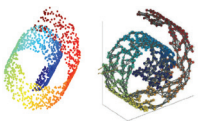
- Key idea
  - Preserve **geodesic distances** as measured along submanifold
- Algorithm in a nutshell
  - Use geodesic instead of (transformed) Euclidean distances in MDS



Step 1. Build Adjacency Graph

- Adjacency graph
  - Vertices represent inputs
  - Undirected edges connect neighbors
- Neighborhood selection
  - Many options:

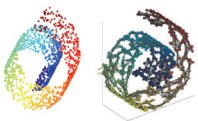
    k-nearest neighbors, ϵ-ball, prior knowledge



graph is discretized approximation of submanifold

Building the Graph

- Computation
  - kNN scales naively as $O(m^2 D)$
  - Faster methods exploit data structures
- Assumptions
  - Graph is connected
  - Neighborhoods on graph reflect neighborhoods on manifold



no "shortcuts" connect different arms of Swiss roll

Step 2. Estimate Geodesics

- Dynamic programming
  - Weight edges by local Euclidean distances
  - Compute **shortest paths** through graph
- Geodesic distances
  - Estimate by lengths Δ_{ij} of shortest paths: denser sampling = better estimates
- Computation
  - Dijkstra's algorithm for all-pair shortest paths scales as $O(m^2k + m^2 \log m)$

shortest path: $O(|E| + |V| \log |V|)$

Step 3. Metric MDS

- Embedding
  - Top d eigenvectors of Gram matrix yield embedding
- Dimensionality
  - Number of significant eigenvalues yield estimate of dimensionality
- Computation
  - Top d eigenvectors can be computed in $O(m^2 d)$

Summary of Isomap

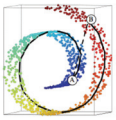
- Algorithm
  (1) k nearest neighbors
  (2) Shortest paths through graph (estimate geodesic distances)
  (3) MDS on geodesic distances

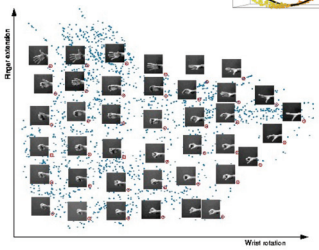
$$\begin{bmatrix} 0 & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{12} & 0 & \Delta_{23} & \Delta_{24} \\ \Delta_{13} & \Delta_{23} & 0 & \Delta_{34} \\ \Delta_{14} & \Delta_{24} & \Delta_{34} & 0 \end{bmatrix}$$

- Impact
  - Much simpler than earlier algorithms for manifold learning
  - Does it work?

Examples

Swiss roll
$m = 1024$
$k = 12$

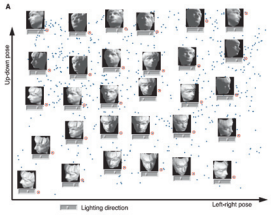




wrist images
$m = 2000$
$k = 6$
$D = 64^2$

Examples

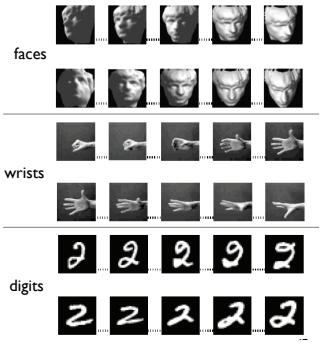
face images
$m = 698$
$k = 6$



digit images
$m = 1000$
$\epsilon = 4.2$
$D = 20^2$

Interpolations

- Linear in Isomap feature space
- Nonlinear in pixel space

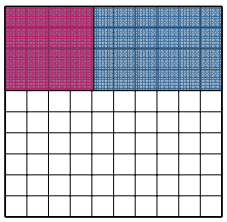


Properties of Isomap

- Strengths
  - Polynomial-time optimizations
  - No local minima
  - Non-iterative (one pass through data)
  - Non-parametric
  - Only heuristic is neighborhood size
- Weaknesses
  - Sensitive to "shortcuts"
  - No out-of-sample extension

Large-Scale Applications

- Problem
  - Too expensive to compute all shortest paths and diagonalize full Gram matrix
- Solution
  - Only compute shortest paths in blue and diagonalize sub-matrix in red



$m \times m$ Gram matrix

Landmark Isomap

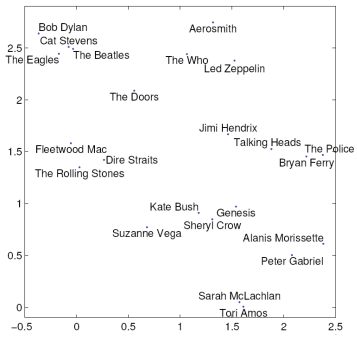
- Approximation
  - Identify subset of inputs as landmarks
  - Estimate geodesics to/from landmarks
  - Apply MDS to landmark distances
  - Embed non-landmarks by triangulation
  - Related to Nystrom approximation
- Computation
  - Reduced by l/m for $l<m$ landmarks
  - Reconstructs large Gram matrix from thin rectangular sub-matrix

Example

- Embedding of sparse music similarity graph

$m = 267K$
$e = 3.22M$
$\ell = 400$
$\tau = 6$ minutes

(Platt, 2004)

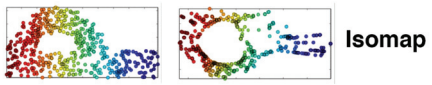


Theoretical Guarantees

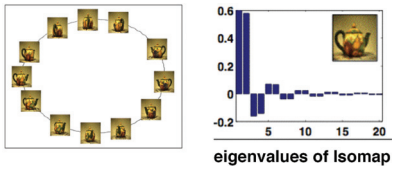
- Asymptotic convergence
  - For data sampled from a submanifold that is isometric to a convex subset of Euclidean space, Isomap will recover the subset up to rotation and translation
    (Tenenbaum et al.; Donoho & Grimes)
- Convexity assumption
  - Geodesic distances are not estimated correctly for manifolds with holes

Connected but Not Convex

- 2-D region with hole



- Images of 360 degree rotated teapot



eigenvalues of Isomap

Spectral Methods

- Common framework
  - Derive sparse graph from kNN
  - Derive matrix from graph weights
  - Derive embedding from eigenvectors
- Varied solutions
  - Algorithms differ in step 2
  - Types of optimization: shortest paths, least squares fitting, semidefinite programming

Graph-Based Methods #2

Locally Linear Embedding (LLE)

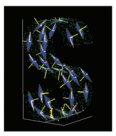
(Roweis and Saul, 2000)

Local Methods

- MDS and Isomap
  - Preserve global pairwise distances
  - Construct large, dense matrices
  - Compute top eigenvectors
- "Local" methods
  - Preserve local geometric relationships
  - Construct large, sparse matrices
  - Compute bottom eigenvectors

How to Exploit Local Linearity?

- Manifolds are globally nonlinear, but locally linear
- Map the inputs into a single continuous global coordinate system of lower dimensionality
  - **Think globally, fit locally**



LLE Algorithm

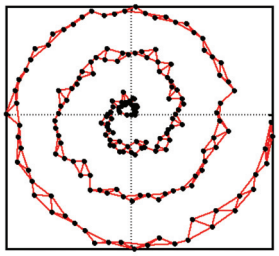
- Steps
  (1) Nearest neighbor search
  (2) Least squares fits
  (3) Sparse eigenvalue problem
- Properties
  - Obtains highly nonlinear embeddings
  - Not prone to local minima
  - Sparse graphs yield sparse eigenvalue problems

Step 1. Identify Neighbors

- Examples of neighborhoods
  - k-nearest neighbors
  - Neighbors within an ϵ-ball
  - Metric based on prior knowledge
- Assumptions
  - Data is sampled from a manifold
  - Manifold is well sampled

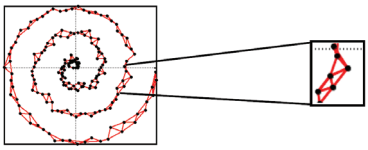
Nearest Neighbor Graph

- Assumptions
  - Graph is connected
  - Neighborhoods on the graph correspond to neighborhoods on the manifold



Step 2. Compute Weights

- Characterize local geometry of each neighborhood by weights W_{ij}



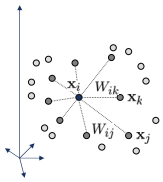
- Compute weights by reconstructing each input (linearly) from neighbors

Linear Reconstructions

- Local linearity
  - Neighbors lie on locally linear patches of a low-dimensional manifold
- Reconstruction errors
  - Least squared errors should be small

$$\Phi(W) = \sum_i \left| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right|^2$$

(in each neighborhood)



Least Squares Fits

- Local reconstructions
  - Choose weights W to minimize the quadratic form:

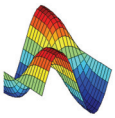
$$\Phi(W) = \sum_i \left| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right|^2$$

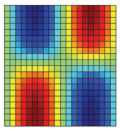
- Constraints
  - Nonzero W_{ij} only for neighbors
  - Weights must sum to one: $\sum_j W_{ij} = 1$

(to carry some useful information)

- Local invariance
  - Optimal weights W_{ij} are invariant to rotation, translation, and scaling

Symmetries





- Local linearity
  - If each neighborhood map looks like a translation, rotation, and rescaling…
- Local geometry
  - …then these transformations do not affect the weights W_{ij}: they remain valid

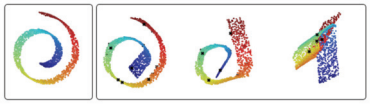
Thought Experiment

1/2

- Reconstruction from landmarks
  - Clamp subset of inputs ("landmarks"), then reconstruct others by minimizing

$$\Phi(W) = \sum_i \left| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right|^2 \text{ with respect to } \mathbf{x}_i$$

$m = 2000$ inputs



number of landmarks: $L = 15$, $L = 10$, $L = 5$

Thought Experiment

2/2

- Locally linear reconstruction
  - Very accurate for sufficiently large number of landmarks
  - Increasingly linearized with decreasing number of landmarks

number of landmarks: $L = 15$, $L = 10$, $L = 5$, $L = 0$?

Step 3. "Linearization"

- Low-dimensional representation
  - Map inputs to outputs: $\mathbf{x}_i \in \mathbb{R}^D$ to $\mathbf{y}_i \in \mathbb{R}^d$
- Minimize reconstruction errors
  - Optimize outputs for fixed weights:

$$\Psi(\mathbf{y}) = \sum_i \left| \mathbf{y}_i - \sum_j W_{ij} \mathbf{y}_j \right|^2$$

- Constraints
  - Center outputs on origin: $\sum_i \mathbf{y}_i = 0$
  - Impose unit covariance matrix: $\frac{1}{m} \sum_i \mathbf{y}_i \mathbf{y}_i^T = I_d$

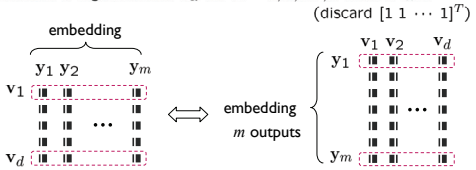
Sparse Eigenvalue Problem

- Quadratic form
  - $\Psi(\mathbf{y}) = \sum_{i,j} \left(\mathbf{y}_i \cdot \mathbf{y}_j \right) \Psi_{ij}$ with $\Psi = (I - W)^T (I - W)$
- Rayleigh–Ritz quotient
  - Optimal embedding given by bottom $d + 1$ eigenvectors
- Solution
  - Discard bottom eigenvector $[1\ 1\ \cdots\ 1]^T$
  - Other eigenvectors satisfy unit covariance constraints (orthonormal)

Unit Covariance Matrix

bottom d eigenvectors $\mathbf{v}_\alpha$ for $\alpha = 1, 2, \ldots, d$ with $d \ll m$

(discard $[1\ 1\ \cdots\ 1]^T$)



$$A = \sum_i \mathbf{y}_i \mathbf{y}_i^T \in \mathbb{R}^{d \times d}$$

$$\text{element } A_{\alpha\beta} = \sum_i y_{i\alpha} y_{i\beta} = \sum_i v_{\alpha i} v_{\beta i} = \mathbf{v}_\alpha^T \mathbf{v}_\beta$$

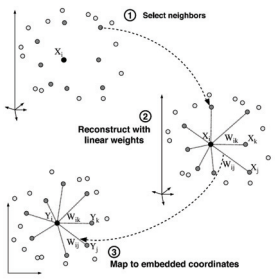
$$\sum_i \mathbf{y}_i \mathbf{y}_i^T = I_d$$

Summary of LLE

- Three steps
  1. Compute k-nearest neighbors
  2. Compute weights W_{ij}
  3. Compute outputs $\mathbf{y}_i$
- Optimizations

$$\Phi(W) = \sum_i \left| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right|^2$$

$$\Psi(\mathbf{y}) = \sum_i \left| \mathbf{y}_i - \sum_j W_{ij} \mathbf{y}_j \right|^2$$

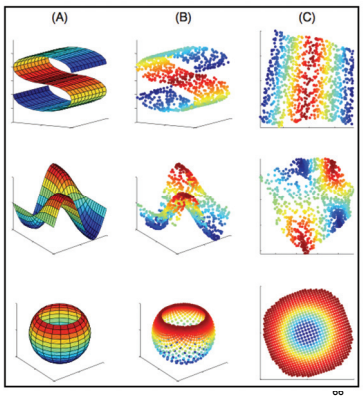


Surfaces

$m = 1000$ inputs

$k = 8$ nearest neighbors

$D = 3$
$d = 2$
dimensions



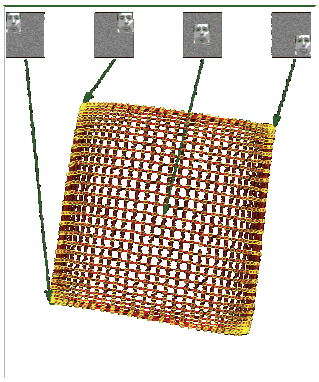
Translated Faces

$m = 961$
images

$k = 4$
nearest
neighbors

$D = 3009$
pixels

$d = 2$
manifold



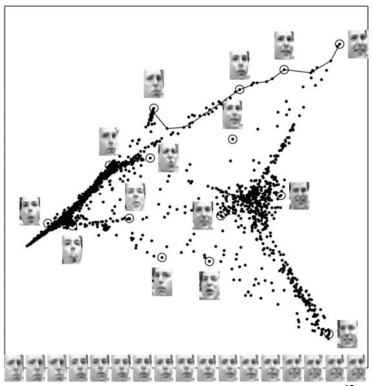
Pose and Expression

$m = 1965$
images

$k = 12$
nearest
neighbors

$D = 560$
pixels

$d = 2$
(shown)



68

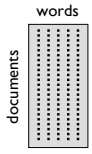
Word-Document Counts

$m = 5000$
words

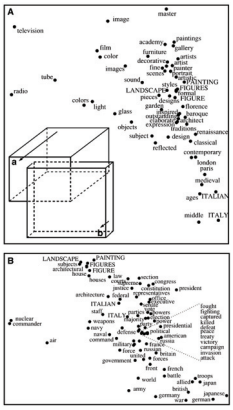
$k = 20$
nearest
neighbors

$D = 31000$
documents

$d = 3, 4, 5$
(shown)



semantic similarity?



69

Properties of LLE

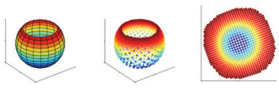
- Strengths
  - Polynomial-time optimizations
  - No local minima
  - Non-iterative (one pass through data)
  - Non-parametric
  - Only heuristic is neighborhood size
- Weaknesses
  - Sensitive to "shortcuts"
  - No out-of-sample extension
  - No estimate of dimensionality

    no significant gaps in the eigenvalue spectrum

70

LLE vs. Isomap

- Many similarities
  - Graph-based, spectral method
  - No local minima
- Essential differences
  - Does not estimate dimensionality
  - No theoretical guarantees
  - Constructs sparse vs. dense matrix
  - Preserves weights vs. distances



71