Spectral Methods for Dimensionality Reduction

Prof. Lawrence Saul

Dept of Computer & Information Science
University of Pennsylvania

UCLA IPAM Tutorial, July 11-14, 2005

Penn
UNIVERSITY of PENNSYLVANIA

Institute for Pure & Applied Mathematics
IPAM
University of California, Los Angeles

Dimensionality reduction

- Question

  How can we detect low dimensional structure in high dimensional data?

- Applications
  - Digital image and speech libraries
  - Neuronal population activities
  - Gene expression microarrays
  - Financial time series

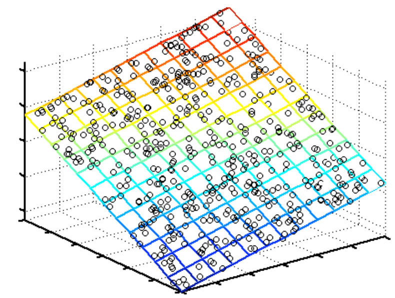
Framework

- **Data representation**

  Inputs are real-valued vectors in a high dimensional space.

- **Linear structure**

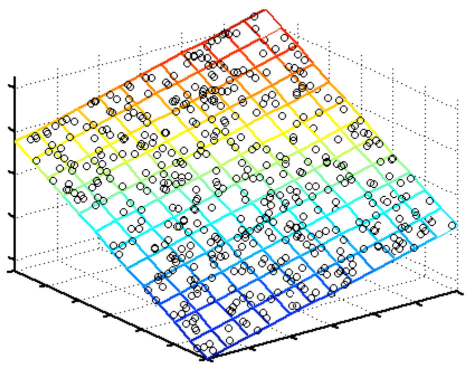
  Does the data live in a low dimensional subspace?



- **Nonlinear structure**

  Does the data live on a low dimensional submanifold?

Linear vs nonlinear



What computational price must we pay for nonlinear dimensionality reduction?

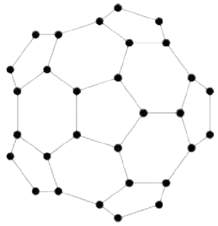
Spectral methods

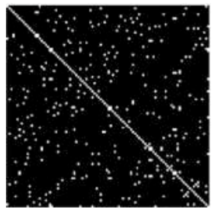
- **Matrix analysis**

  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 structure.

Notation

- **Inputs (high dimensional)**

  $$x_i \in \Re^D \text{ with } i = 1, 2, \ldots, n$$

- **Outputs (low dimensional)**

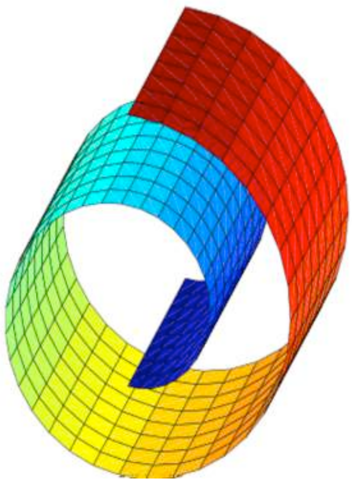
  $$y_i \in \Re^d \text{ where } d \square D$$

- **Goals**

  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?



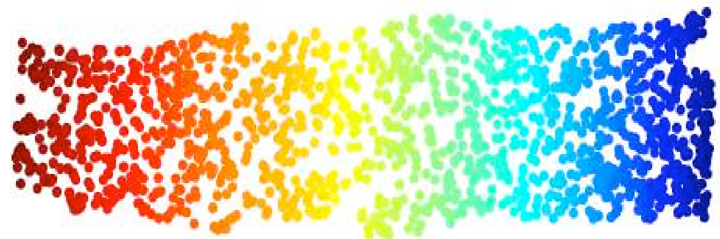
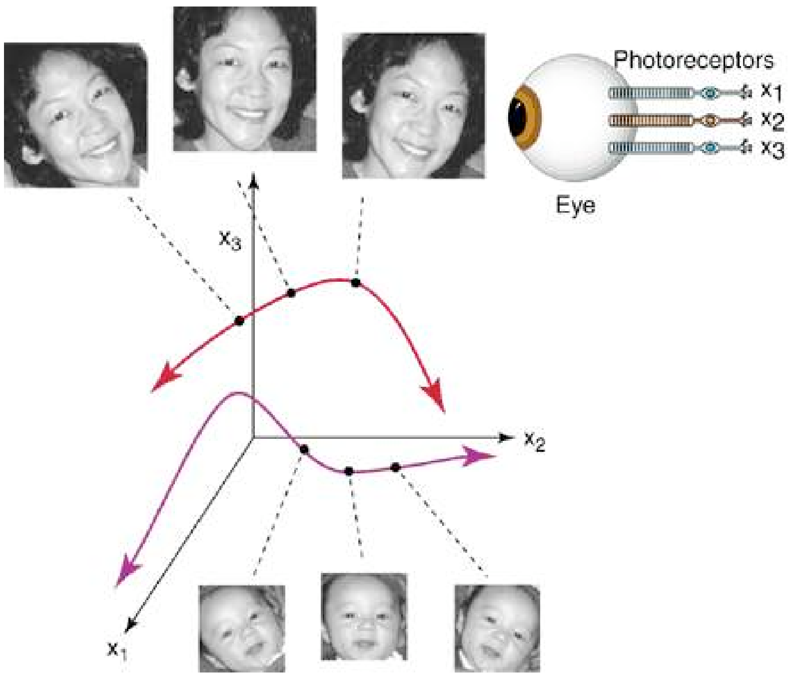
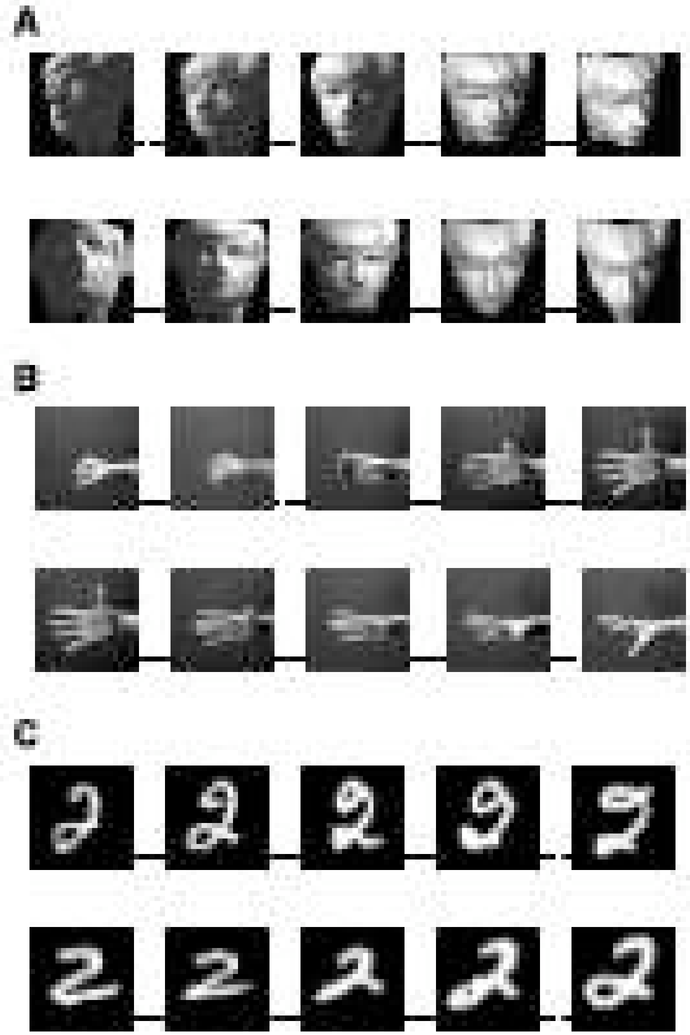


Image Manifolds





(Seung & Lee, 2000)
(Tenenbaum et al, 2000)

Outline

- **Day 1** - linear, nonlinear, and graph-based methods
- **Day 2** - sparse matrix methods
- **Day 3** - semidefinite programming
- **Day 4** - kernel methods

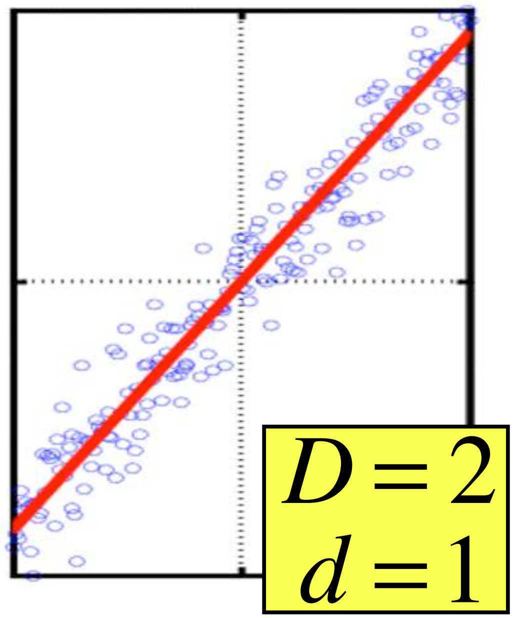
Questions for today

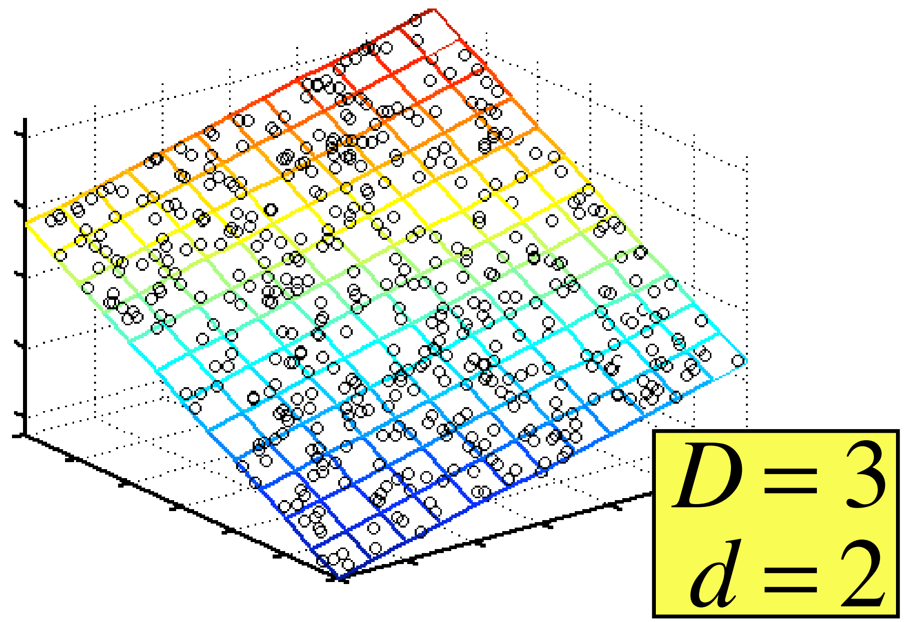
- **How to detect linear structure?**
  - principal components analysis
  - metric multidimensional scaling
- **How (not) to generalize these methods?**
  - neural network autoencoders
  - nonmetric multidimensional scaling
- **How to detect nonlinear structure?**
  - graphs as discretized manifolds
  - Isomap algorithm

Linear method #1

Principal Components Analysis (PCA)

Principal components analysis





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

Maximum variance subspace

- **Assume inputs are centered:**

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

- **Project into subspace:**

$$\vec{y}_i = P\vec{x}_i \quad \text{with} \quad P^2 = P$$

- **Maximize projected variance:**

$$\mathrm{var}(\vec{y}) = \frac{1}{n}\sum_i \| P\vec{x}_i \|^2$$

Matrix diagonalization

- **Covariance matrix**

$$\mathrm{var}(\vec{y}) = \mathrm{Tr}(PCP^{\mathrm{T}}) \quad \text{with} \quad C = n^{-1} \sum_i \vec{x}_i \vec{x}_i^{T}$$

- **Spectral decomposition**

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

- **Maximum variance projection**

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

Projects into subspace spanned by top d eigenvectors.

Interpreting PCA

- **Eigenvectors:**

  principal axes of maximum variance subspace.

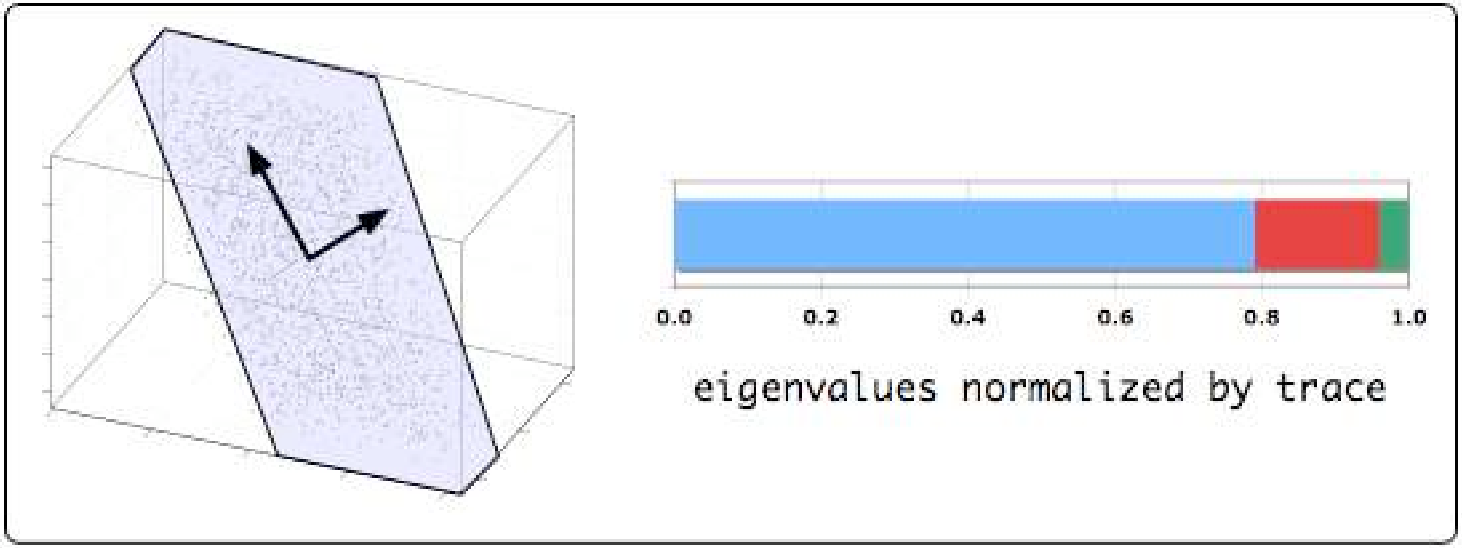
- **Eigenvalues:**

  projected variance of inputs along principle axes.

- **Estimated dimensionality:**

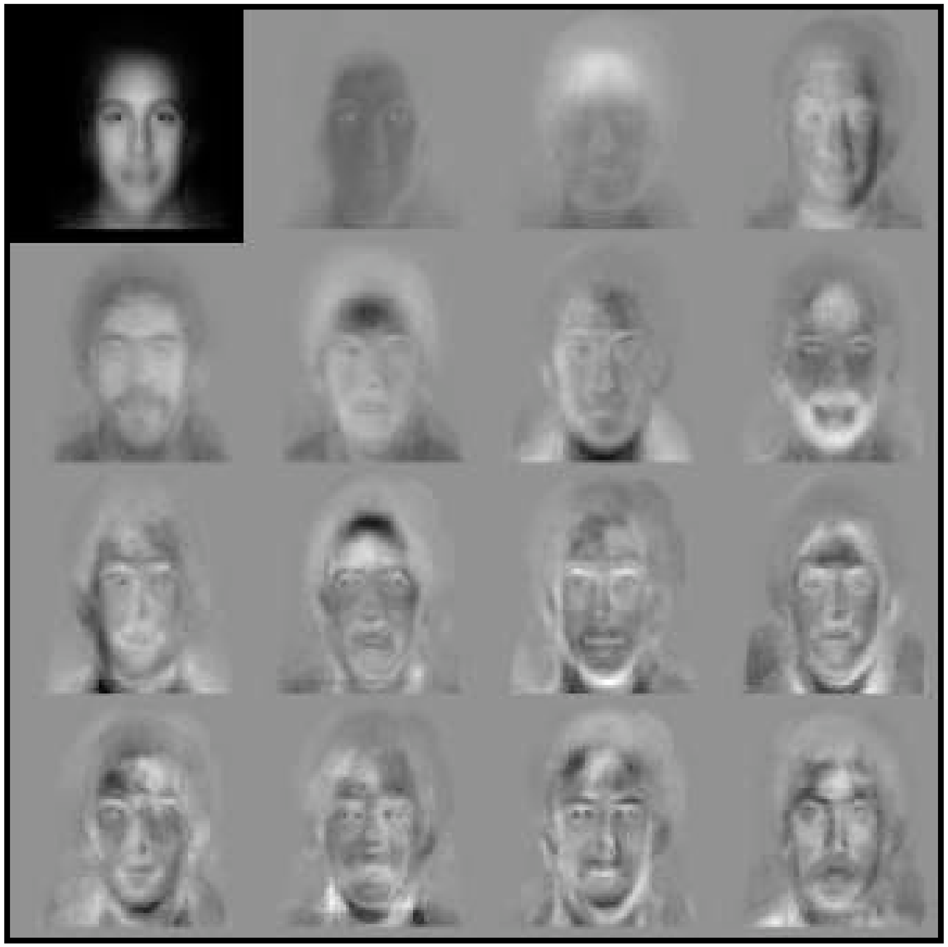
  number of significant (nonnegative) eigenvalues.

Example of PCA



Eigenvectors and eigenvalues of covariance matrix for $n=1600$ inputs in $d=3$ dimensions.

Example: faces



Eigenfaces from 7562 images:

top left image is linear combination of rest.

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

Another interpretation of PCA:

- **Assume inputs are centered:**

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

- **Project into subspace:**

$$\vec{y}_i = P\vec{x}_i \quad \text{with} \quad P^2 = P$$

- **Minimize reconstruction error:**

$$\mathrm{err}(\vec{y}) = n^{-1}\sum_i \| \vec{x}_i - P\vec{x}_i \|^2$$

Equivalence

- **Minimum reconstruction error:**

$$\mathrm{err}(\vec{y}) = n^{-1} \sum_i \| \vec{x}_i - P\vec{x}_i \|^2$$

- **Maximum variance subspace**

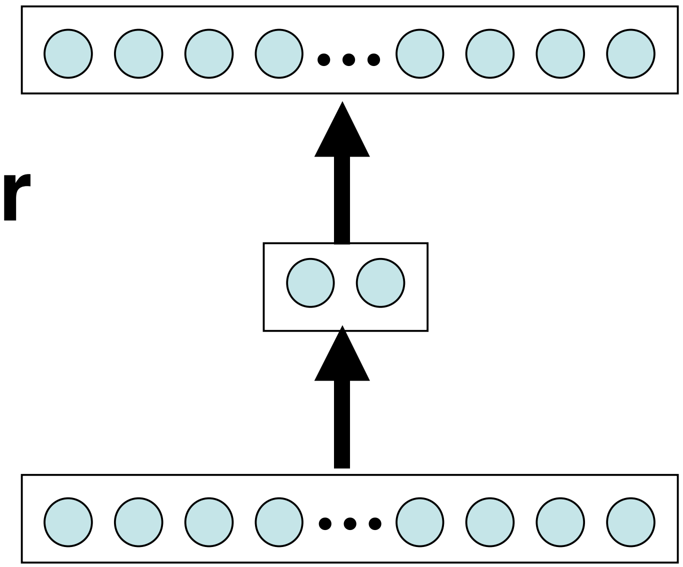
$$\mathrm{var}(\vec{y}) = n^{-1} \sum_i \| P\vec{x}_i \|^2$$

Both models for linear dimensionality reduction yield the same solution.

PCA as linear autoencoder

- **Network**

  Each layer implements a linear transformation.



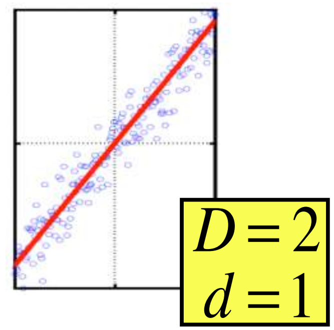
- **Cost function**

  Minimize reconstruction error through bottleneck:

$$\mathrm{err}(P) = n^{-1} \sum_i \left\| \vec{x}_i - P^{\mathrm{T}} P \vec{x}_i \right\|^2$$

Summary of PCA

1) **Center inputs on origin.**
2) **Compute covariance matrix.**
3) **Diagonalize.**
4) **Project.**

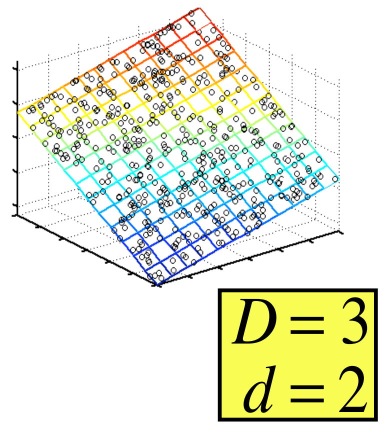


$$1)\quad \vec{0} = \sum_i \vec{x}_i$$

$$2)\quad C = n^{-1} \sum_i \vec{x}_i \vec{x}_i^{\mathrm{T}}$$

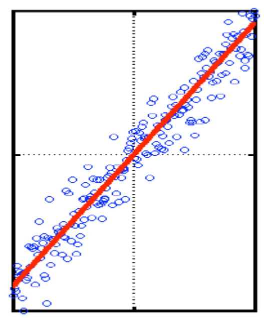
$$3)\quad C = \sum_\alpha \lambda_\alpha \vec{e}_\alpha \vec{e}_\alpha^{\mathrm{T}}$$

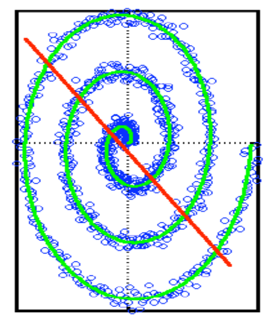
$$4)\quad \vec{y}_i = P\vec{x}_i \quad \text{with} \quad P = \sum_{\alpha \le d} \vec{e}_\alpha \vec{e}_\alpha^{\mathrm{T}}$$



Properties of PCA

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



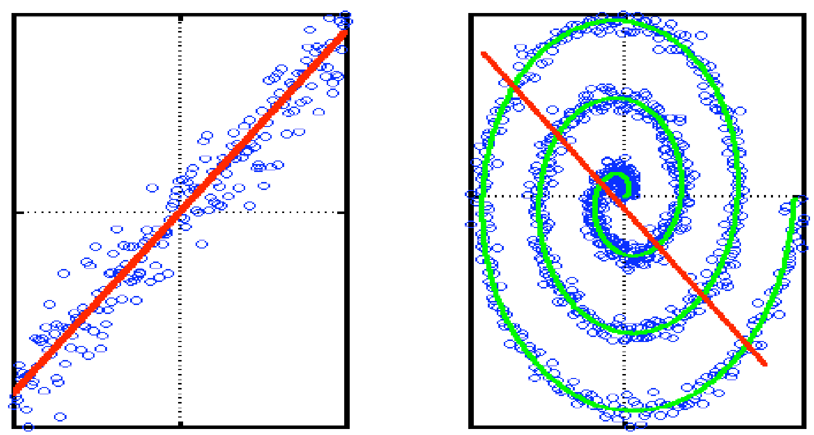


- **Weaknesses**
  - Limited to second order statistics
  - Limited to linear projections

So far..

- Q: How to detect linear structure?
  A: Principal components analysis
  - Maximum variance subspace
  - Minimum reconstruction error
  - Linear network autoencoders

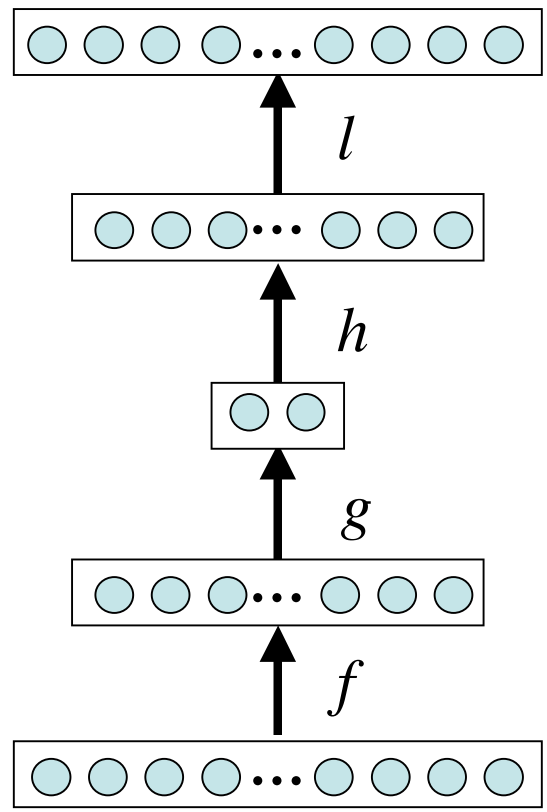
- Q: How (not) to generalize for manifolds?



Nonlinear autoencoder

- **Neural network**

  Each layer parameterizes a nonlinear transformation.



- **Cost function**

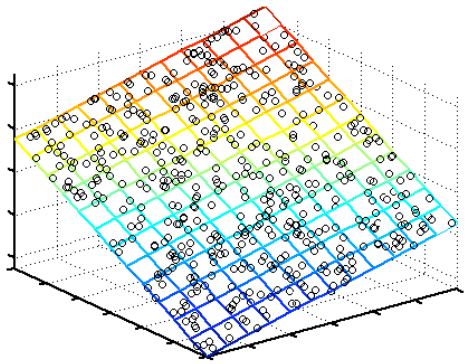
  Minimize reconstruction error:

$$\mathrm{err}(W) = n^{-1} \sum_i \left\| x_i - l_W(h_W(g_W(f_W(x_i) \right\|^2$$

Properties of neural network

- Strengths
  - Parameterizes nonlinear mapping (in both directions).
  - Generalizes to new inputs.
- Weaknesses
  - Many unspecified choices: network size, parameterization, learning rates.
  - Highly nonlinear, iterative optimization with local minima.

Linear vs nonlinear

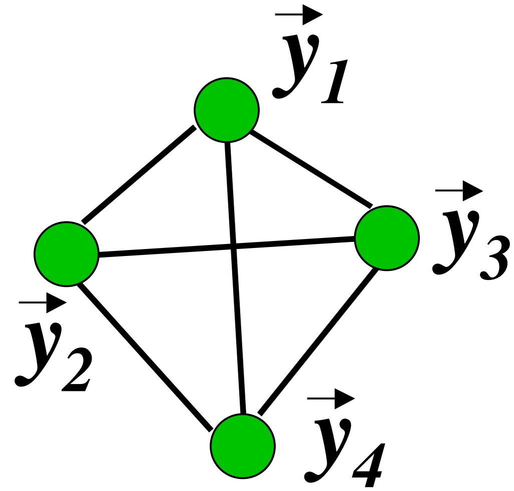
What computational price must we pay for nonlinear dimensionality reduction?

Linear method #2

Metric Multidimensional Scaling (MDS)

Multidimensional scaling

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



Given $n(n\text{-}1)/2$ pairwise distances Δ_{ij}, find vectors $\vec{y}_i$ such that $\|\vec{y}_i - \vec{y}_j\| \approx \Delta_{ij}$.

Metric Multidimensional Scaling

- **Lemma**

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

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

- **Optimization**

  Preserve dot products (proxy for distances). Choose vectors $\vec{y}_i$ to minimize:

$$\mathrm{err}(\vec{y}) = \sum_{ij} \left(G_{ij} - \vec{y}_i \cdot \vec{y}_j \right)^2$$

Matrix diagonalization

- **Gram matrix "matching"**

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

- **Spectral decomposition**

$$G = \sum_{\alpha=1}^{n} \lambda_\alpha \vec{v}_\alpha \vec{v}_\alpha^{\mathrm{T}} \quad \text{with} \quad \lambda_1 \geq \cdots \geq \lambda_n \geq 0$$

- **Optimal approximation**

$$\boxed{y_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i} \quad \text{for} \quad \alpha = 1, 2, \ldots, d \quad \text{with } d \leq n}$$

(scaled truncated eigenvectors)

Interpreting MDS

$$y_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i} \quad \text{for} \quad \alpha = 1, 2, \ldots, d \quad \text{with } d \,\square\, n$$

- **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**

$$C_{\alpha\beta} = n^{-1} \sum_i x_{i\alpha} x_{i\beta} \quad \text{covariance matrix } (D \times D)$$

$$G_{ij} = x_i \bullet x_j \quad \text{Gram matrix} \quad (n \times n)$$

- **Same eigenvalues**

Matrices share nonzero eigenvalues up to constant factor.

- **Same results, different computation**

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

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

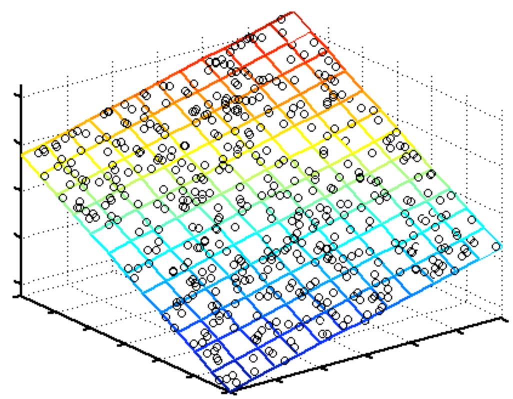
So far..

- Q: How to detect linear structure?

  A1: Principal components analysis

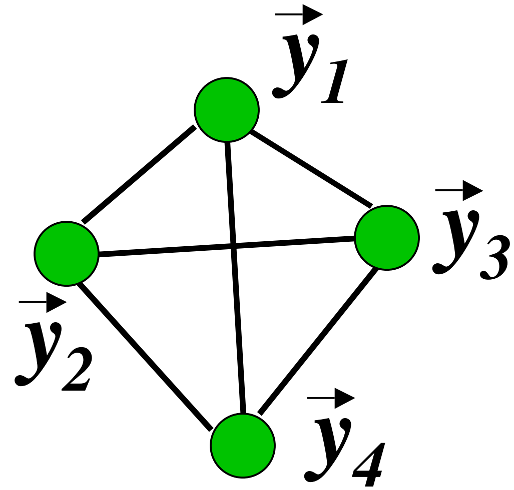
  A2: Metric multidimensional scaling

- Q: How (not) to generalize for manifolds?



Nonmetric 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: $\Delta_{ij} \longrightarrow g(\Delta_{ij})$.
Find vectors $\vec{y}_i$ such that $\|\vec{y}_i - \vec{y}_j\| \approx g(\Delta_{ij})$.

Non-Metric MDS

- **Distance transformation**

  Nonlinear, but monotonic.
  Preserves rank order of distances.

- **Optimization**

  Preserve transformed distances.
  Choose vectors $\vec{y}_i$ to minimize:

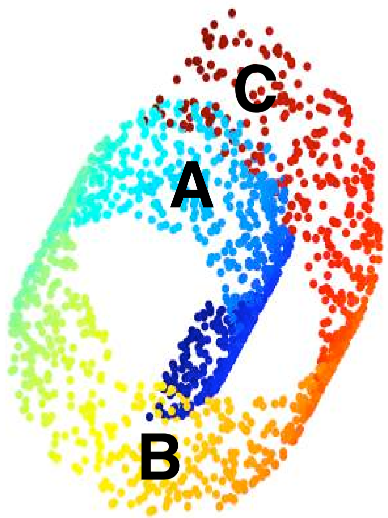
$$\mathrm{err}(\vec{y}) = \sum_{ij} \left(g(\Delta_{ij}) - \| \vec{y}_i - \vec{y}_j \| \right)^2$$

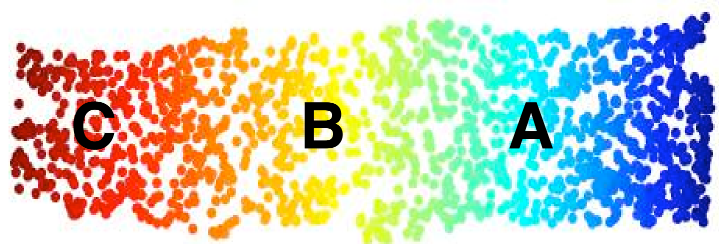
Properties of non-metric MDS

- 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".

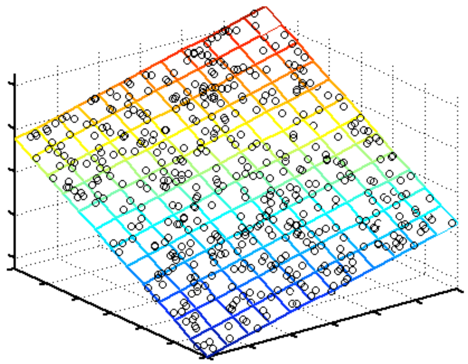




$d(A,C) < d(A,B)$

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

Linear vs nonlinear



What computational price must we pay for nonlinear dimensionality reduction?

Graph-based method #1

Isometric mapping of data manifolds (ISOMAP)

(Tenenbaum, de Silva, & Langford, 2000)

Dimensionality reduction

- **Inputs**

$$\vec{x}_i \in \Re^D \text{ with } i = 1, 2, \ldots, n$$

- **Outputs**

$$\vec{y}_i \in \Re^d \text{ where } d \ll D$$

- **Goals**

Nearby points remain nearby.
Distant points remain distant.
(Estimate *d*.)

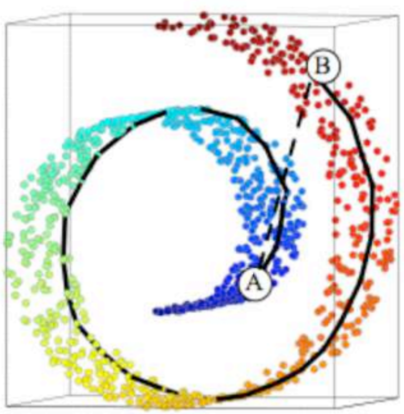
Isomap

- **Key idea:**

  Preserve geodesic distances as measured along submanifold.

- **Algorithm in a nutshell:**

  Use geodesic instead of (transformed) Euclidean distances in MDS.



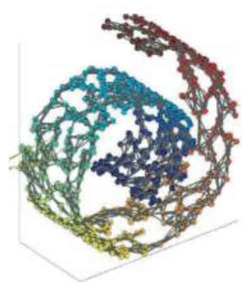
Step 1. Build adjacency graph.

- **Adjacency graph**

  Vertices represent inputs.
  Undirected edges connect neighbors.

- **Neighborhood selection**

  Many options: k-nearest neighbors, inputs within radius r, prior knowledge.



Graph is discretized approximation of submanifold.

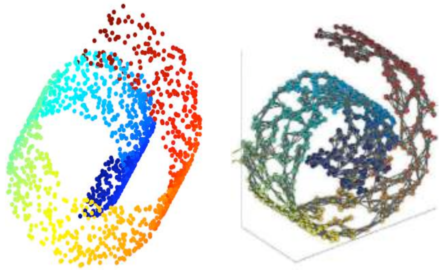
Building the graph

- **Computation**

  kNN scales naively as $O(n^2D)$.
  Faster methods exploit data structures.

- **Assumptions**

  1) Graph is connected.
  2) 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 distances.
  Compute shortest paths through graph.

- **Geodesic distances**

  Estimate by lengths Δ_{ij} of shortest paths:
  denser sampling = better estimates.

- **Computation**

  Djikstra's algorithm for shortest paths
  scales as $O(n^2 \log n + n^2 k)$.

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 $O(n^2d)$.

Summary

- **Algorithm**

  1) k nearest neighbors

  2) shortest paths through graph

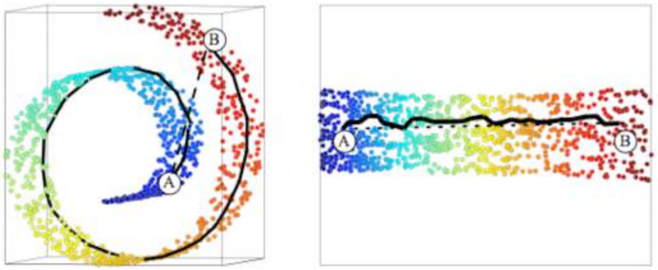
  3) MDS on geodesic distances

- **Impact**

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

Examples

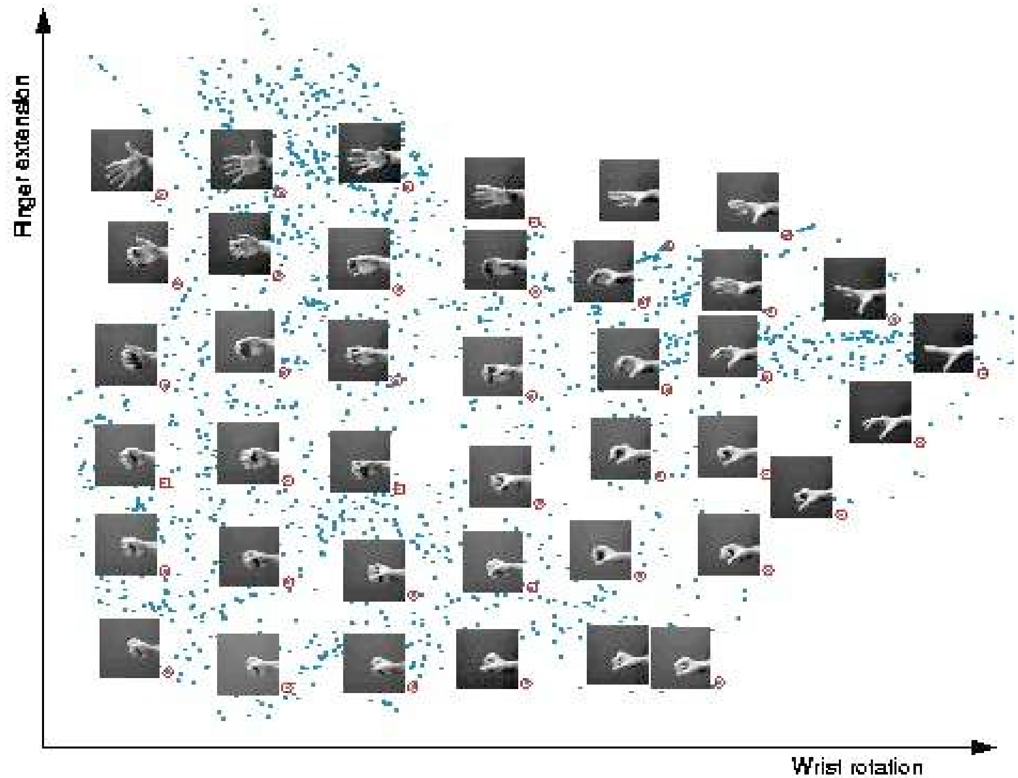
- **Swiss roll**



$$n = 1024$$
$$k = 12$$

- **Wrist images**

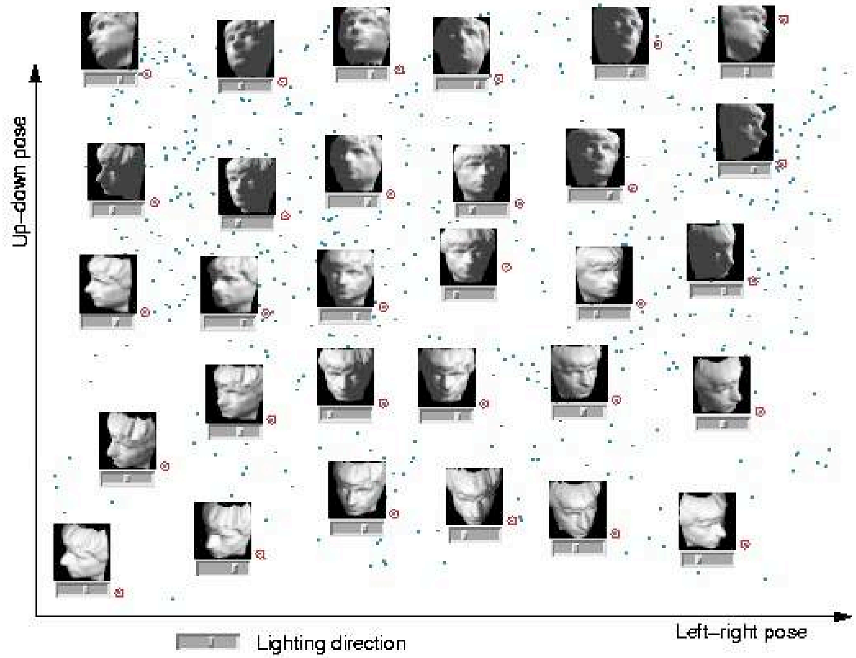
$$n = 2000$$
$$k = 6$$
$$D = 64^2$$



Examples

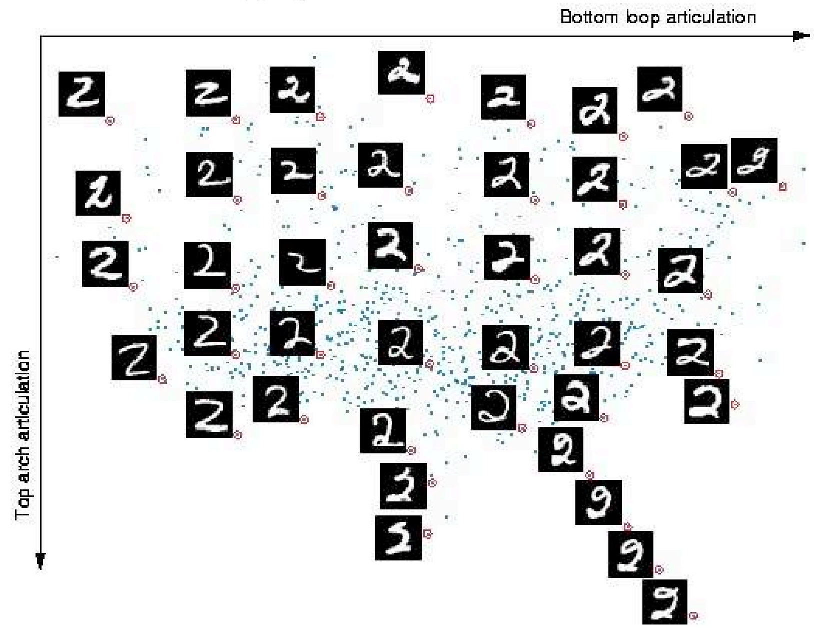
- **Face images**

$$n = 698$$
$$k = 6$$



- **Digit images**

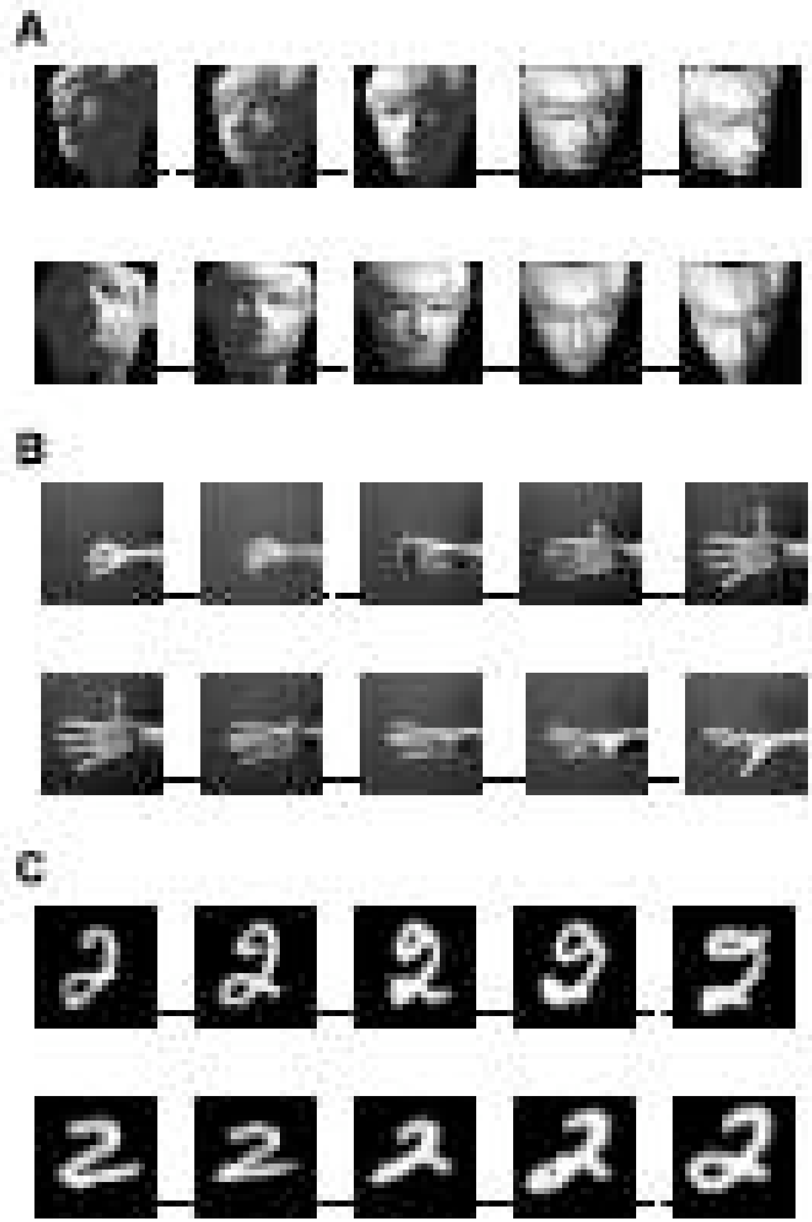
$$n = 1000$$
$$r = 4.2$$
$$D = 20^2$$



Interpolations

A. Faces
B. Wrists
C. Digits

Linear in Isomap feature space.
Nonlinear in pixel space.



Properties of Isomap

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

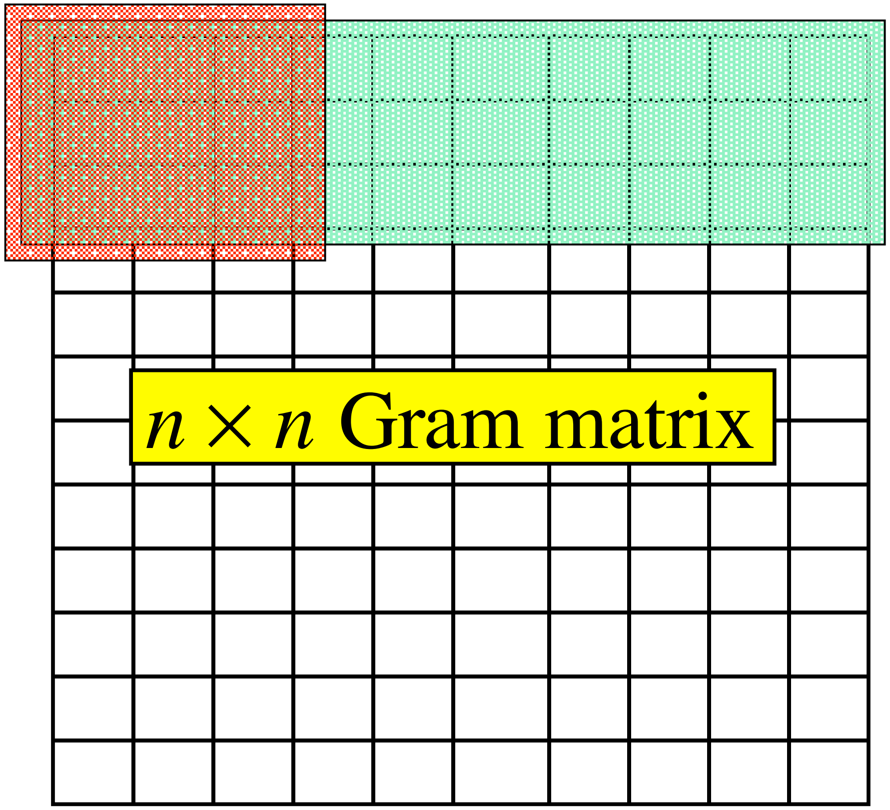
Large-scale applications

Problem:

Too expensive to compute all shortest paths and diagonalize full Gram matrix.

Solution:

Only compute shortest paths in green and diagonalize sub-matrix in red.



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/n for $l<n$ landmarks.
  - Reconstructs large Gram matrix from thin rectangular sub-matrix.

Example

Embedding of sparse music similarity graph

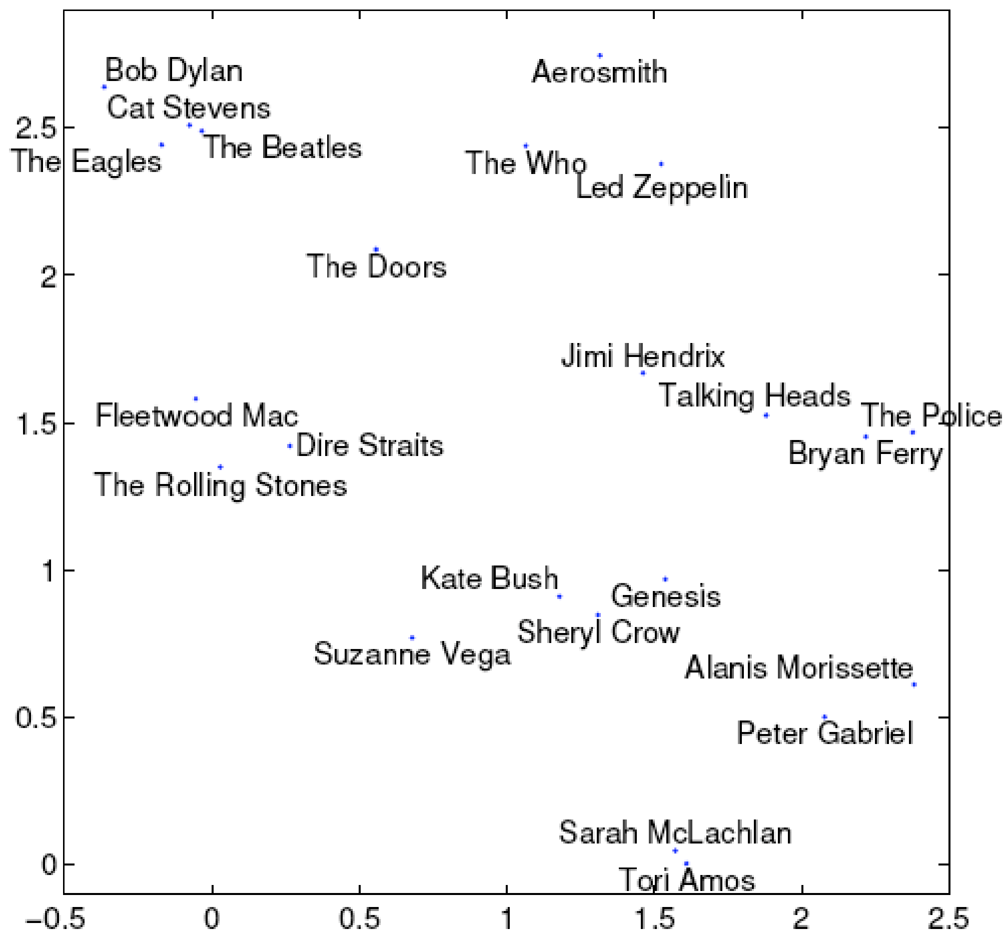
$n = 267\text{K}$

$e = 3.22\text{M}$

$\square = 400$

$\tau = 6$ 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 & translation.

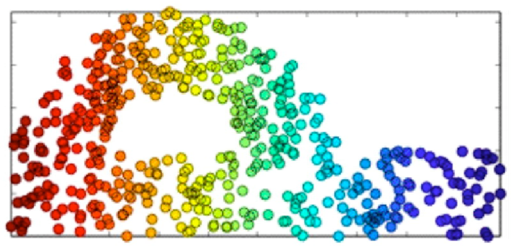
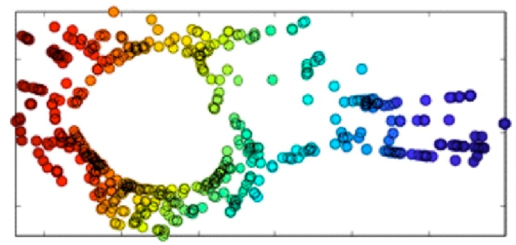
  (Tenenbaum et al; Donoho & Grimes)

- **Convexity assumption**

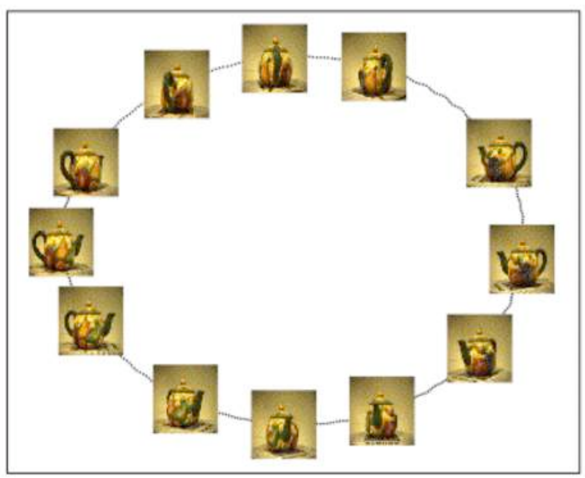
  Geodesic distances are not estimated correctly for manifolds with holes…

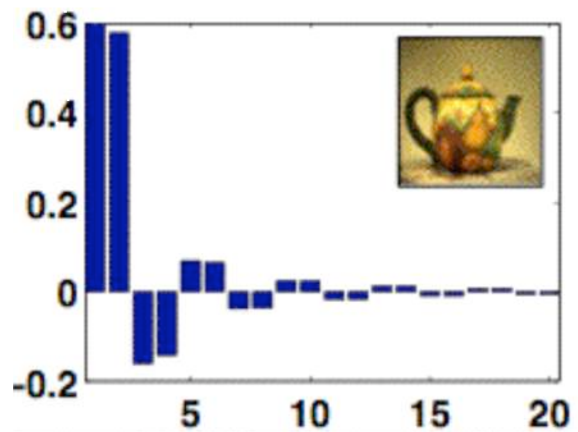
Connected but not convex

- 2d region with hole

input   Isomap

- Images of 360º rotated teapot



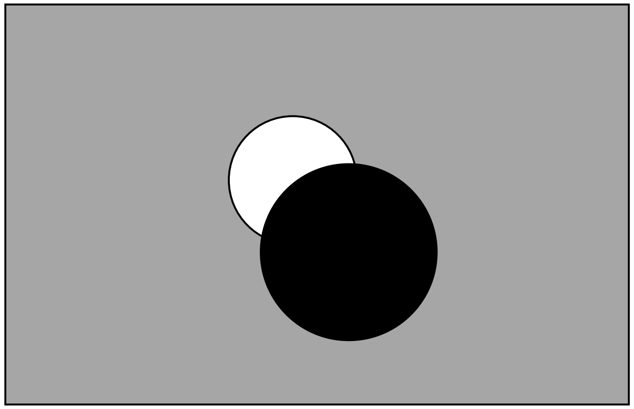


eigenvalues of Isomap

Connected but not convex

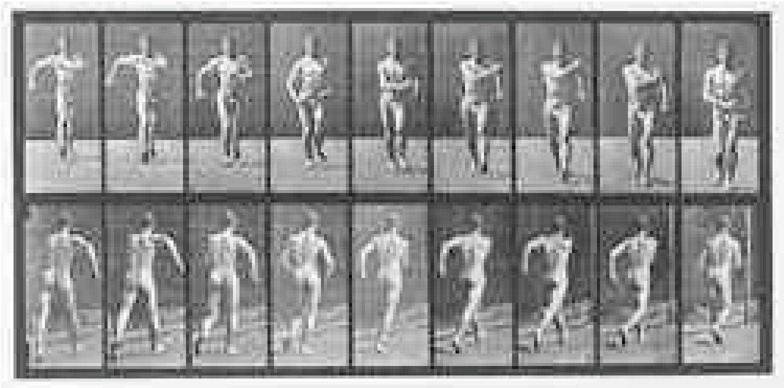
- **Occlusion**

  Images of two disks, one occluding the other.

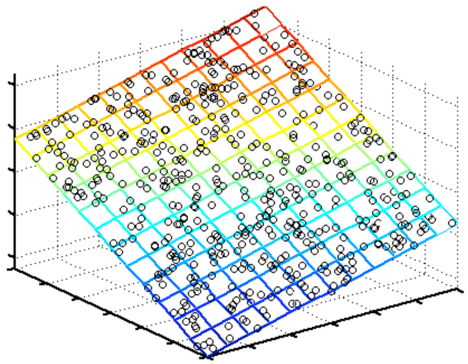


- **Locomotion**

  Images of periodic gait.



Linear vs nonlinear



What computational price must we pay for nonlinear dimensionality reduction?

Nonlinear dimensionality reduction since 2000…

These strengths and weaknesses are typical of graph-based spectral methods for dimensionality reduction.

Properties of Isomap

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

Spectral Methods

- **Common framework**
  1) Derive sparse graph from kNN.
  2) Derive matrix from graph weights.
  3) Derive embedding from eigenvectors.
- **Varied solutions**

  Algorithms differ in step 2.

  Types of optimization: shortest paths, least squares fits, semidefinite programming.

Algorithms

2000
Isomap
(Tenenbaum, de Silva, & Langford)

Locally Linear Embedding
(Roweis & Saul)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

2004
Maximum variance unfolding
(Weinberger & Saul)
(Sun, Boyd, Xiao, & Diaconis)

2005
Conformal eigenmaps
(Sha & Saul)

Looking ahead

- **Trade-offs**

  Sparse vs dense eigensystems?

  Preserving distances vs angles?

  Connected vs convex sets?

- **Connections**

  Spectral graph theory

  Convex optimization

  Differential geometry

Tuesday

2000
Isomap
(Tenenbaum, de Silva, & Langford)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

2004
Maximum variance unfolding
(Weinberger & Saul)
(Sun, Boyd, Xiao, & Diaconis)

2005
Conformal eigenmaps
(Sha & Saul)

Locally Linear Embedding
(Roweis & Saul)

Sparse Matrix Methods

Wednesday

2000
Isomap
(Tenenbaum, de Silva, & Langford)

Locally Linear Embedding
(Roweis & Saul)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

2004
Maximum variance unfolding
(Weinberger & Saul)
(Sun, Boyd, Xiao, & Diaconis)

2005
Conformal eigenmaps
(Sha & Saul)

Semidefinite Programming

To be continued...

See you tomorrow.