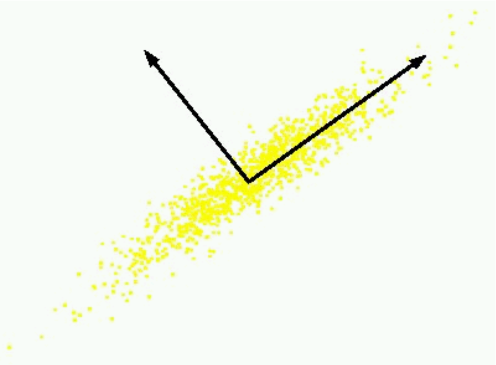
Machine Learning

Data visualization and dimensionality reduction

Eric Xing

Lecture 7, August 13, 2010



Text document retrieval/labelling

- Represent each document by a high-dimensional vector in the space of words

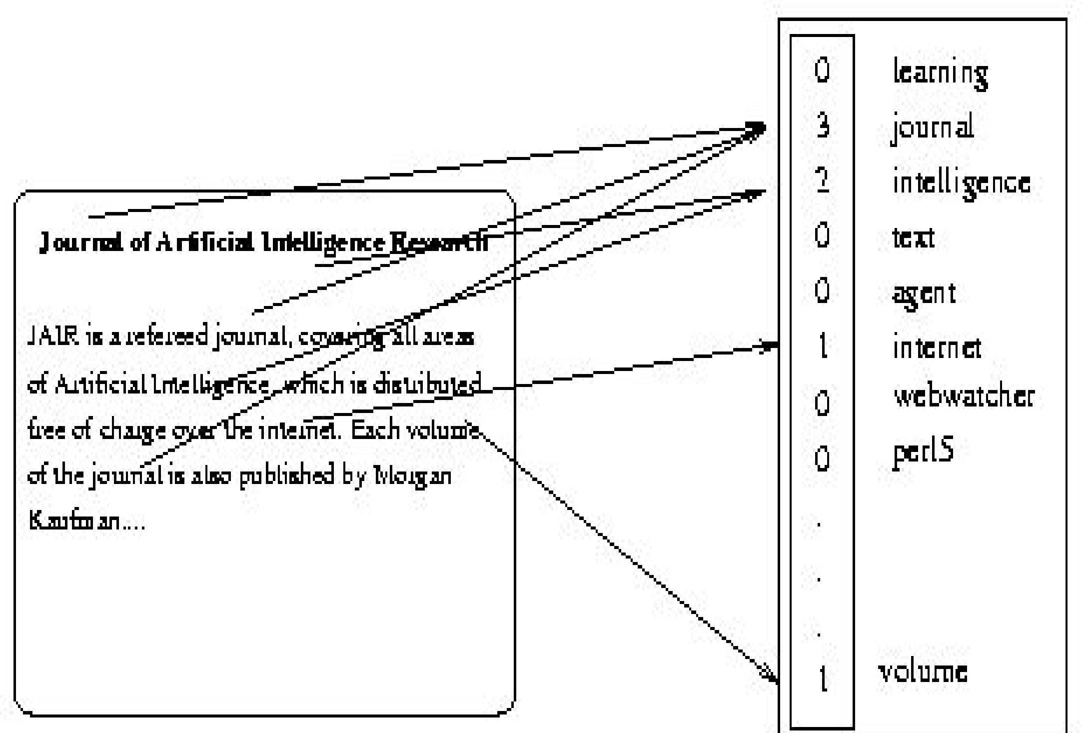


Image retrieval/labelling

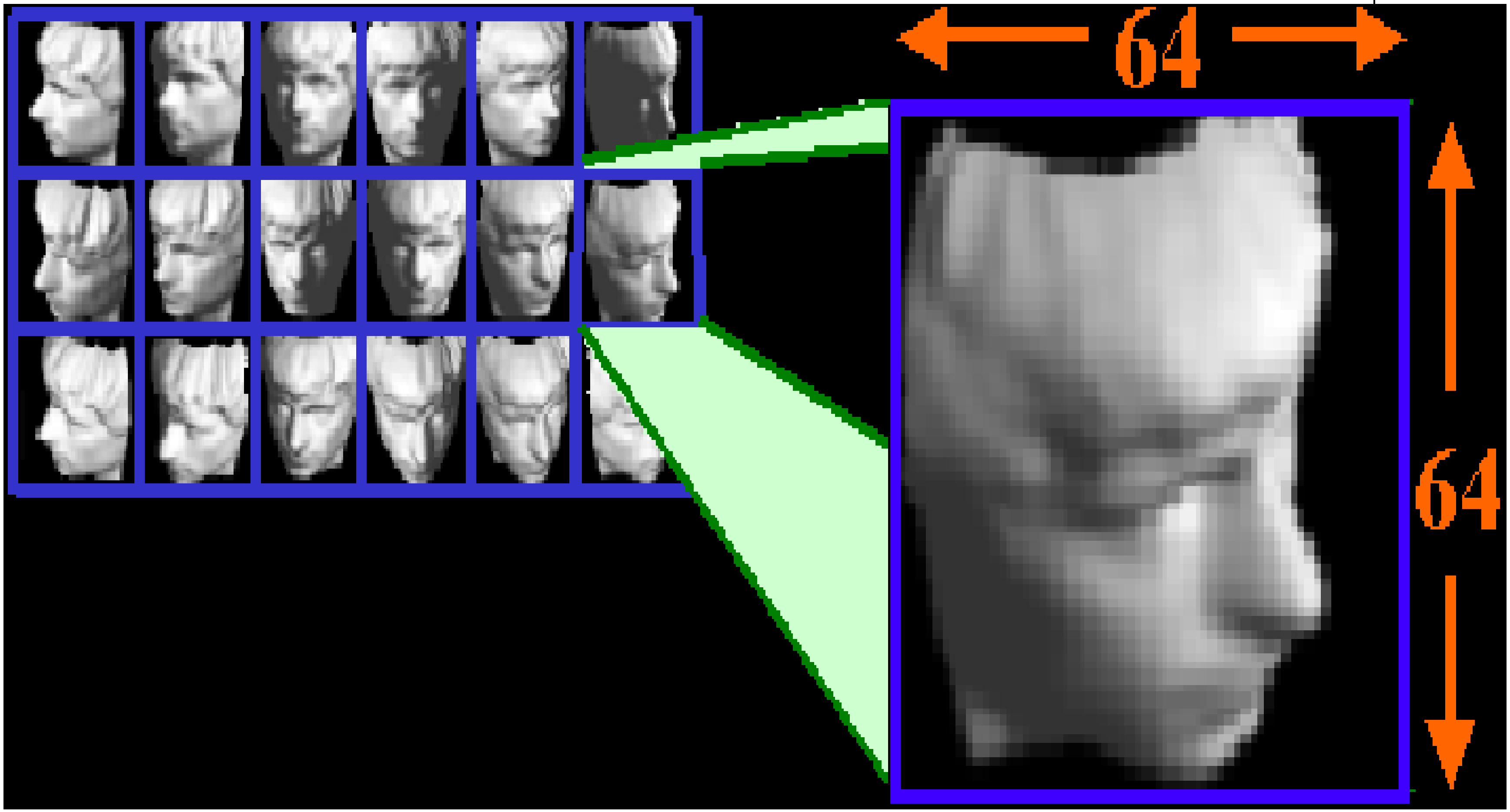
$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Dimensionality Bottlenecks

- Data dimension
  - Sensor response variables X:
    - 1,000,000 samples of an EM/Acoustic field on each of N sensors
    - 1024^2 pixels of a projected image on a IR camera sensor
    - N^2 expansion factor to account for all pairwise correlations
- Information dimension
  - Number of free parameters describing probability densities f(X) or f(S|X)
    - For known statistical model: info dim = model dim
    - For unknown model: info dim = dim of density approximation
- Parametric-model driven dimension reduction
  - DR by sufficiency, DR by maximum likelihood
- Data-driven dimension reduction
  - Manifold learning, structure discovery

Intuition: how does your brain store these pictures?

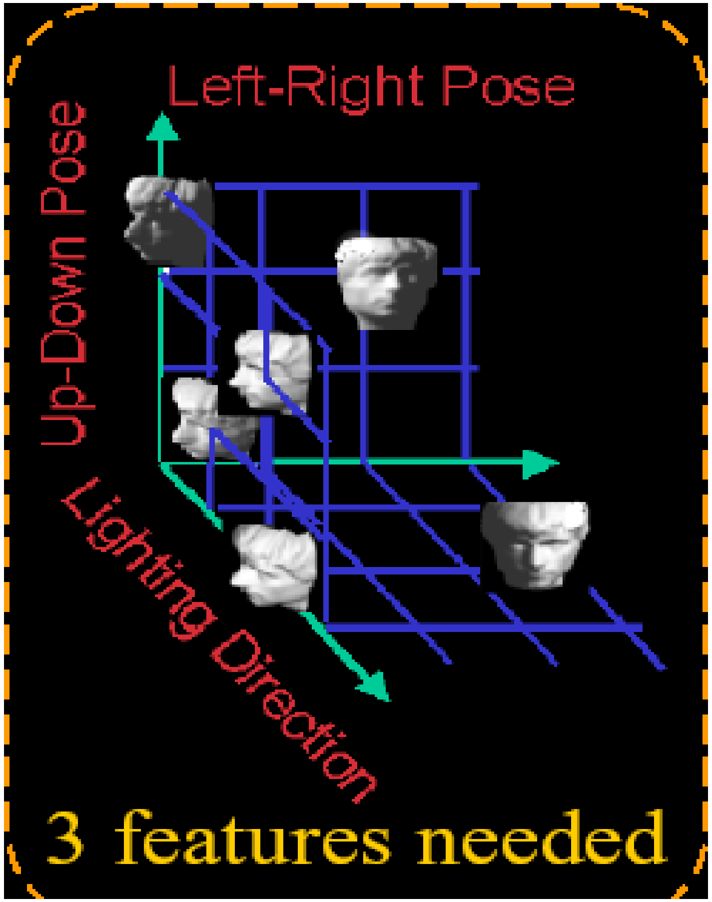
Brain Representation



Brain Representation

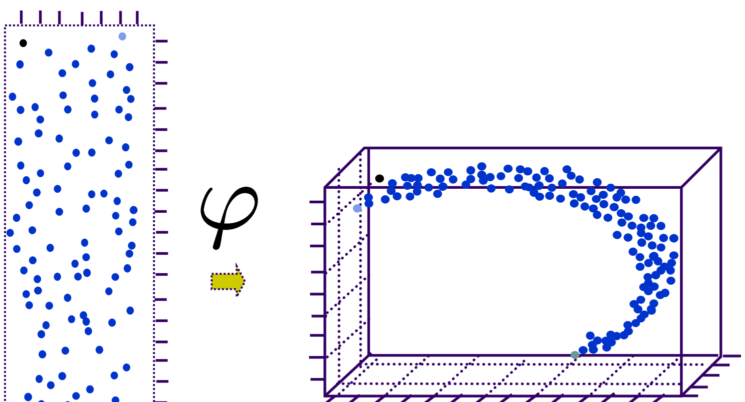
- Every pixel?
- Or perceptually meaningful structure?
  - Up-down pose
  - Left-right pose
  - Lighting direction

  So, your brain successfully reduced the high-dimensional inputs to an intrinsically 3-dimensional manifold!



Two Geometries to Consider

(Metric) data geometry



Domain

$\mathcal{Z}$

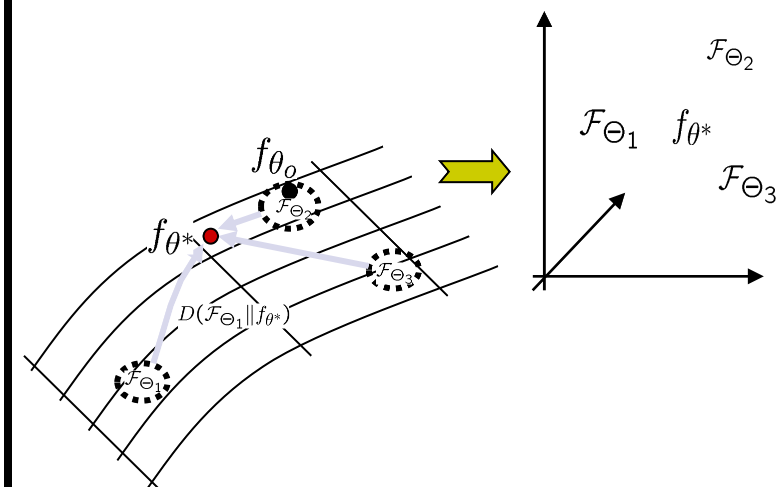
Manifold Embedding

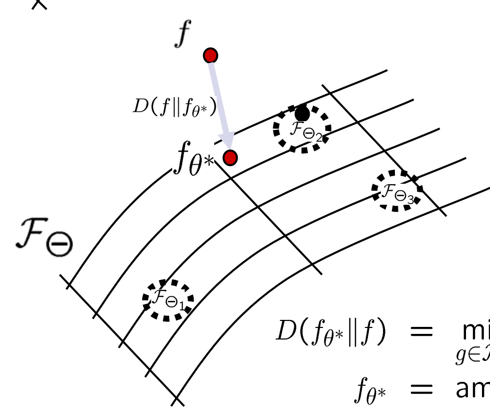
$\mathcal{M} \subset R^D$

$\{X_i \in \mathcal{M}\}_i$ are i.i.d. samples from $f_X(x)$

$$\begin{aligned} P(X \in B) &= \int_{B \cap \mathcal{M}} f_X(x)dx \\ &= \int_{\varphi^{-1}(B \cap \mathcal{M})} f_Z(z)dz \end{aligned}$$

(Non-metric) information geometry

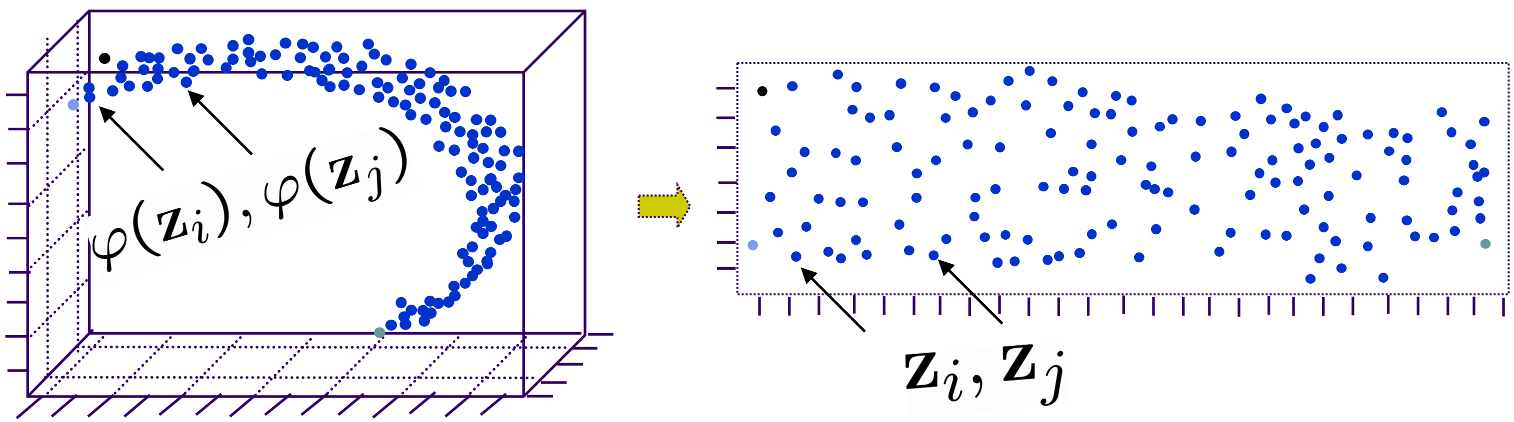




$$\begin{aligned} D(f_{\theta^*} \| f) &= \min_{g \in \mathcal{F}_\Theta} D(g \| f) \\ f_{\theta^*} &= \text{amin}_{g \in \mathcal{F}_\theta} D(g \| f) \end{aligned}$$

Data-driven DR

- Data-driven projection to lower dimensional subsapce
- Extract low-dim structure from high-dim data
- Data may lie on curved (but locally linear) subspace



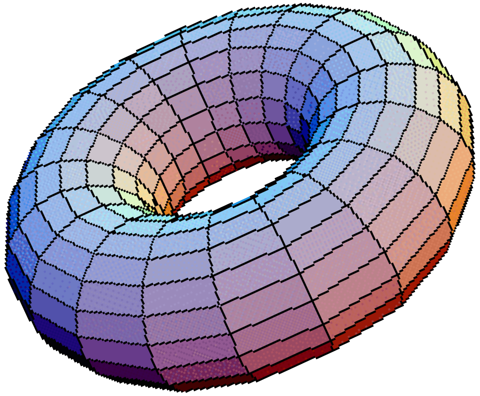
[1] Josh .B. Tenenbaum, Vin de Silva, and John C. Langford "A Global Geometric Framework for Nonlinear Dimensionality Reduction" *Science*, 22 Dec 2000.

[2] Jose Costa, Neal Patwari and Alfred O. Hero, "Distributed Weighted Multidimensional Scaling for Node Localization in Sensor Networks", *IEEE/ACM Trans. Sensor Networks*, to appear 2005.

[3] Misha Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, 2003.

What is a Manifold?

- A manifold is a topological space which is locally Euclidean.
- Represents a very useful and challenging unsupervised learning problem.
- In general, any object which is nearly "flat" on small scales is a manifold.



Manifold Learning

- Discover low dimensional structures (smooth manifold) for data in high dimension.

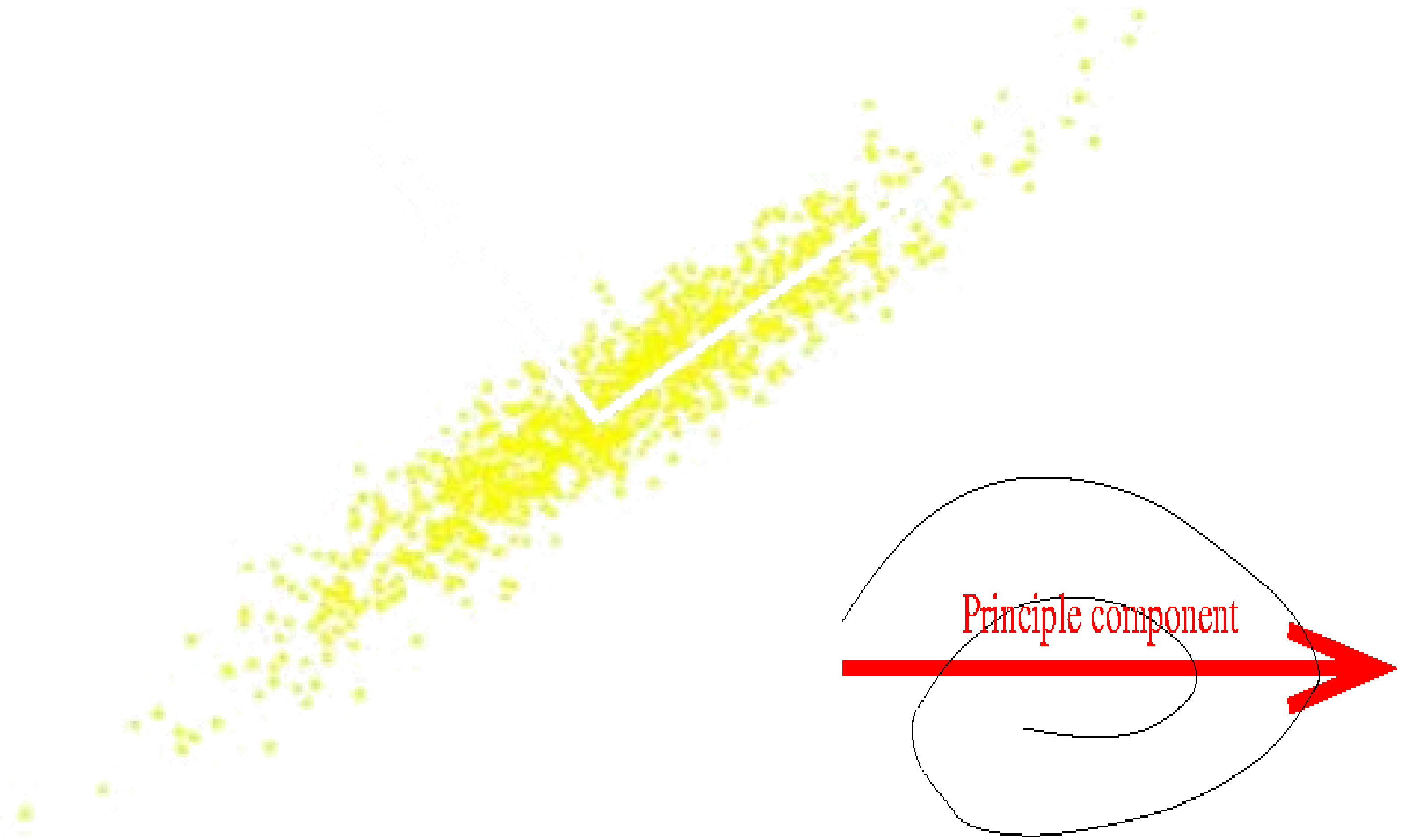
- Linear Approaches
  - Principal component analysis.
  - Multi dimensional scaling.

- Non Linear Approaches
  - Local Linear Embedding
  - ISOMAP
  - Laplacian Eigenmap.

Principal component analysis

- Areas of variance in data are where items can be best discriminated and key underlying phenomena observed
- If two items or dimensions are highly correlated or dependent
  - They are likely to represent highly related phenomena
  - We want to combine related variables, and focus on uncorrelated or independent ones, especially those along which the observations have high variance
- We look for the phenomena underlying the observed covariance/co-dependence in a set of variables
- These phenomena are called “factors” or “principal components” or “independent components,” depending on the methods used
  - Factor analysis: based on variance/covariance/correlation
  - Independent Component Analysis: based on independence

An example:

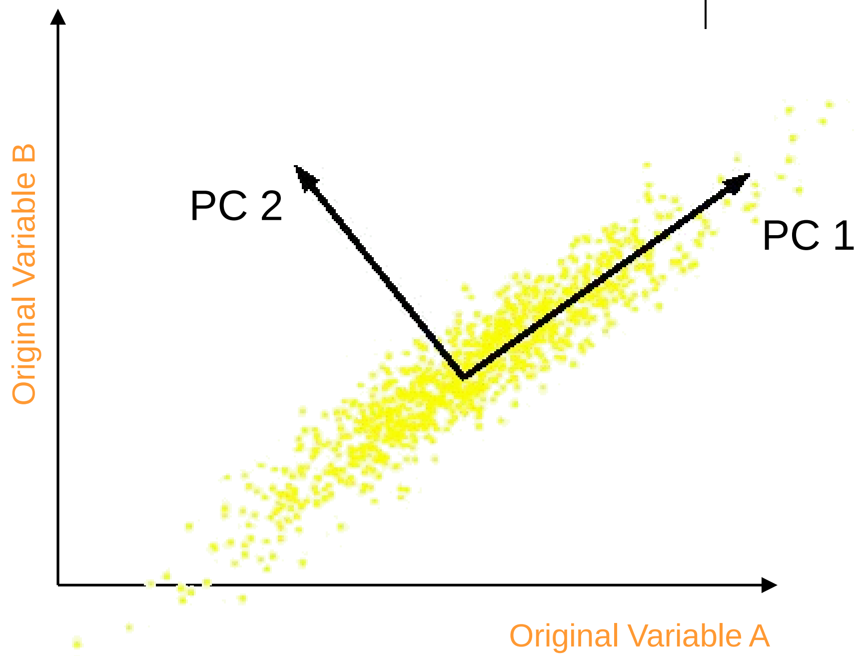


Principal Component Analysis

- The new variables/dimensions
  - Are linear combinations of the original ones
  - Are uncorrelated with one another
    - Orthogonal in original dimension space
  - Capture as much of the original variance in the data as possible
  - Are called Principal Components

- Orthogonal directions of greatest variance in data

- Projections along PC1 discriminate the data most along any one axis



- First principal component is the direction of greatest variability (covariance) in the data
- Second is the next orthogonal (uncorrelated) direction of greatest variability
  - So first remove all the variability along the first component, and then find the next direction of greatest variability
- And so on …

Computing the Components

- Projection of vector **x** onto an axis (dimension) **u** is $\mathbf{u}^T\mathbf{x}$
- Direction of greatest variability is that in which the average square of the projection is greatest:

Maximize $\mathbf{u}^T\mathbf{X}\mathbf{X}^T\mathbf{u}$

s.t $\mathbf{u}^T\mathbf{u} = 1$

Construct Langrangian $\mathbf{u}^T\mathbf{X}\mathbf{X}^T\mathbf{u} - \lambda\mathbf{u}^T\mathbf{u}$

Vector of partial derivatives set to zero

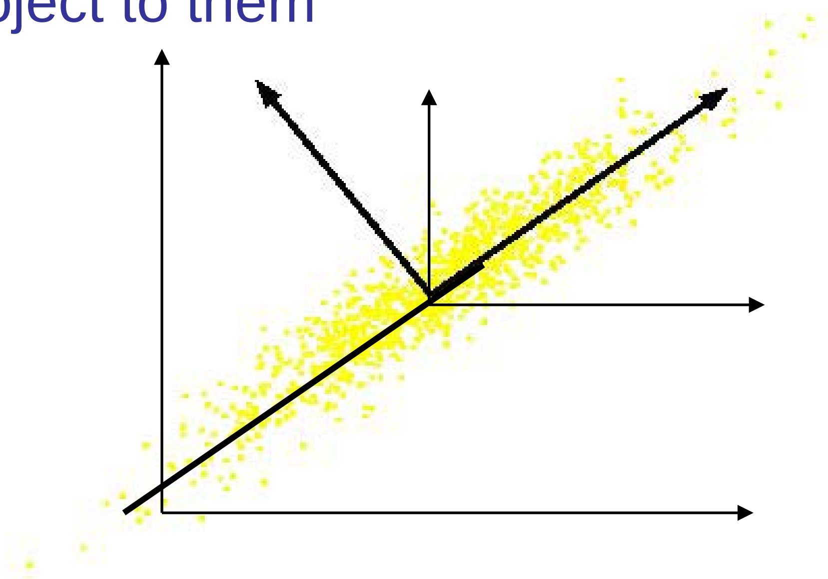
$$\mathbf{x}\mathbf{x}^T\mathbf{u} - \lambda\mathbf{u} = (\mathbf{x}\mathbf{x}^T - \lambda\mathbf{I})\,\mathbf{u} = 0$$

As $\mathbf{u} \neq \mathbf{0}$ then **u** must be an eigenvector of $\mathbf{X}\mathbf{X}^T$ with eigenvalue λ

- λ is the principal eigenvalue of the **correlation matrix** $\mathbf{C} = \mathbf{X}\mathbf{X}^T$
- The eigenvalue denotes the amount of variability captured along that dimension

Computing the Components

- Similarly for the next axis, etc.
- So, the new axes are the eigenvectors of the matrix of correlations of the original variables, which captures the similarities of the original variables based on how data samples project to them



- Geometrically: centering followed by rotation
  - Linear transformation

Eigenvalues & Eigenvectors

- For symmetric matrices, eigenvectors for distinct eigenvalues are **orthogonal**

$$Sv_{\{1,2\}} = \lambda_{\{1,2\}} v_{\{1,2\}}, \text{ and } \lambda_1 \neq \lambda_2 \Rightarrow v_1 \bullet v_2 = 0$$

- All eigenvalues of a real symmetric matrix are **real**.

$$\text{if } |S - \lambda I| = 0 \text{ and } \mathrm{S} = \mathrm{S}^{\mathrm{T}} \Rightarrow \lambda \in \Re$$

- All eigenvalues of a positive semidefinite matrix are **non-negative**

$$\forall w \in \Re^n, w^T S w \geq 0, \text{ then if } \mathrm{Sv} = \lambda \mathrm{v} \Rightarrow \lambda \geq 0$$

Eigen/diagonal Decomposition

- Let $\mathbf{S} \in \mathbb{R}^{m \times m}$ be a **square** matrix with ***m* linearly independent eigenvectors** (a “non-defective” matrix)

- **Theorem**: Exists an **eigen decomposition**

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$$

diagonal

Unique for distinct eigen-values

(cf. matrix diagonalization theorem)

- Columns of ***U*** are **eigenvectors** of ***S***
- Diagonal elements of $\mathbf{\Lambda}$ are **eigenvalues** of $\mathbf{S}$

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}$$

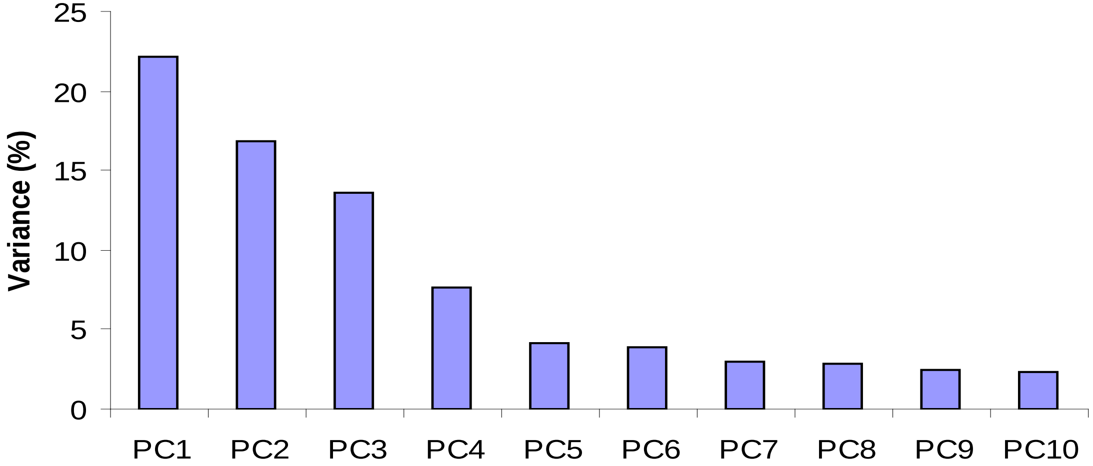
PCs, Variance and Least-Squares

- The first PC retains the greatest amount of variation in the sample
- The k^{th} PC retains the kth greatest fraction of the variation in the sample
- The k^{th} largest eigenvalue of the correlation matrix C is the variance in the sample along the k^{th} PC
- The least-squares view: PCs are a series of linear least squares fits to a sample, each orthogonal to all previous ones

How Many PCs?

- For n original dimensions, sample covariance matrix is nxn, and has up to n eigenvectors. So n PCs.
- Where does dimensionality reduction come from?

  Can *ignore* the components of lesser significance.



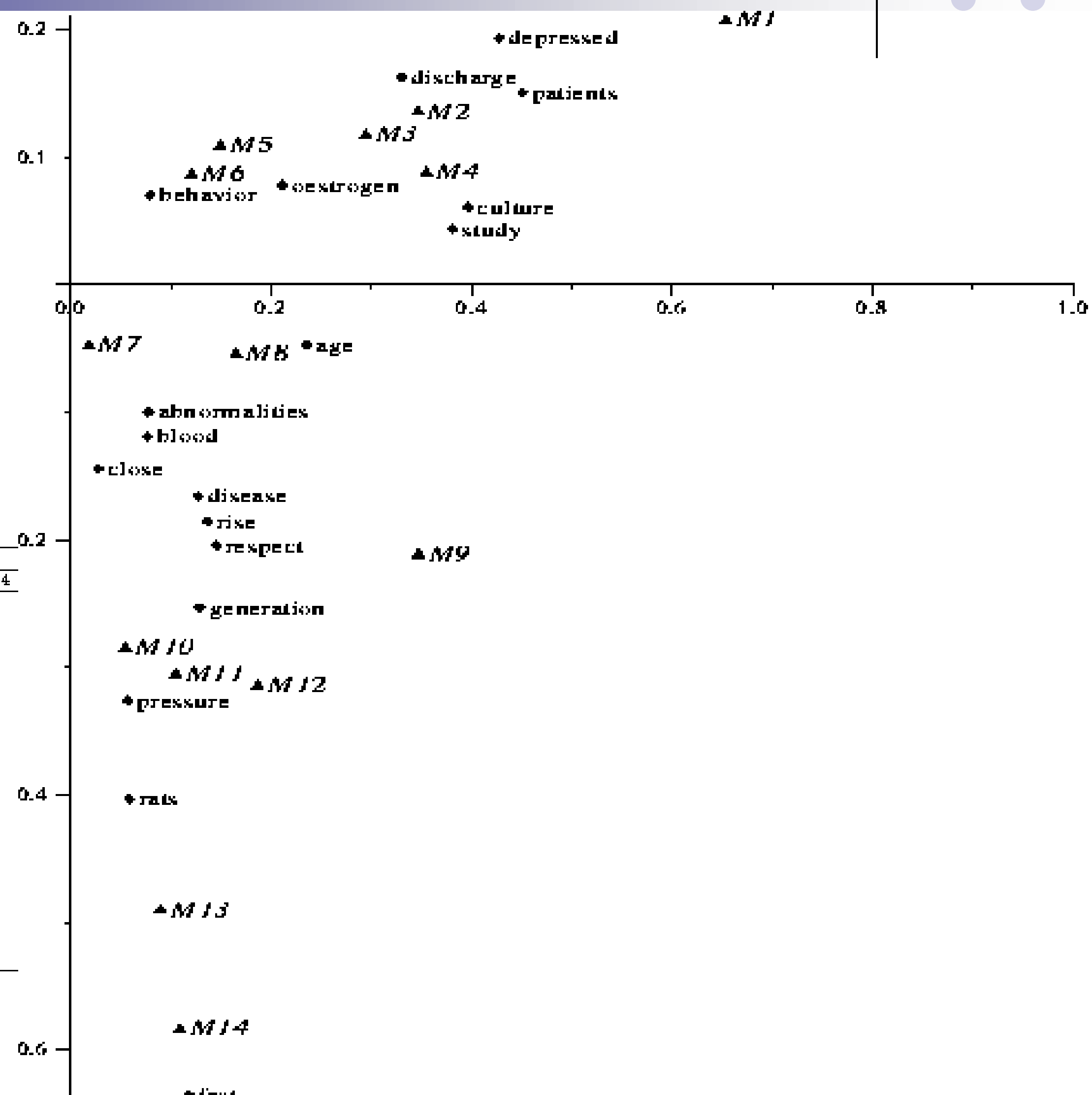
You do lose some information, but if the eigenvalues are small, you don't lose much

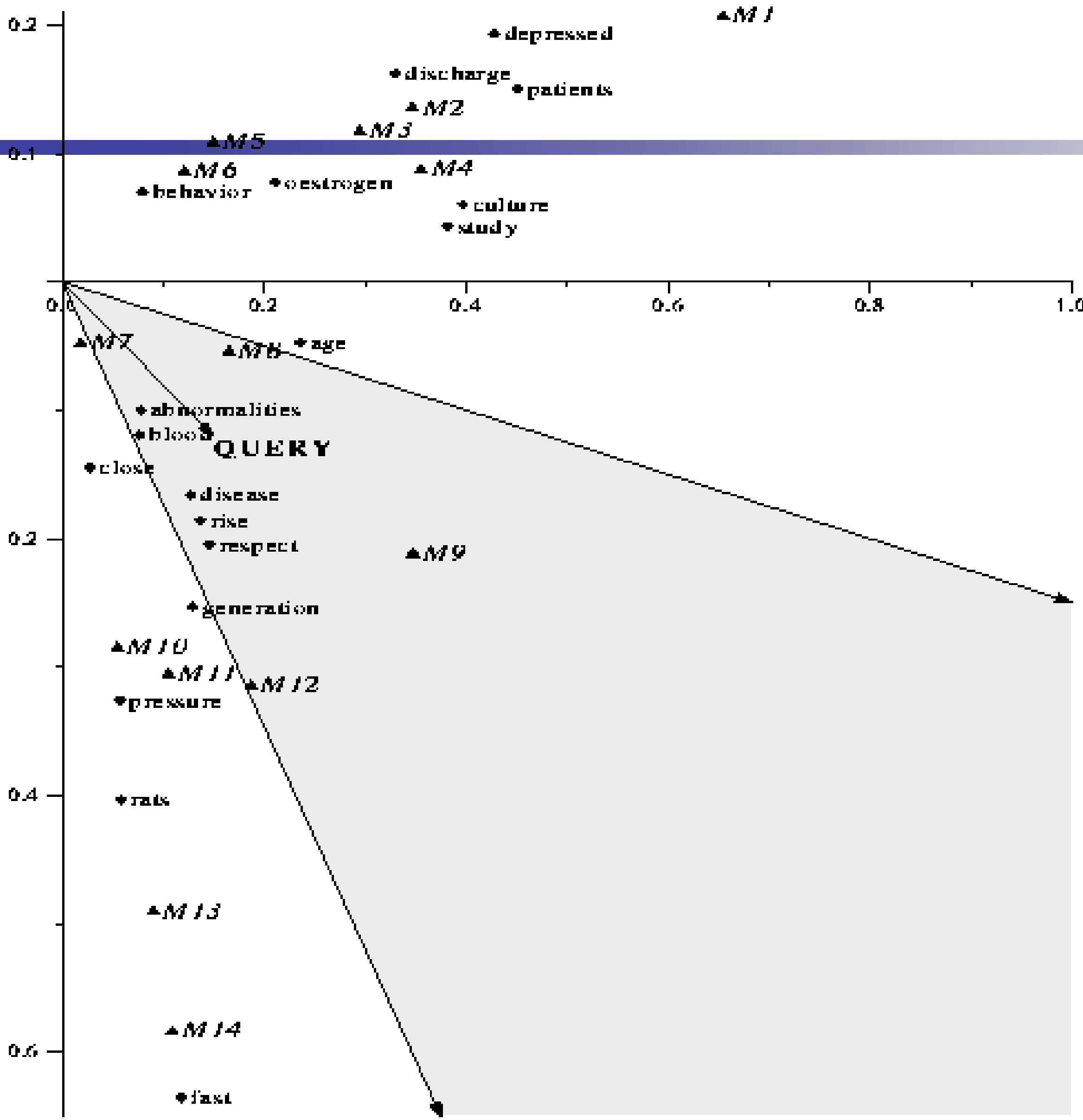
- n dimensions in original data
- calculate n eigenvectors and eigenvalues
- choose only the first p eigenvectors, based on their eigenvalues
- final data set has only p dimensions

Application: querying text doc.

Label	Medical Topic
M1	study of depressed patients after discharge with regard to age of onset and culture
M2	culture of pleuropneumonia like organisms found in vaginal discharge of patients
M3	study showed oestrogen production is depressed by ovarian irradiation
M4	cortisone rapidly depressed the secondary rise in oestrogen output of patients
M5	boys tend to react to death anxiety by acting out behavior while girls tended to become depressed
M6	changes in children's behavior following hospitalization studied a week after discharge
M7	surgical technique to close ventricular septal defects
M8	chromosomal abnormalities in blood cultures and bone marrow from leukaemic patients
M9	study of christmas disease with respect to generation and culture
M10	insulin not responsible for metabolic abnormalities accompanying a prolonged fast
M11	close relationship between high blood pressure and vascular disease
M12	mouse kidneys show a decline with respect to age in the ability to concentrate the urine during a water fast
M13	fast cell generation in the eye lens epithelium of rats
M14	fast rise of cerebral oxygen pressure in rats

Terms	Documents													
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	1	0	1	1	1	0	0	0	0	0	0	0	0	0
discharge	1	1	0	0	0	1	0	0	0	0	0	0	0	0
disease	0	0	0	0	0	0	0	0	1	0	1	0	0	0
fast	0	0	0	0	0	0	0	0	0	1	0	1	1	1
generation	0	0	0	0	0	0	0	0	1	0	0	0	1	0
oestrogen	0	0	1	1	0	0	0	0	0	0	0	0	0	0
patients	1	1	0	1	0	0	0	1	0	0	0	0	0	0
pressure	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rats	0	0	0	0	0	0	0	0	0	0	0	0	1	1
respect	0	0	0	0	0	0	0	1	0	0	0	1	0	0
rise	0	0	0	1	0	0	0	0	0	0	0	0	0	1
study	1	0	1	0	0	0	0	0	1	0	0	0	0	0





$$\begin{pmatrix} 0.1491 & -0.1199 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^{T} \begin{pmatrix} 0.1623 & -0.1372 \\ 0.2068 & -0.0488 \\ 0.0597 & 0.0614 \\ 0.1663 & -0.1313 \\ 0.0258 & -0.1246 \\ 0.4534 & 0.0386 \\ 0.3579 & 0.1710 \\ 0.2931 & 0.1426 \\ 0.0690 & -0.1576 \\ 0.0940 & -0.6535 \\ 0.0599 & -0.2378 \\ 0.1560 & 0.0661 \\ 0.4948 & 0.1091 \\ 0.0460 & -0.3393 \\ 0.0369 & -0.4196 \\ 0.1797 & -0.1456 \\ 0.1087 & -0.2126 \\ 0.3814 & 0.0941 \end{pmatrix} \begin{pmatrix} 3.5919 & 0 \\ 0 & 2.6471 \end{pmatrix}^{-1}$$

Number of Factors					
$k = 2$		$k = 4$		$k = 8$	
M 9	1.00	M 8	0.92	M 8	0.67
M12	0.88	M 9	0.89	M12	0.55
M 8	0.85	M 2	0.64	M10	0.54
M11	0.82	M10	0.48		
M10	0.79	M12	0.46		
M 7	0.74	M11	0.40		
M14	0.72				
M13	0.71				
M 4	0.67				
M 1	0.56				
M 2	0.42				

Within .40 threshold

K is the number of singular values used

Summary:

- Principle
  - Linear projection method to reduce the number of parameters
  - Transfer a set of correlated variables into a new set of uncorrelated variables
  - Map the data into a space of lower dimensionality
  - Form of unsupervised learning
- Properties
  - It can be viewed as a rotation of the existing axes to new positions in the space defined by original variables
  - New axes are orthogonal and represent the directions with maximum variability
- Application: In many settings in pattern recognition and retrieval, we have a feature-object matrix.
  - For text, the terms are features and the docs are objects.
  - Could be opinions and users …
  - This matrix may be redundant in dimensionality.
  - Can work with low-rank approximation.
  - If entries are missing (e.g., users' opinions), can recover if dimensionality is low.

Going beyond

- What is the essence of the C matrix?

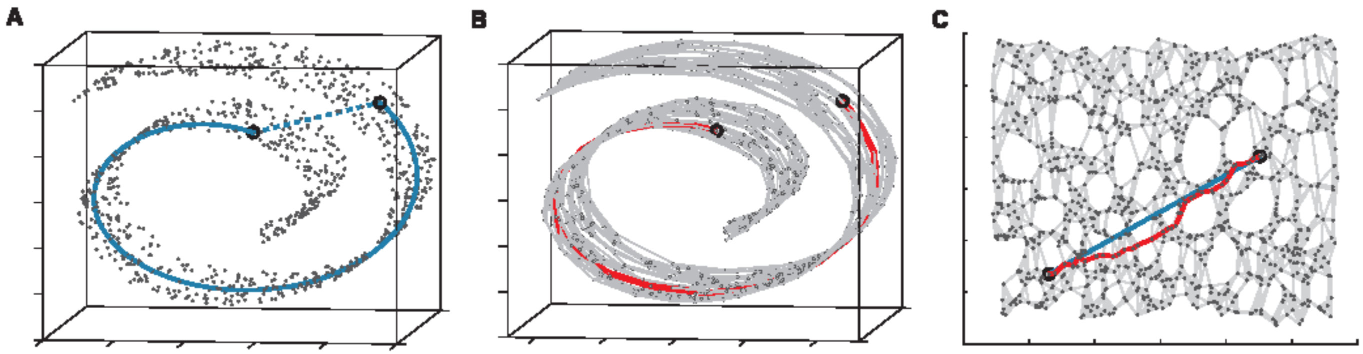
$$C = E[XX^T] = \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

- The elements in C captures some kind of affinity between a pair of data points in the semantic space

- We can replace it with any reasonable affinity measure
  - E.g., $D = \left(\|x_i - x_j\|^2\right)_{ij}$: distance matrix MDS
  - E.g., the geodistance ISOMAP

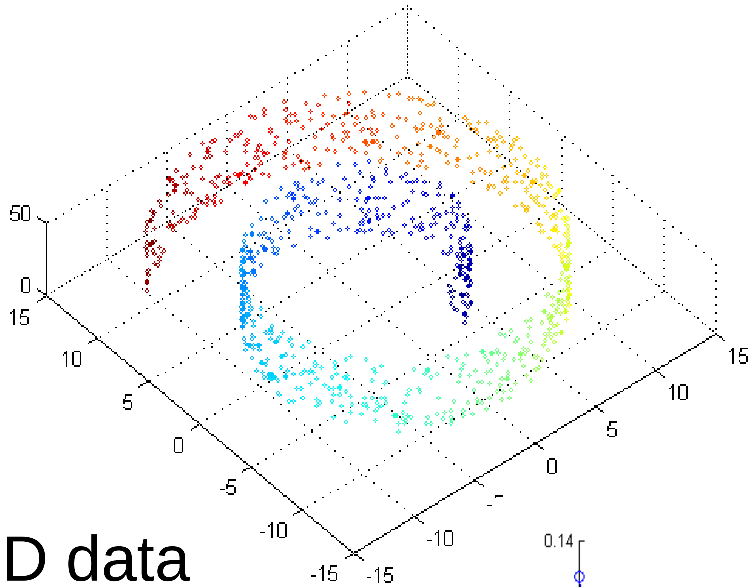
Nonlinear DR – Isomap

[Josh. Tenenbaum, Vin de Silva, John langford 2000]

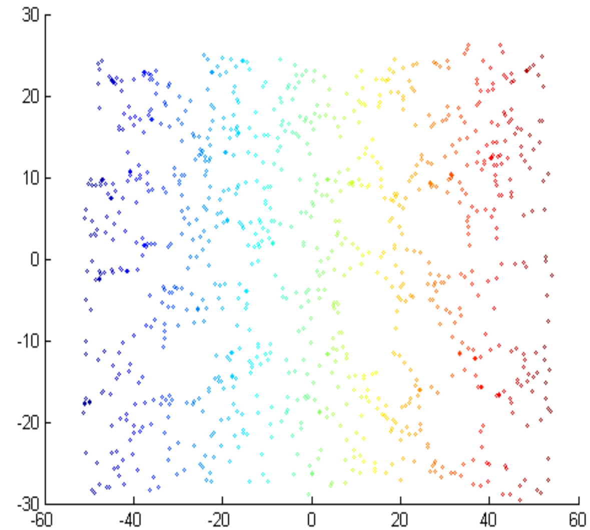


- Constructing neighbourhood graph G
- For each pair of points in G, Computing shortest path distances ---- geodesic distances.
  - Use Dijkstra's or Floyd's algorithm
- Apply kernel PCA for C given by the centred matrix of squared geodesic distances.
- Project test points onto principal components as in kernel PCA.

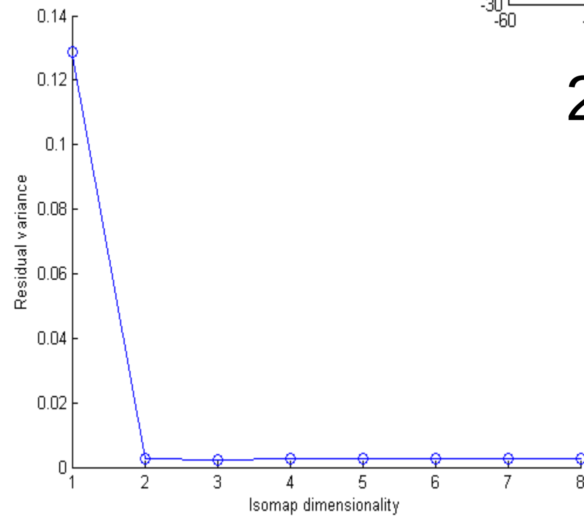
“Swiss Roll” dataset



3D data



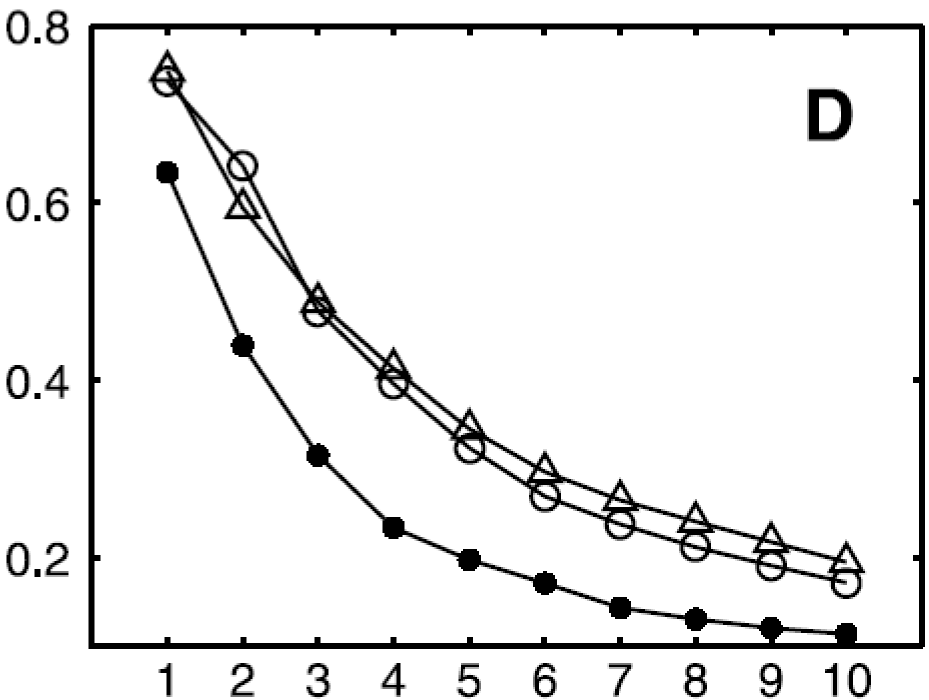
2D coord chart

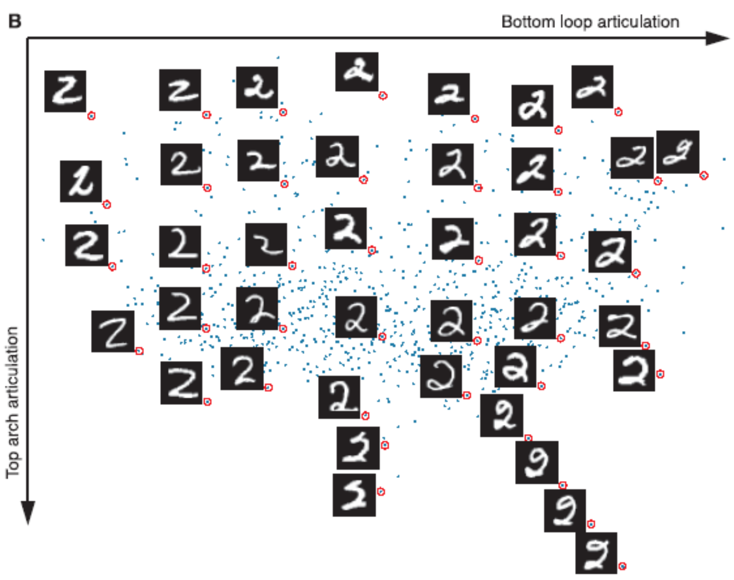


Error vs. dimensionality of coordinate chart

PCA, MD vs ISOMAP

- The residual variance of PCA (open triangles), MDS (open circles), and Isomap





ISOMAP algorithm Pros/Cons

Advantages:

- Nonlinear
- Globally optimal
- Guarantee asymptotically to recover the true dimensionality

Drawback:

- May not be stable, dependent on topology of data
- As N increases, pair wise distances provide better approximations to geodesics, but cost more computation

Local Linear Embedding (a.k.a LLE)

- LLE is based on simple geometric intuitions.

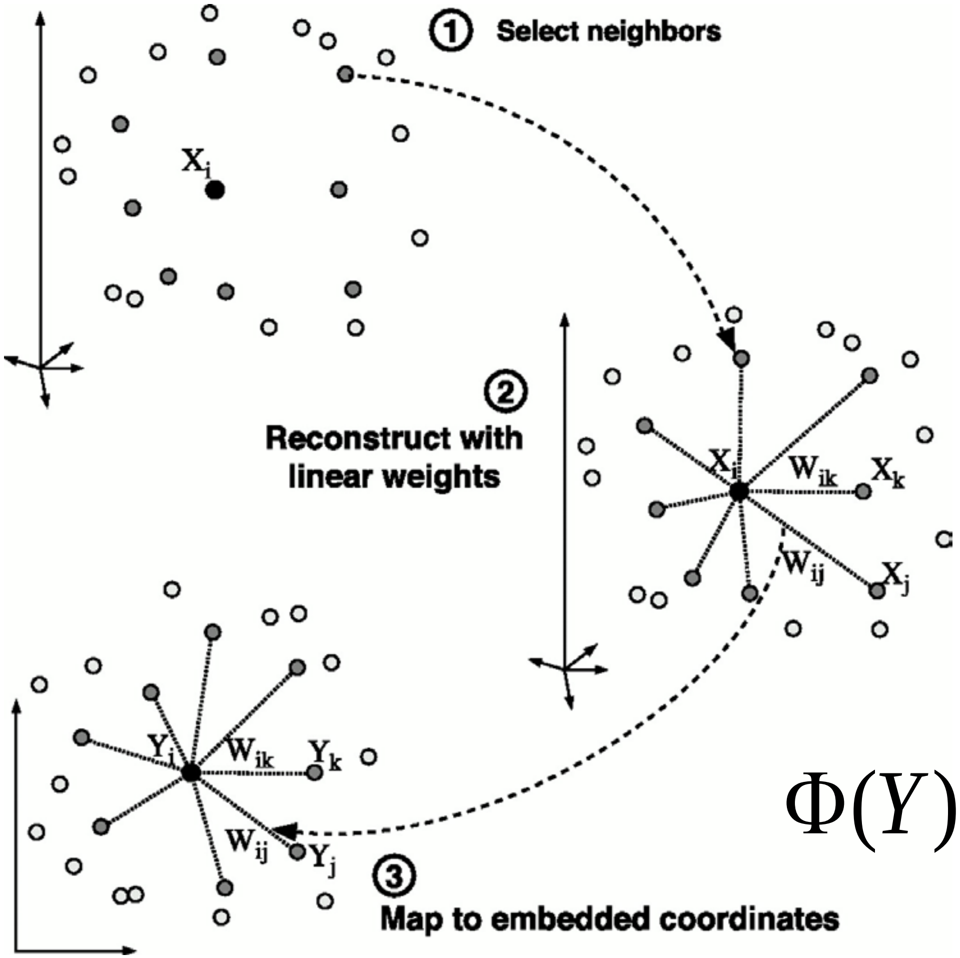
- Suppose the data consist of *N* real-valued vectors X_i, each of dimensionality *D.*

- Each data point and its neighbors expected to lie on or close to a locally linear patch of the manifold.

Steps in LLE algorithm

- Assign neighbors to each data point $\vec{X}_i$
- Compute the weights W_{ij} that best linearly reconstruct the data point from its neighbors, solving the constrained least-squares problem.
- Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}.

Fit locally, Think Globally



From Nonlinear Dimensionality Reduction by Locally Linear Embedding

Sam T. Roweis and Lawrence K. Saul

$$\Phi(Y) = \sum_i |\vec{Y} - \sum_j W_{ij}\vec{Y}_j|^2$$

Super-Resolution Through Neighbor Embedding [Yeung et al CVPR 2004]

Training Xs^i

img1.jpg

Training Ys^i

Testing Xt

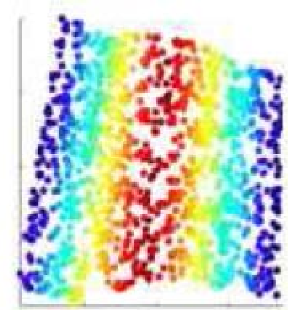
?

Testing Yt

Intuition

- Patches of the image lie on a manifold

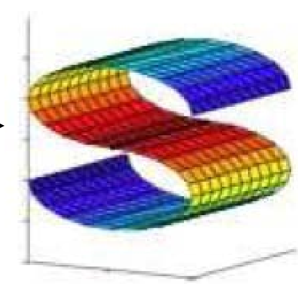
Training Xs^i



Low dimensional Manifold

img1.jpg

Training Ys^i



High dimensional Manifold

Algorithm

1. Get feature vectors for each low resolution training patch.
2. For each test patch feature vector find K nearest neighboring feature vectors of training patches.
3. Find optimum weights to express each test patch vector as a weighted sum of its K nearest neighbor vectors.
4. Use these weights for reconstruction of that test patch in high resolution.

Results

img1.jpg

Training Xs^i

Training Ys^i

Testing Xt

Testing Yt

Summary:

- Principle
  - Linear and nonlinear projection method to reduce the number of parameters
  - Transfer a set of correlated variables into a new set of uncorrelated variables
  - Map the data into a space of lower dimensionality
  - Form of unsupervised learning
- Applications
  - PCA and Latent semantic indexing for text mining
  - Isomap and Nonparametric Models of Image Deformation
  - LLE and Isomap Analysis of Spectra and Colour Images
  - Image Spaces and Video Trajectories: Using Isomap to Explore Video Sequences
  - Mining the structural knowledge of high-dimensional medical data using isomap

Isomap Webpage: http://isomap.stanford.edu/

Applying PCA and LDA: Eigen-faces and Fisher-faces

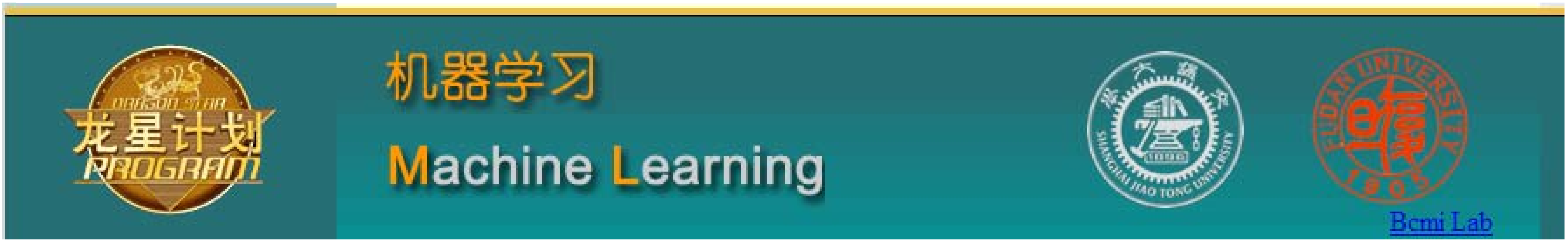
L. Fei-Fei
Computer Science Dept.
Stanford University

Machine learning in computer vision

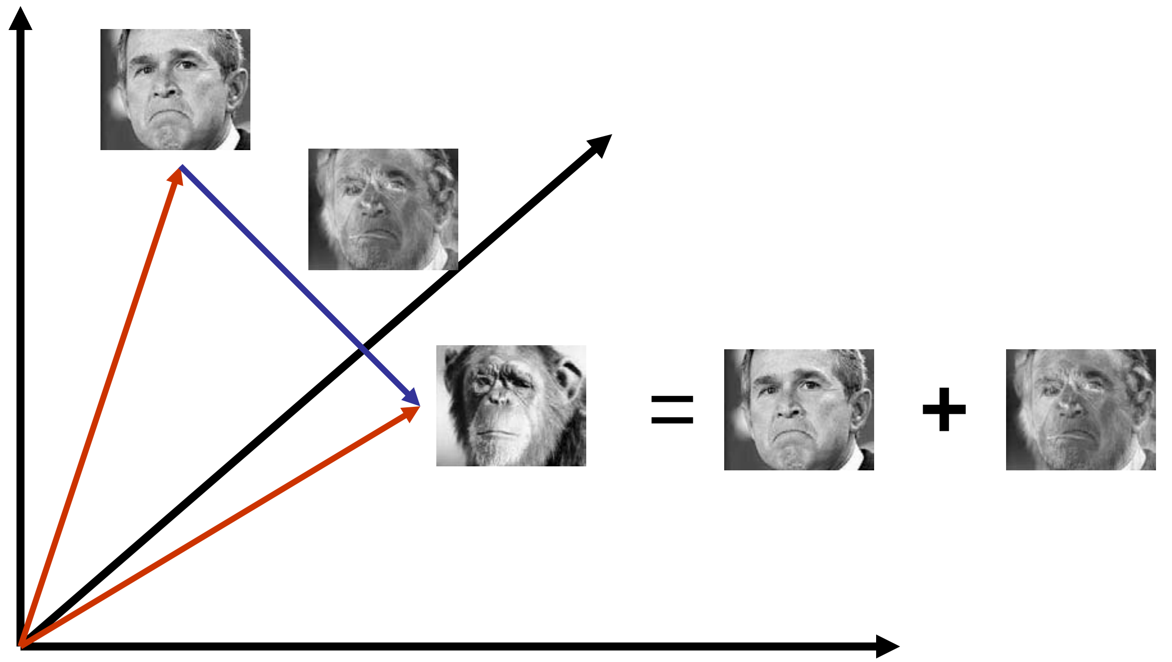
- Aug 13, Lecture 7: Dimensionality reduction, Manifold learning
  - Eigen- and Fisher- faces
  - Applications to object representation

References:

1. Turk and Penland, Eigenfaces for Recognition, 1991
2. Belhumeur, Hespanha and Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection



The Space of Faces



- An image is a point in a high dimensional space
  - An N x M image is a point in R^{NM}
  - We can define vectors in this space as we did in the 2D case

[Thanks to Chuck Dyer, Steve Seitz, Nishino]

Key Idea

- Images in the possible set $\chi = \{\hat{x}_{RL}^{P}\}$ are highly correlated.

- So, compress them to a low-dimensional subspace that captures key appearance characteristics of the visual DOFs.

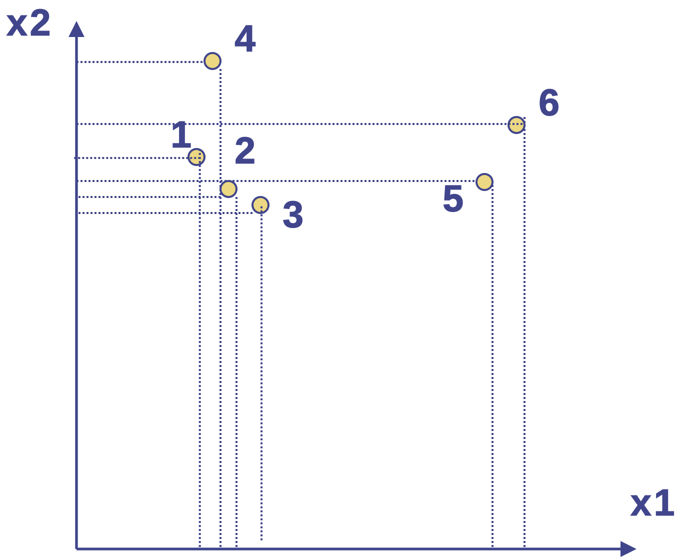
- EIGENFACES: [Turk and Pentland]

USE PCA!

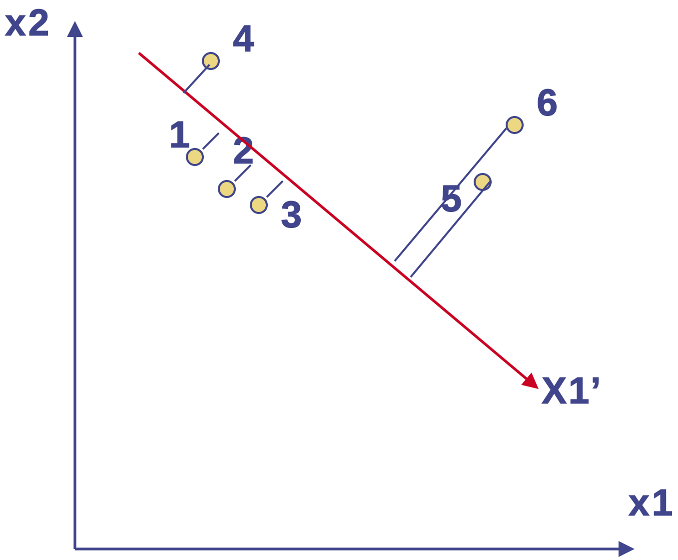
Principal Component Analysis (PCA)

- PCA is used to determine the most representing features among data points.
  - It computes the p-dimensional subspace such that the projection of the data points onto the subspace has the largest variance among all p-dimensional subspaces.

Illustration of PCA

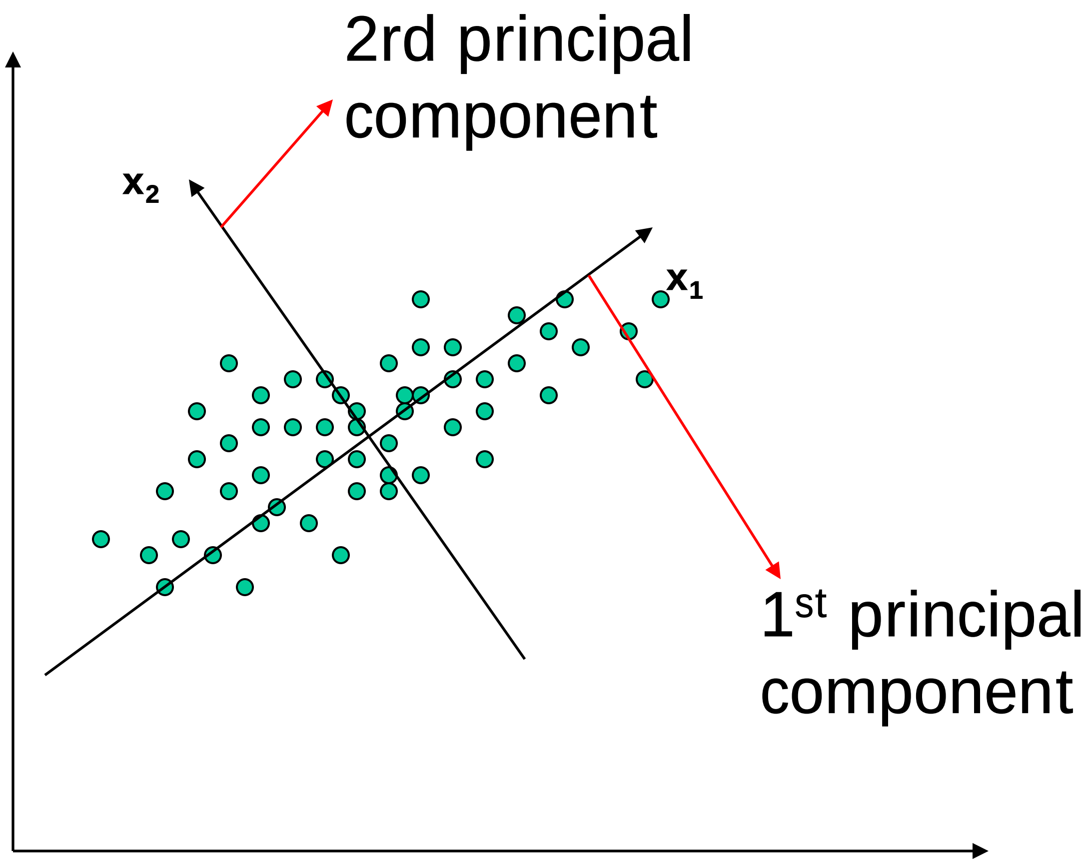


One projection



PCA projection

Illustration of PCA



Mathematical Formulation

Find a transformation, W,

$$\mathbf{y}_k = W^T \mathbf{x}_k \qquad k = 1, 2, \dots, N$$

m-dimensional | **Orthonormal** $W \in \mathbb{R}^{n \times m}$ | **n-dimensional**

Total scatter matrix: $$S_T = \sum_{k=1}^{N} (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T$$

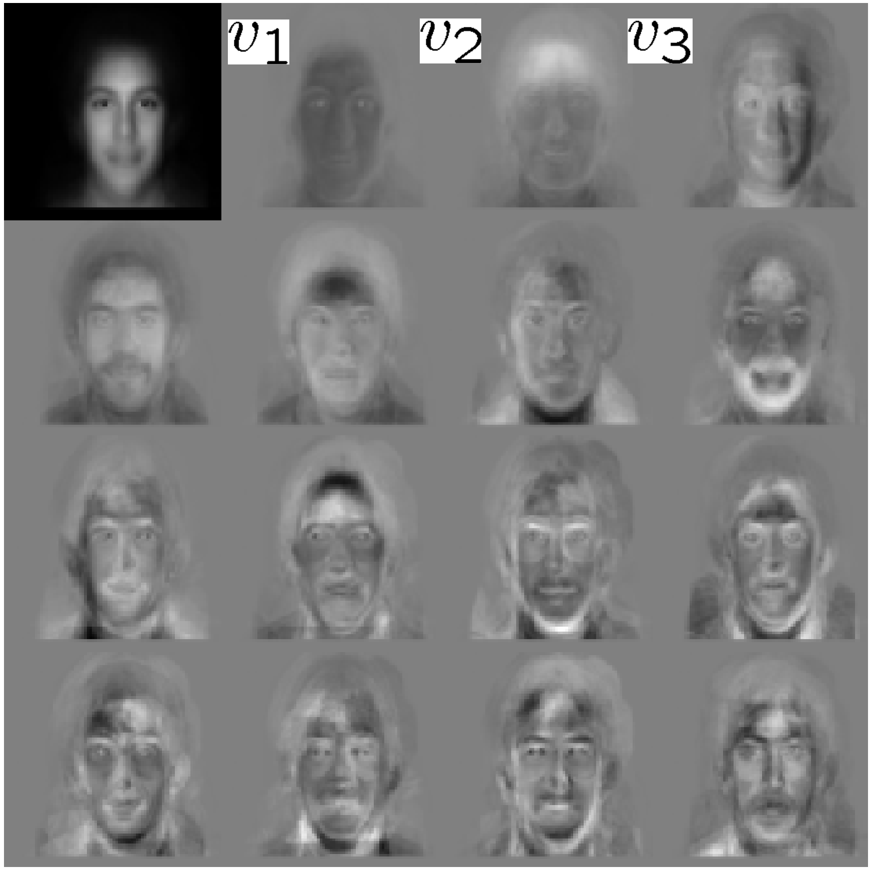
$$W_{opt} = \arg\max_W \left| W^T S_T W \right| = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_m \end{bmatrix}$$

W_{opt} corresponds to m eigen-vectors of S_T

Eigenfaces

- PCA extracts the eigenvectors of **A**
  - Gives a set of vectors $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, ...
  - Each one of these vectors is a direction in face space
    - what do these look like?

$\overline{x}$



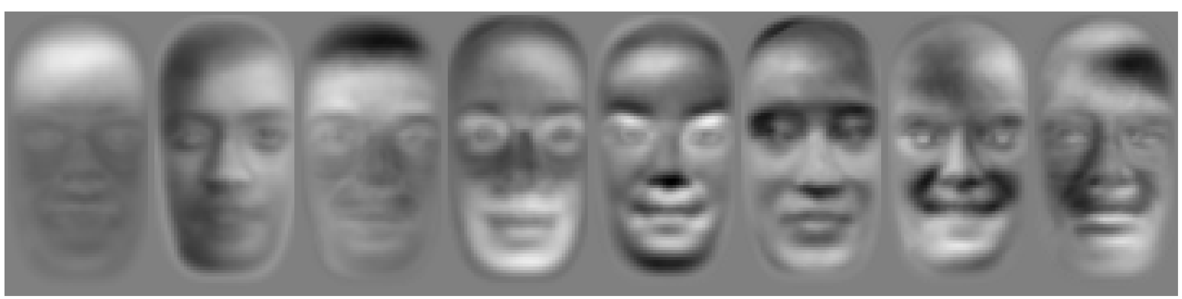
Projecting onto the Eigenfaces

- The eigenfaces $\mathbf{v_1}, \ldots, \mathbf{v_K}$ span the space of faces
  - A face is converted to eigenface coordinates by

$$\mathbf{x} \rightarrow (\underbrace{(\mathbf{x} - \overline{\mathbf{x}}) \cdot \mathbf{v}_1}_{a_1}, \ \underbrace{(\mathbf{x} - \overline{\mathbf{x}}) \cdot \mathbf{v}_2}_{a_2}, \ldots, \ \underbrace{(\mathbf{x} - \overline{\mathbf{x}}) \cdot \mathbf{v}_\mathbf{K}}_{a_K})$$

$$\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v}_1 + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$$

$\mathbf{x}$



$a_1\mathbf{v}_1$ $a_2\mathbf{v}_2$ $a_3\mathbf{v}_3$ $a_4\mathbf{v}_4$ $a_5\mathbf{v}_5$ $a_6\mathbf{v}_6$ $a_7\mathbf{v}_7$ $a_8\mathbf{v}_8$

Algorithm

Training

1. Align training images $x_1, x_2, \ldots, x_N$

Note that each image is formulated into a long vector!

2. Compute average face $u = 1/N \Sigma x_i$

3. Compute the difference image $\varphi_i = x_i - u$

Algorithm

4. Compute the covariance matrix (total scatter matrix)

$$S_T = 1/N\Sigma\ \varphi_i\ \varphi_i^T = BB^T,\ B=[\varphi_1, \varphi_2 \ldots \varphi_N]$$

5. Compute the eigenvectors of the covariance matrix , W

Testing

1. Projection in Eigenface

$$\text{Projection } \omega_i = W\,(X - u),\ W = \{\text{eigenfaces}\}$$

2. Compare projections

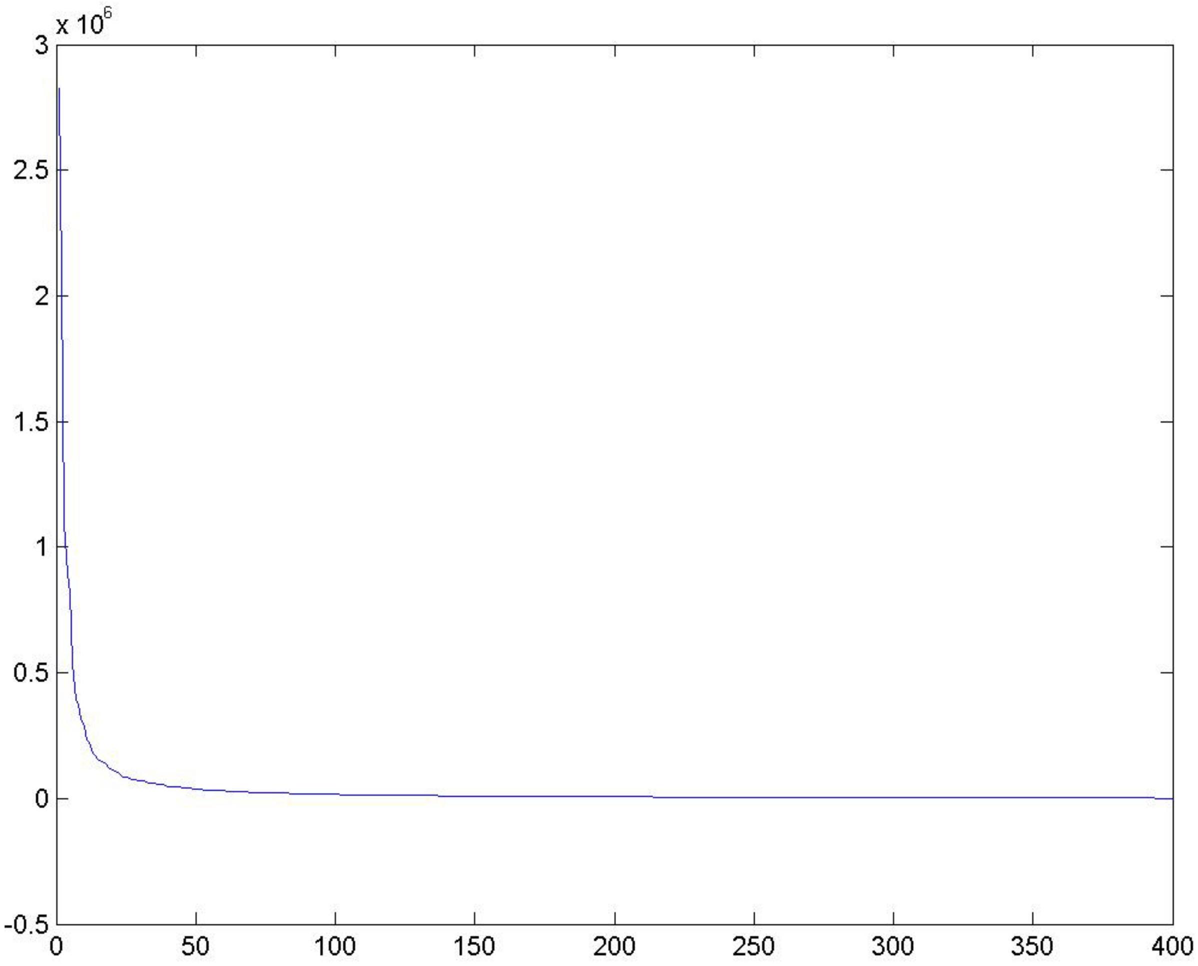
Illustration of Eigenfaces

- The visualization of eigenvectors:

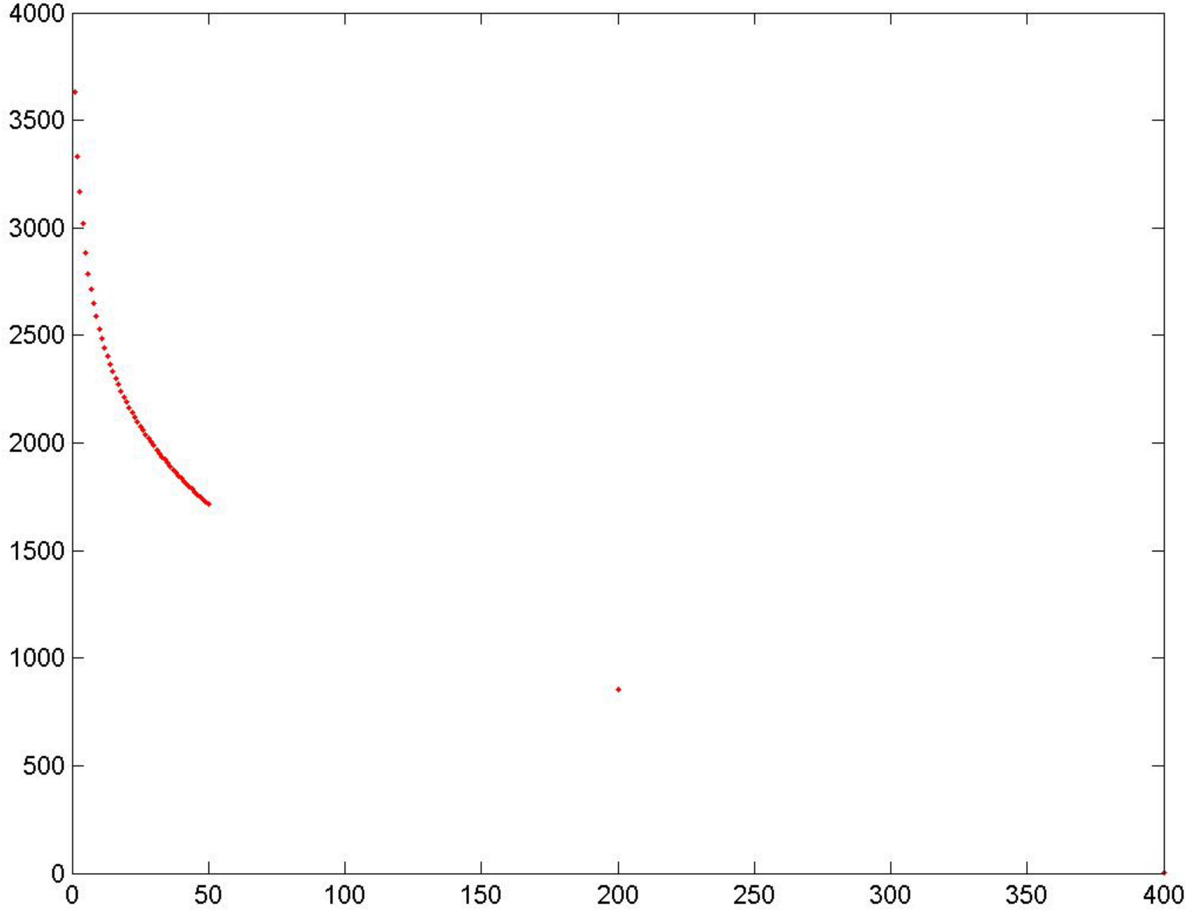
These are the first 4 eigenvectors from a training set of 400 images (ORL Face Database). They look like faces, hence called Eigenface.

Eigenfaces look somewhat like generic faces.

Eigenvalues



Reconstruction and Errors



imensionality.

ıd hence less

Summary for PCA and Eigenface

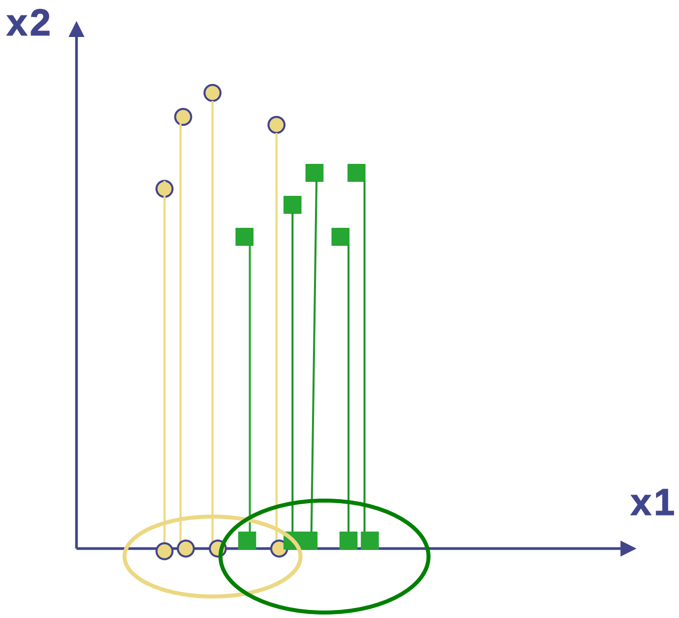
- Non-iterative, globally optimal solution
- PCA projection is **optimal for reconstruction** from a low dimensional basis, but **may NOT be optimal for discrimination…**

Linear Discriminant Analysis (LDA)

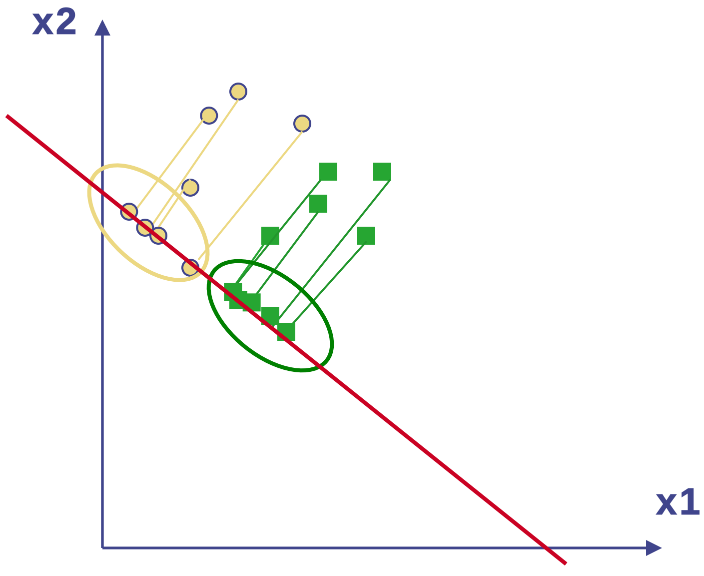
- Using Linear Discriminant Analysis (LDA) or Fisher's Linear Discriminant (FLD)
- Eigenfaces attempt to maximise the scatter of the training images in face space, while Fisherfaces attempt to maximise the **between class scatter**, while minimising the **within class scatter**.

Illustration of the Projection

- Using two classes as example:

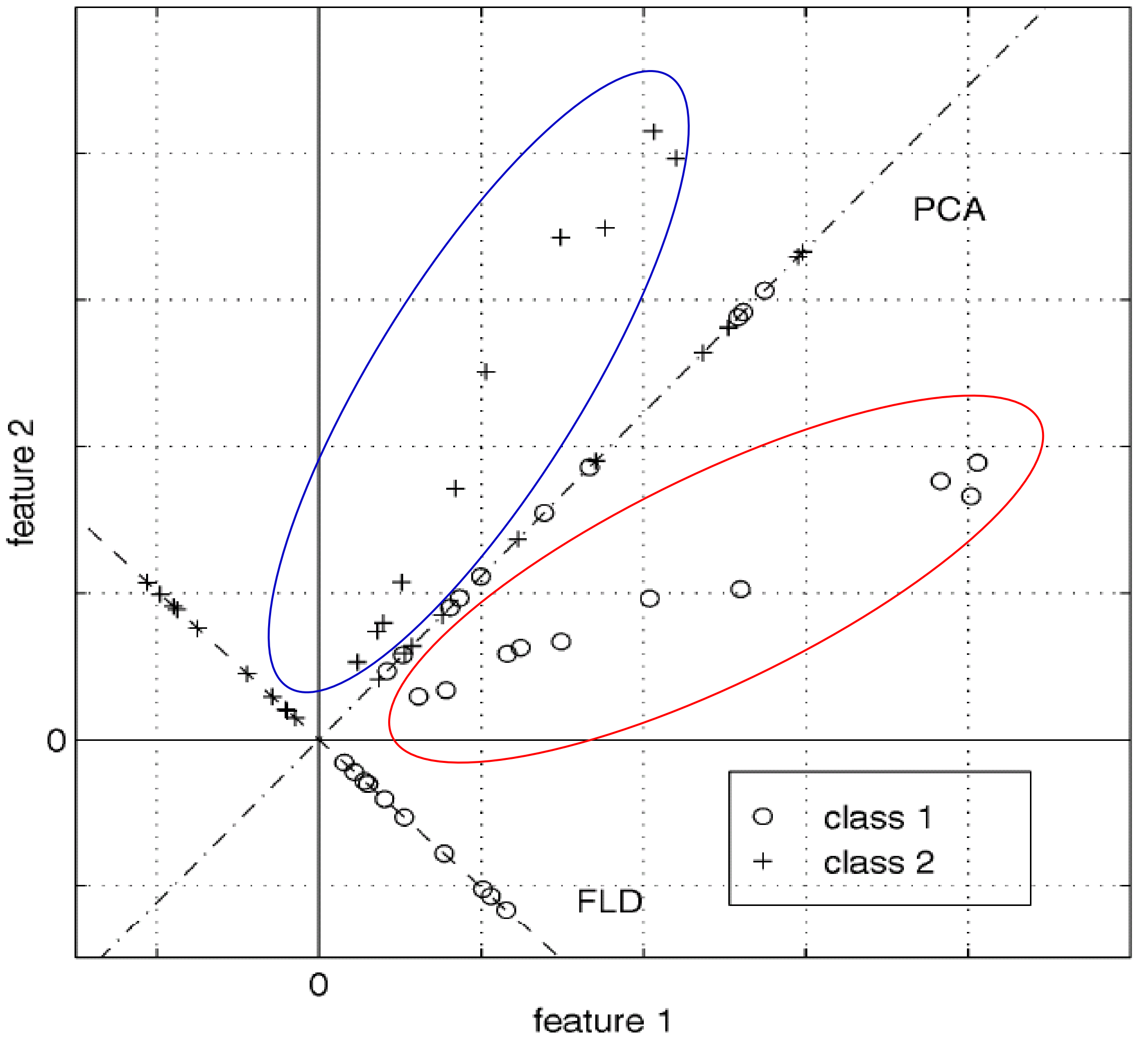


Poor Projection



Good Projection

Comparing with PCA



Variables

- N Sample images: $\{x_1, \cdots, x_N\}$
- c classes: $\{\chi_1, \cdots, \chi_c\}$

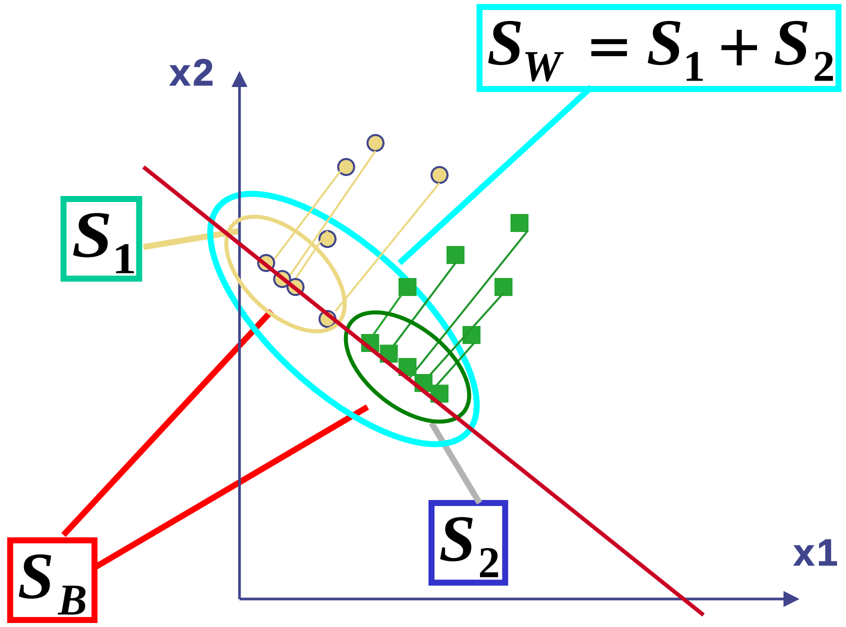
- Average of each class: $$\mu_i = \frac{1}{N_i} \sum_{x_k \in \chi_i} x_k$$

- Total average: $$\mu = \frac{1}{N} \sum_{k=1}^{N} x_k$$

Scatters

- Scatter of class i: $\boxed{S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T}$
- Within class scatter: $\boxed{S_W = \sum_{i=1}^{c} S_i}$
- Between class scatter: $\boxed{S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T}$
- Total scatter: $\boxed{S_T = S_W + S_B}$

Illustration



Mathematical Formulation (1)

- After projection: $\boxed{y_k = W^T x_k}$

- Between class scatter (of y's): $\boxed{\tilde{S}_B = W^T S_B W}$
- Within class scatter (of y's): $\boxed{\tilde{S}_W = W^T S_W W}$

Mathematical Formulation (2)

- The desired projection:

$$\boxed{W_{opt} = \arg\max_{W} \frac{\left|\tilde{S}_B\right|}{\left|\tilde{S}_W\right|} = \arg\max_{W} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}}$$

- How is it found ? → Generalized Eigenvectors

$$\boxed{S_B w_i = \lambda_i S_W w_i \qquad i = 1, \ldots, m}$$

- Data dimension is much larger than the number of samples $n >> N$
- The matrix S_W is singular: $\boxed{Rank(S_W) \leq N - c}$

Fisherface (PCA+FLD)

- Project with PCA to $N - c$ space $\boxed{z_k = W_{pca}^{\ T} x_k}$

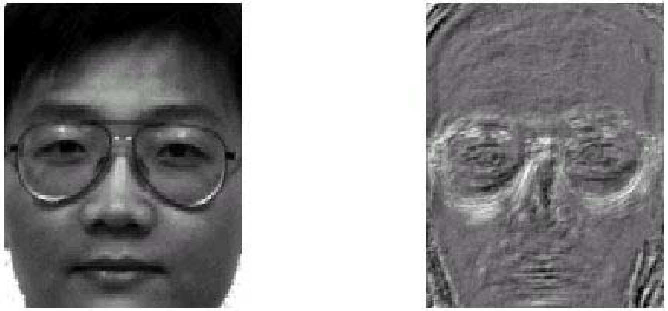
$$\boxed{W_{pca} = \arg\max_{W} \left| W^T S_T W \right|}$$

- Project with FLD to $c - 1$ space $\boxed{y_k = W_{fld}^{\ T} z_k}$

$$\boxed{W_{fld} = \arg\max_{W} \frac{\left| W^T W_{pca}^T S_B W_{pca} W \right|}{\left| W^T W_{pca}^T S_W W_{pca} W \right|}}$$

Illustration of FisherFace

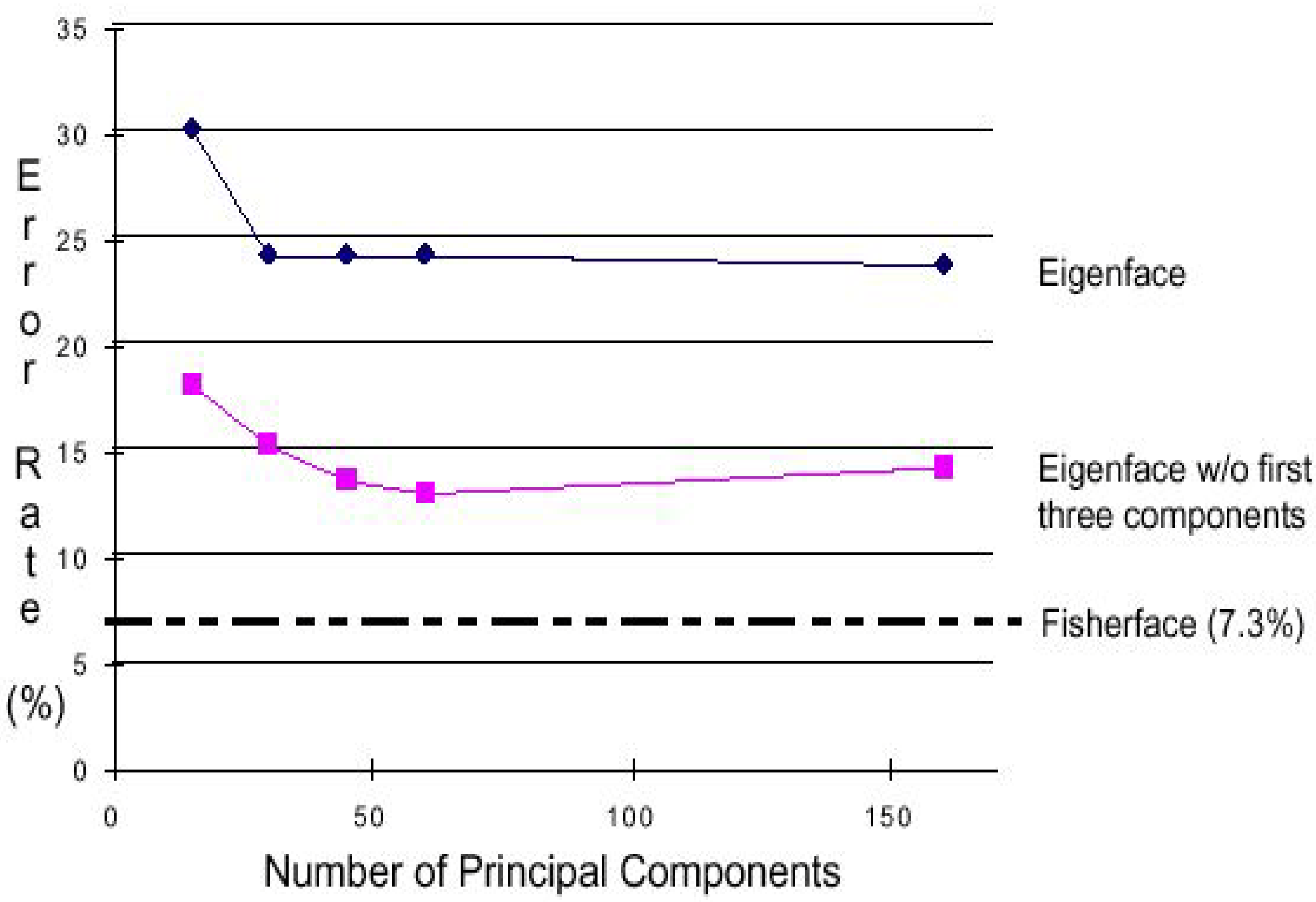
- Fisherface



Results: Eigenface vs. Fisherface (1)

- Input: 160 images of 16 people
- Train: 159 images
- Test: 1 image
- Variation in Facial Expression, Eyewear, and Lighting

Eigenface vs. Fisherface (2)



discussion

- Removing the first three principal components results in better performance under variable lighting conditions
- The Firsherface methods had error rates lower than the Eigenface method for the small datasets tested.

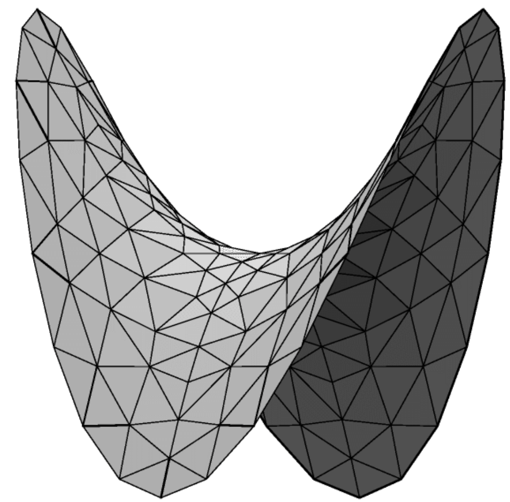
Manifold Learning for Object Representation

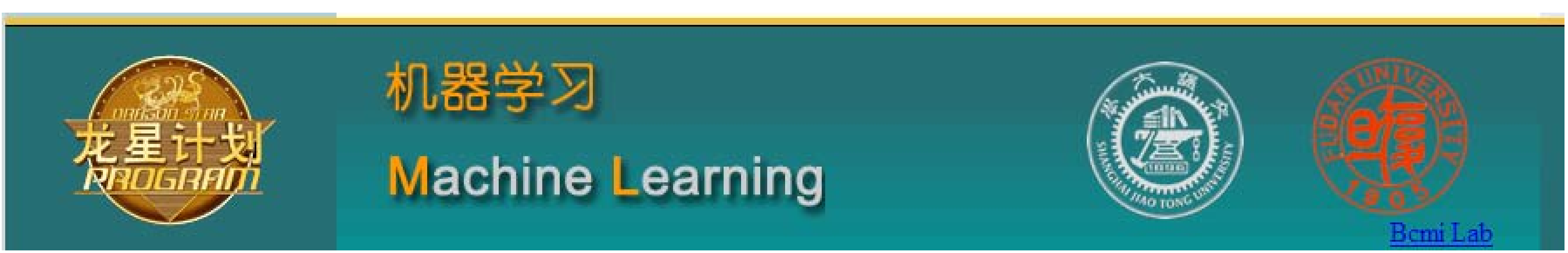
L. Fei-Fei
Computer Science Dept.
Stanford University

Machine learning in computer vision

- Aug 13, Lecture 7: Dimensionality reduction, Manifold learning
  - Eigen- and Fisher- faces
  - Applications to object representation

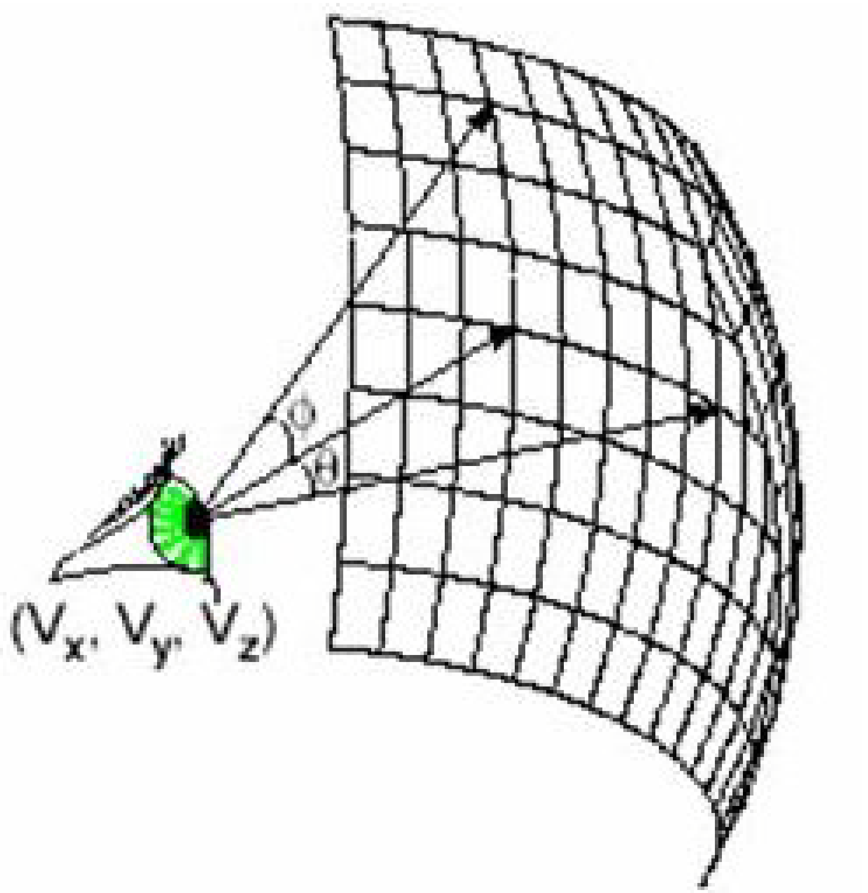
  (slides courtesy to David Thompson)





manifolds in vision

plenoptic function



manifolds in vision

appearance variation

images from hormel corp.

manifolds in vision

deformation

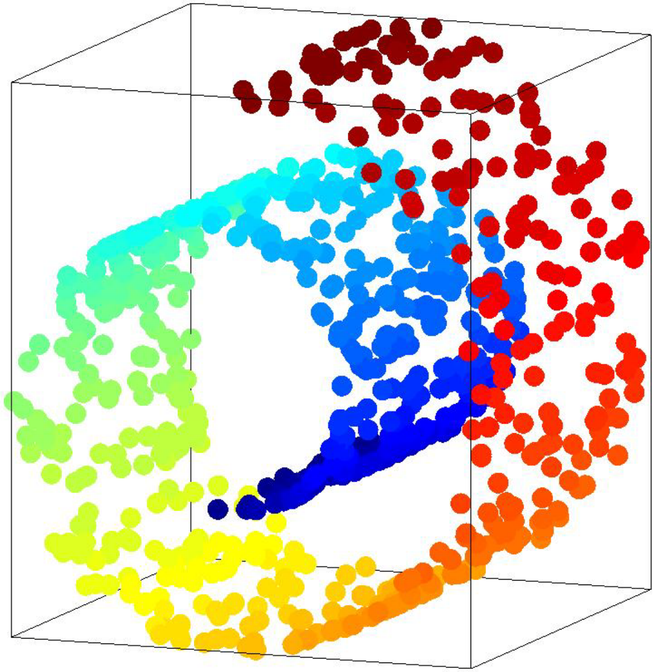
images from www.golfswingphotos.com

manifold learning

Find a low-D basis for describing high-D data.

X ~= X' S.T.
dim(X') << dim(X)

uncovers the intrinsic dimensionality



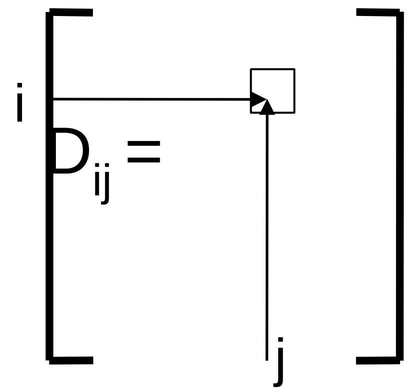
If we knew all pairwise distances…

	Chicago	Raleigh	Boston	Seattle	S.F.	Austin	Orlando
Chicago	0						
Raleigh	641	0					
Boston	851	608	0				
Seattle	1733	2363	2488	0			
S.F.	1855	2406	2696	684	0		
Austin	972	1167	1691	1764	1495	0	
Orlando	994	520	1105	2565	2458	1015	0

Distances calculated with geobytes.com/CityDistanceTool

Multidimensional Scaling (MDS)

For *n* data points, and a distance matrix D,

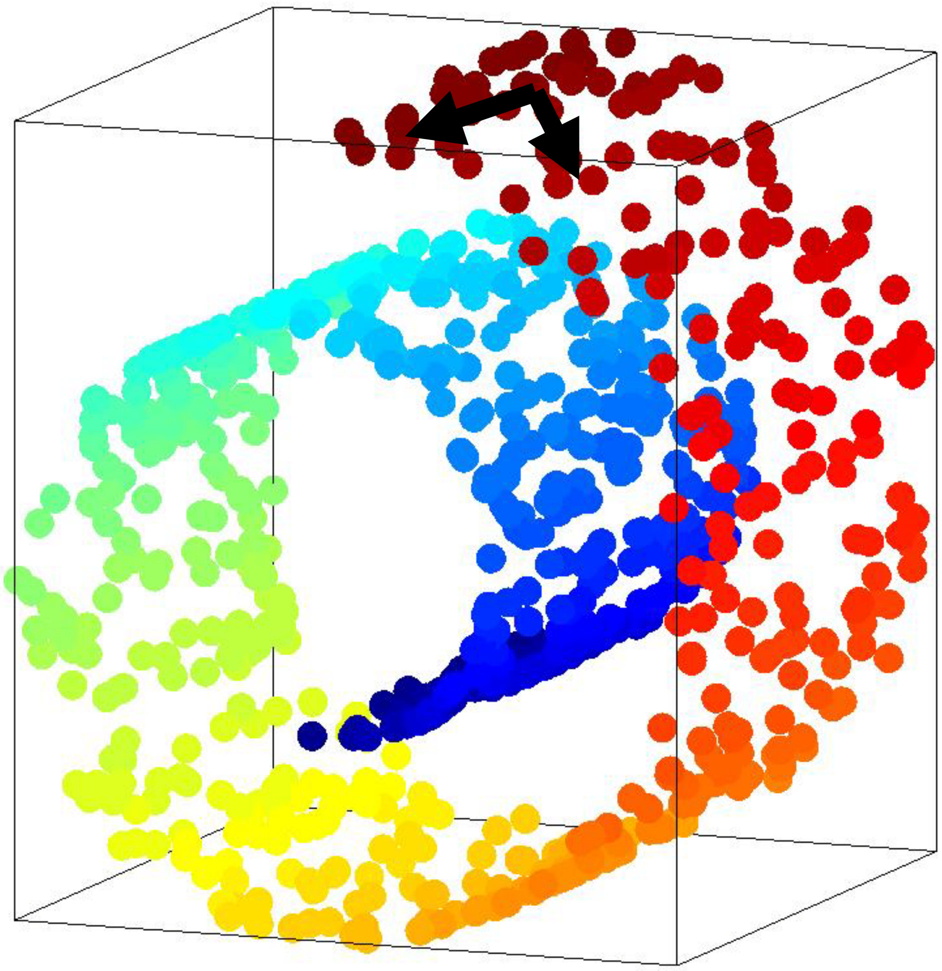


...we can construct a *m*-dimensional space to preserve inter-point distances by using the top eigenvectors of D scaled by their eigenvalues

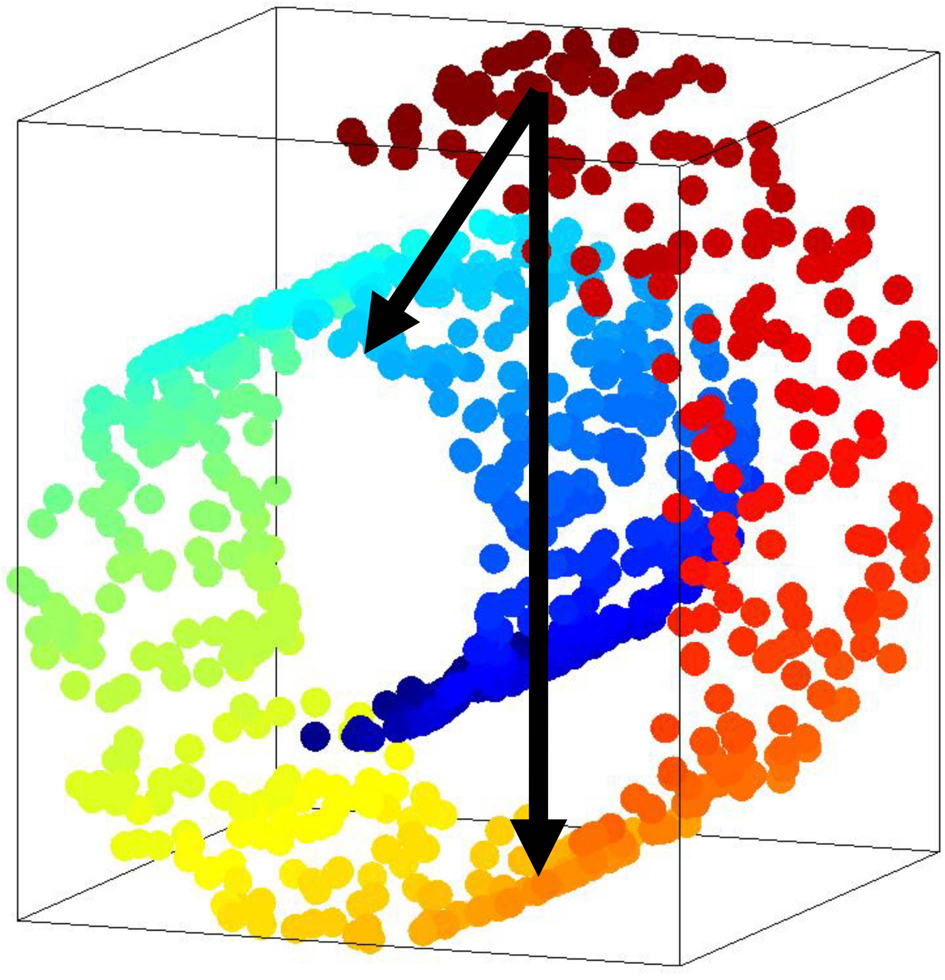
MDS result in 2D

Actual plot of cities

Don’t know distances



Don’t know distnaces



why do manifold learning?

1. data compression

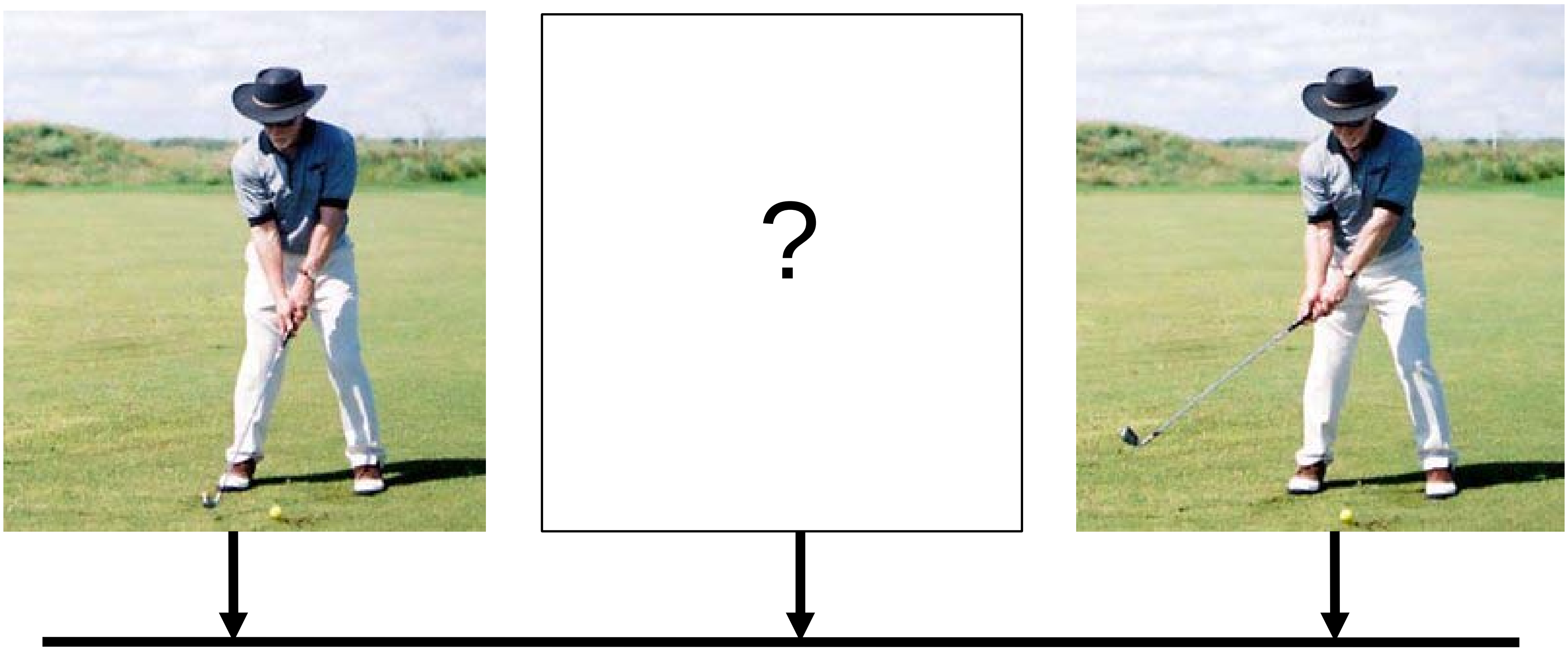
2. “curse of dimensionality”

3. de-noising

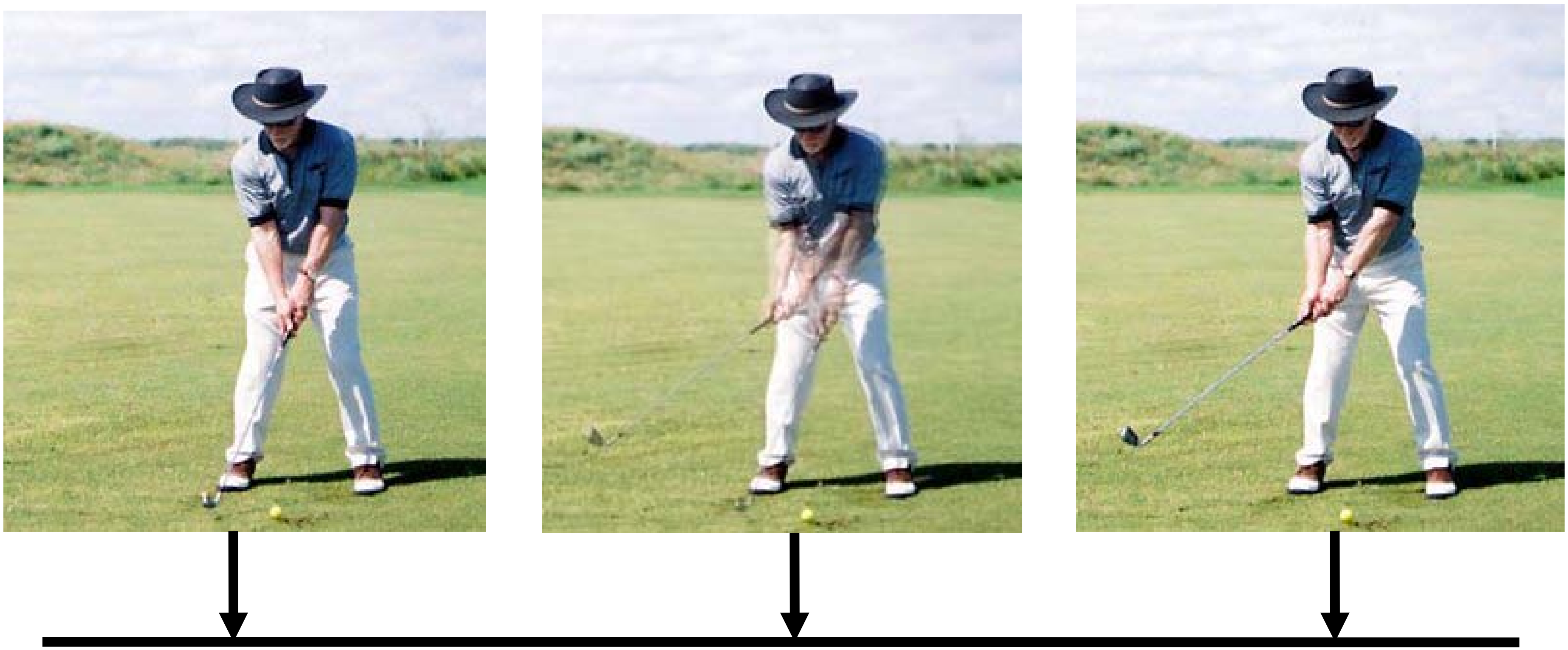
4. visualization

5. reasonable distance metrics

reasonable distance metrics

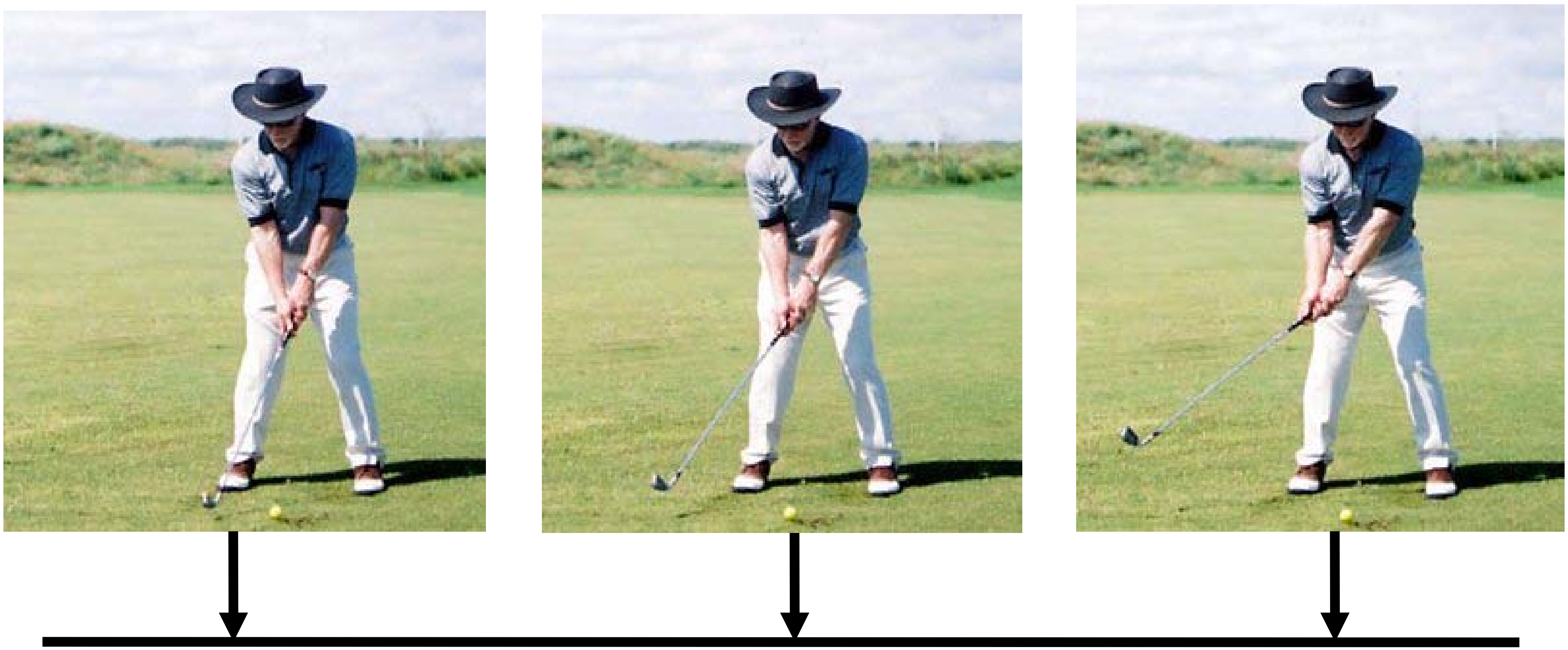


reasonable distance metrics



linear interpolation

reasonable distance metrics



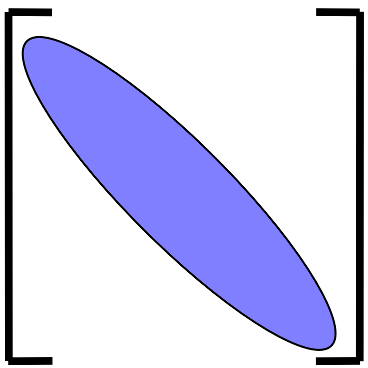
manifold interpolation

Isomap for images

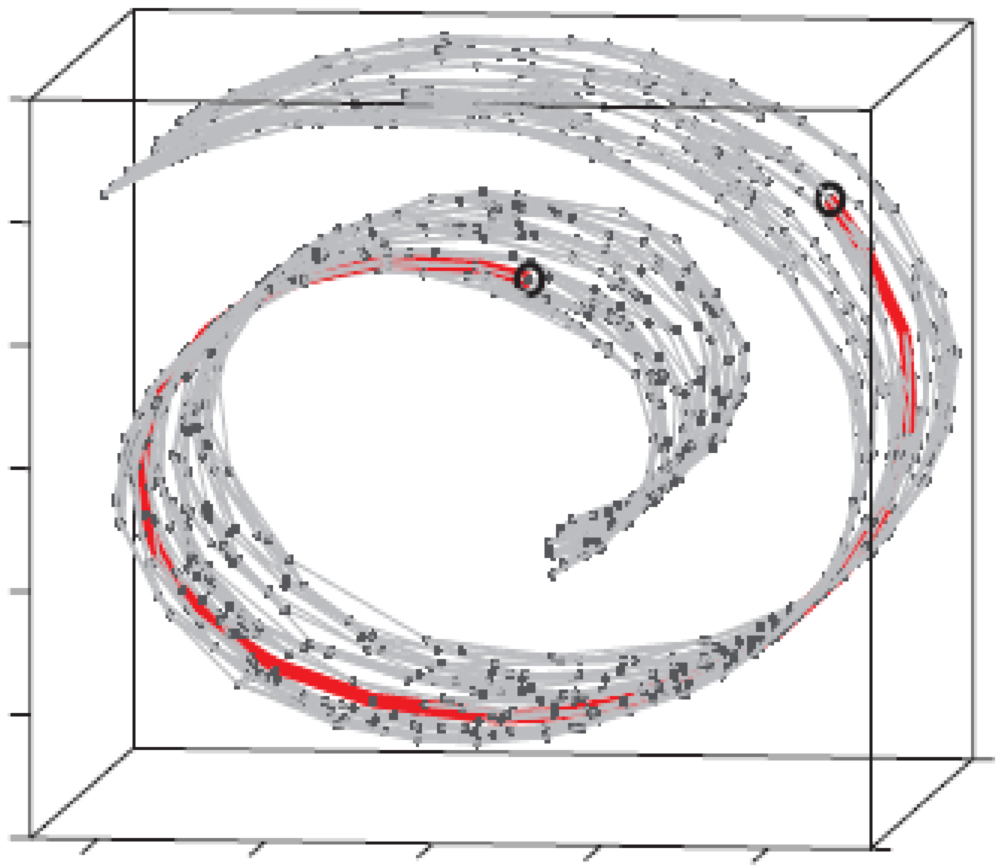
- Build a data graph G.
- Vertices: images
- (u,v) is an edge iff SSD(u,v) is small
- For any two images, we approximate the distance between them with the “shortest path” on G

Isomap

1. Build a sparse graph with K-nearest neighbors

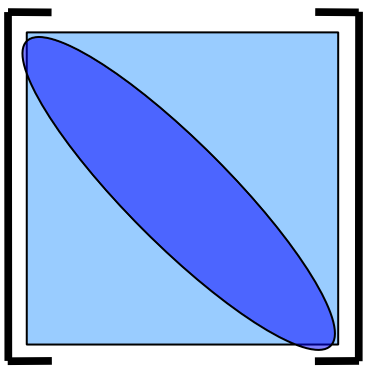
D_g = 

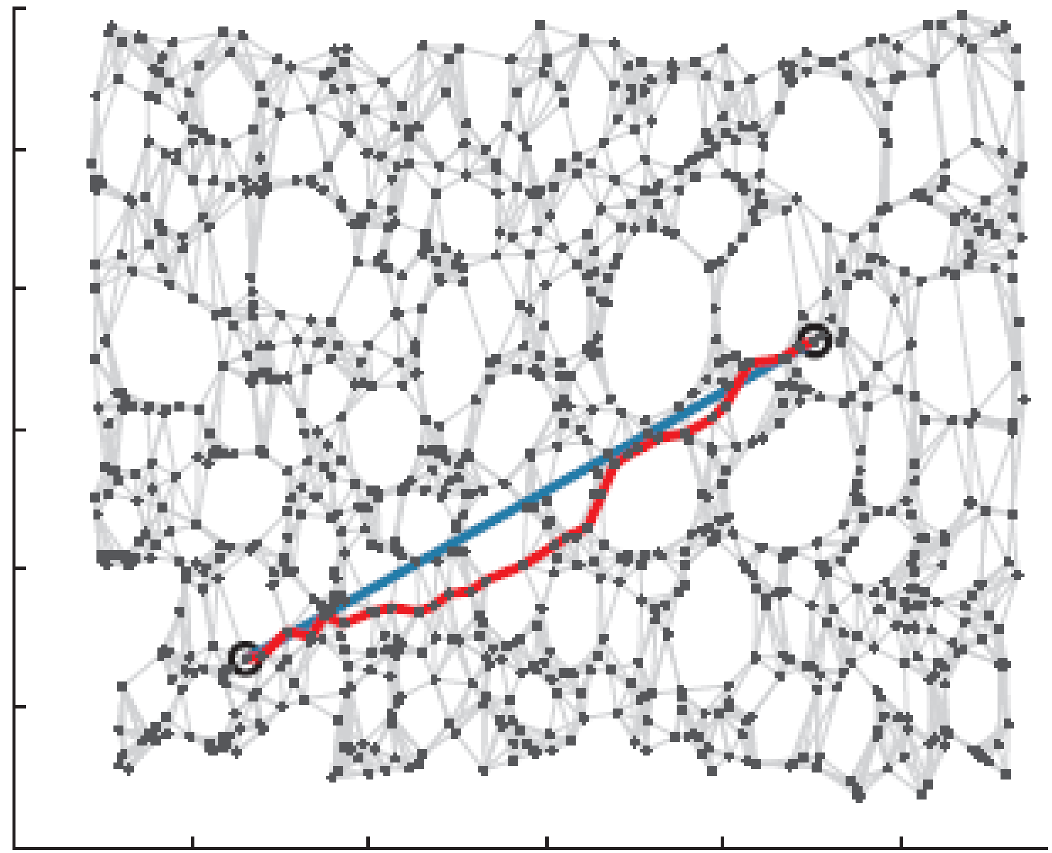
(distance matrix is sparse)



Isomap

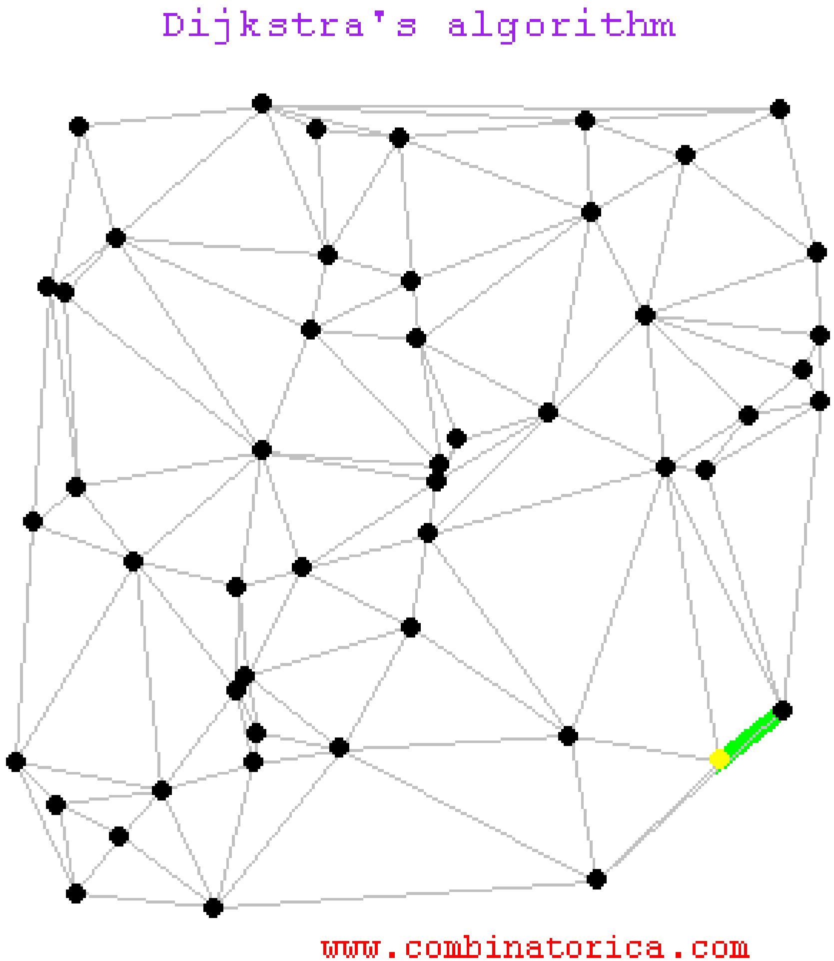
2. Infer other interpoint distances by finding shortest paths on the graph (Dijkstra's algorithm).

$D_g =$ 

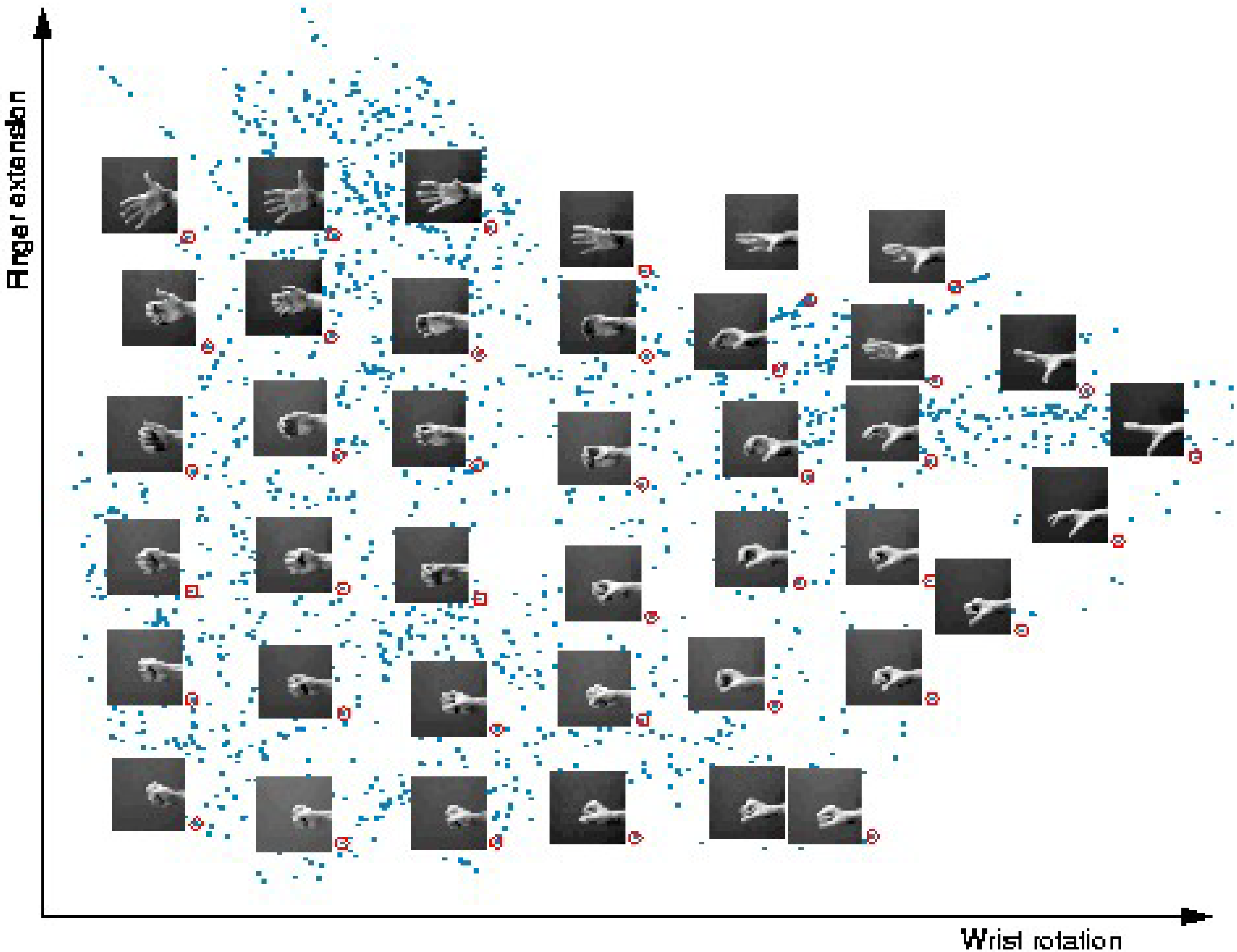


Isomap

shortest-distance on a graph is easy to compute



Isomap results: hands



Isomap: pro and con

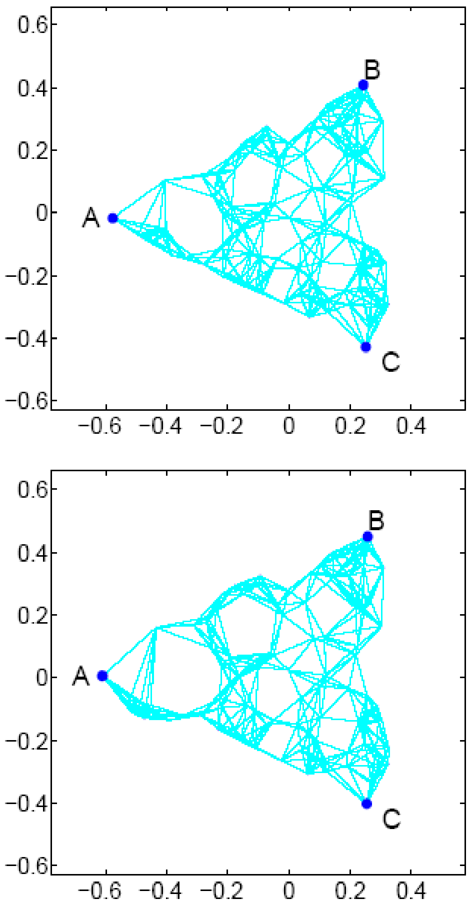
- preserves global structure

- few free parameters

- sensitive to noise, noise edges

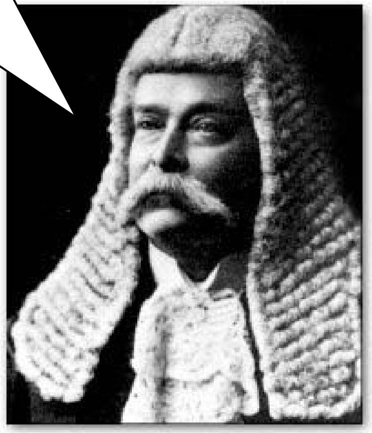
- computationally expensive (dense matrix eigen-reduction)

Leakage problem

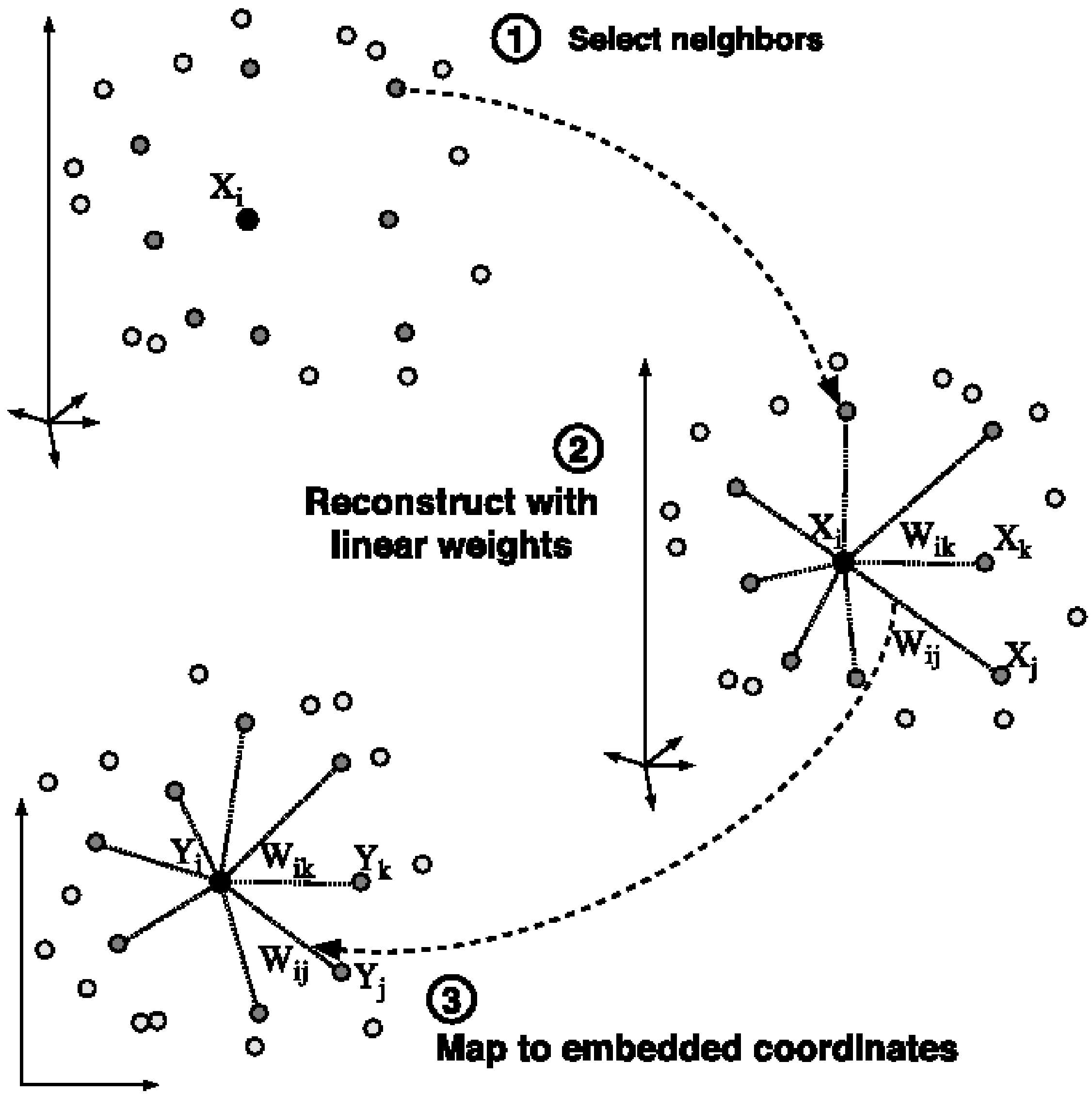


Locally Linear Embedding

Find a mapping to preserve local linear relationships between neighbors



Locally Linear Embedding



LLE: Two key steps

1. Find weight matrix W of linear coefficients:

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

Enforce sum-to-one constraint.

LLE: Two key steps

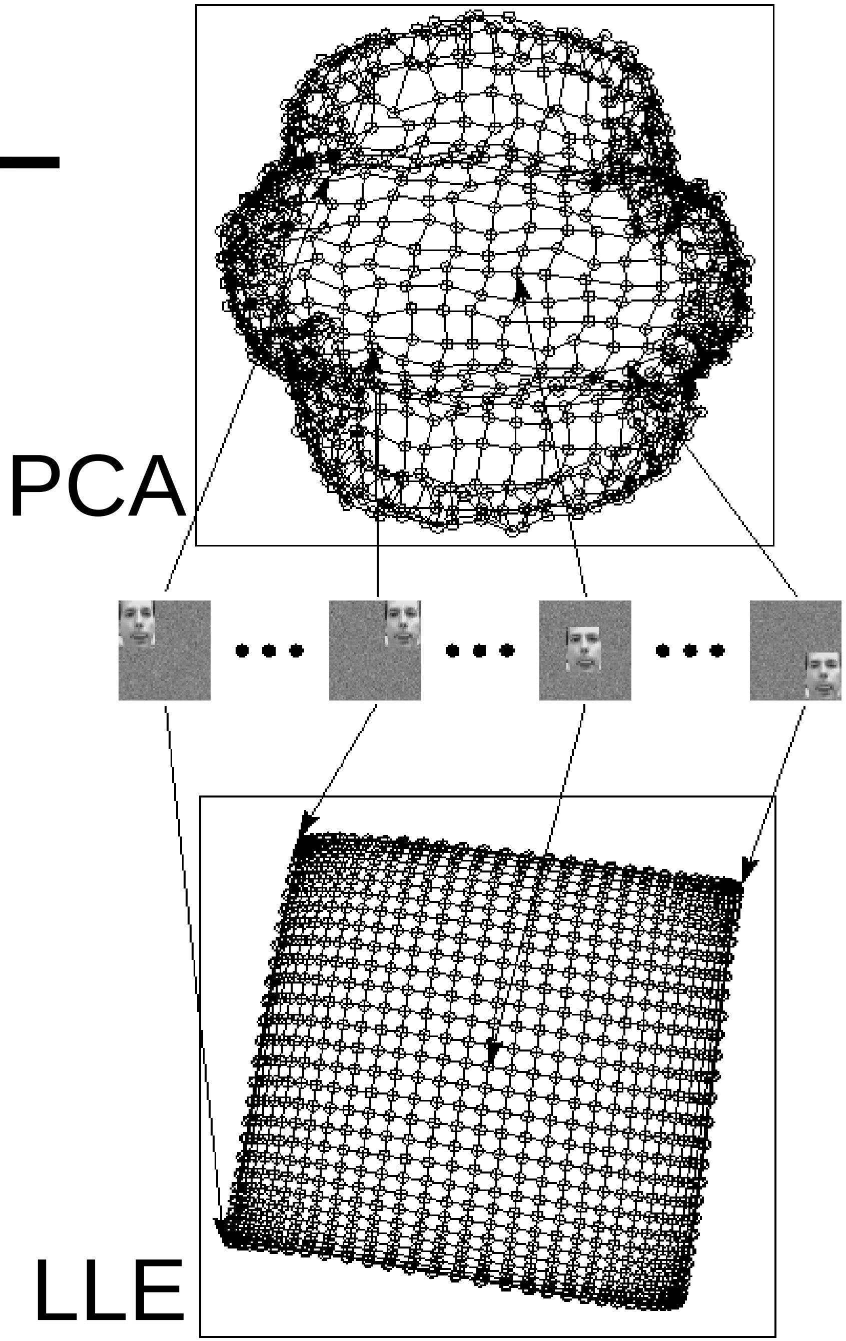
2. Find projected vectors Y to minimize reconstruction error

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

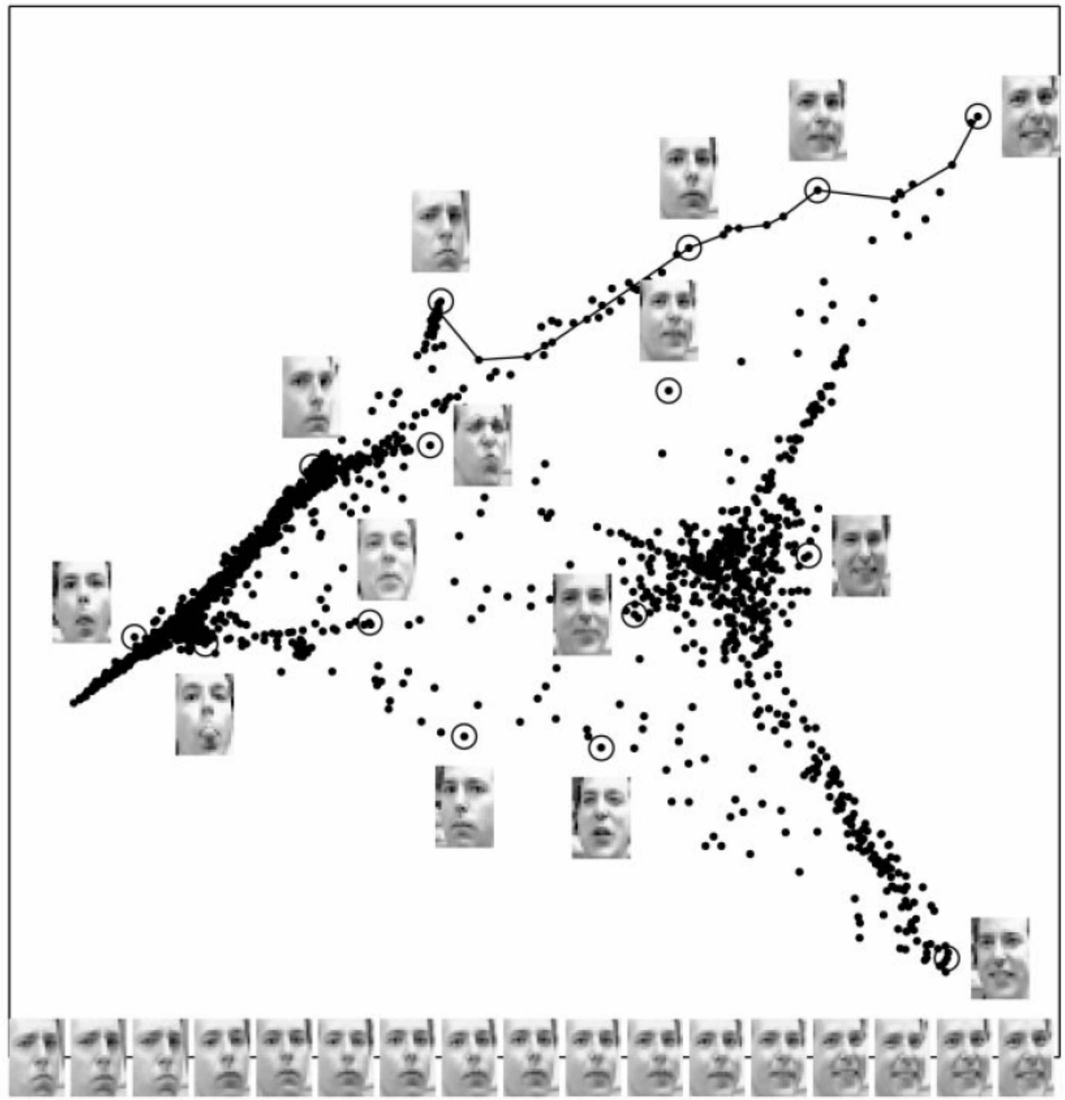
must solve for whole dataset simultaneously

LLE: Result

preserves local topology



LLE: Result



LLE: Result

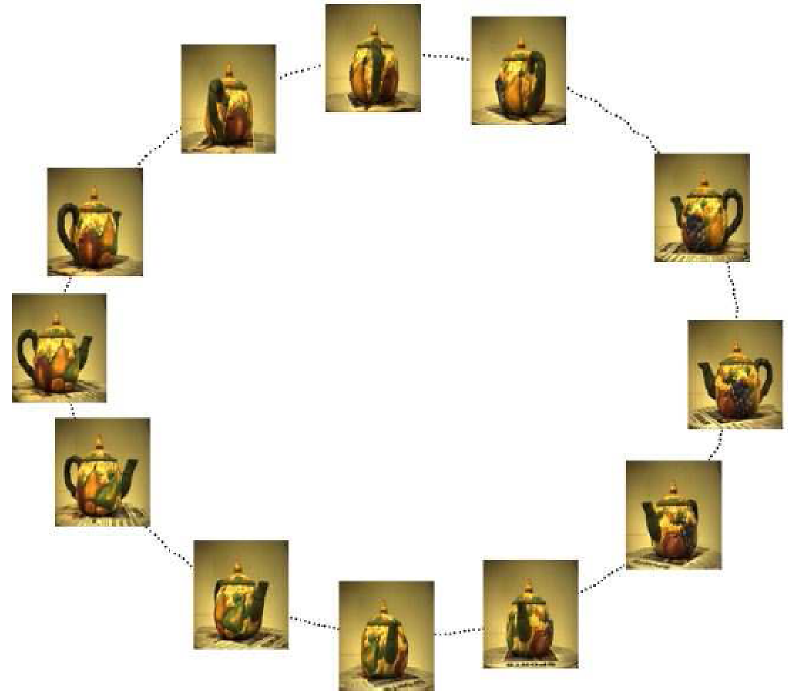


Figure 3. Two dimensional embedding of $N = 400$ images of a rotating teapot, obtained by SDE using $k = 4$ nearest neighbors. For this experiment, the teapot was rotated 360 degrees; the low dimensional embedding is a full circle. A representative sample of images are superimposed on top of the embedding.

LLE: Result

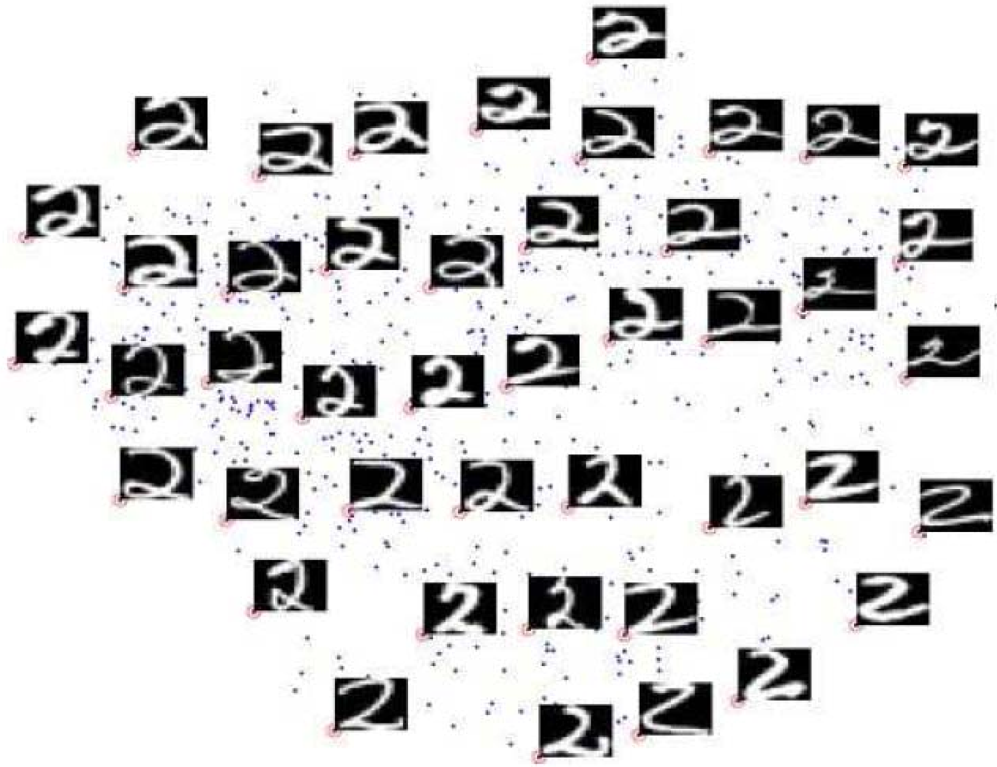


Figure 6. Results of SDE using $k = 4$ nearest neighbors on $N = 638$ images of handwritten TWOS. Representative images are shown next to circled points.

LLE: pro and con

- no local minima, one free parameter

- incremental & fast

- simple linear algebra operations

- can distort global structure