Previously

- Focus was on solving matrix inversion problems
- Now we look at other properties of matrices
- Useful when A represents a transformations

$$y = Ax$$

- Or A simply represents data
- Notion of eigenvectors, eigenvalues
- Diagonalization

Eigenvalues and Eigenvectors: Diagonalization

- Given a square matrix A and its eigenvalues and eigenvectors – matrix can be diagonalized

$$A = S \Lambda S^{-1}$$

$$A = S \Lambda S^{-1}$$

Matrix of eigenvectors

Diagonal matrix of eigenvalues

$$AS = S\Lambda$$

$$A \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} = \begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \dots & \lambda_n x_n \end{bmatrix}$$

$$A\mathbf{x} = \lambda \mathbf{x}$$

$$\begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \dots & \lambda_n x_n \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 \; \dots & \\ & & \lambda_n \end{bmatrix}$$

$$A = S \Lambda S^{-1}$$

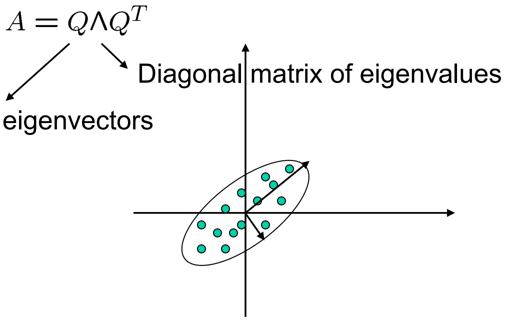
- If some of the eigenvalues are the same, eigenvectors are not independent

Diagonalization

- If there are no zero eigenvalues – matrix is invertible
- If there are no repeated eigenvalues – matrix is diagonalizable
- If all the eigenvalues are different then eigenvectors are linearly independent

For Symmetric Matrices

If A is symmetric



$$A = Q\Lambda Q^T$$

Diagonal matrix of eigenvalues

orthonormal matrix of eigenvectors

i.e. for a covariance matrix
or some matrix B = A^TA

Dimensionality Reduction

- Many dimensions are often interdependent (correlated);
- Reduce the dimensionality of problems;
- Transform interdependent coordinates into significant and independent ones;
- Linear and non-linear techniques

Singular Value Decomposition

- Previously eigenvalue-eigenvector factorization of a symmetric matrix, using diagonal an orthogonal matrix

$$A = Q\Lambda Q^T$$

- Eigenvectors of symmetric matrices can be chosen orthogonal
- What about general non-square matrices
- Key of working with non-square matrices is to consider

$$A^T A \quad or \quad AA^T$$

Singular value Decomposition

- Any *m x n* matrix A can be factored into

$$A = Q_1 \Sigma Q_2^T$$

- Where columns of *m x m* matrix Q_1 are eigenvectors of AA^T, and the columns of *n x n* matrix $A^T A$ are eigenvectors of Q_2 . TheΣ singular values on the *m x m* diagonal matrix are square roots of non-zero eigenvalues of both $A^T A$ and AA^T.

Singular value Decomposition

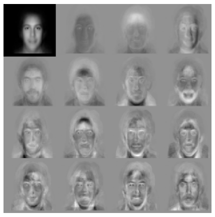
- Any *m x n* matrix A can be factored into

$$A = Q_1 \Sigma Q_2^T$$

- Columns of Q_1 and Q_2 give orthonormal basis of all fundamental subspaces of A
- First *r* columns of Q_1 : column space of A
- Last *m-r* columns of Q_1: left null space of A
- First *r* columns of Q_2: row space of A
- Last *n-r* columns of Q_2: null space of A

Example

- Image processing, computer vision face recognition
- Image e.g. matrix of 200 x 200 grey-level values can be considered as point in high-dimensional space of dimension 40,000
- Consider set of all images of faces
- Premise set of all face images spans lower dimensional linear subspace
- Use SVD to find the sub-space
- Blackboard example



Principal Component Analysis -- PCA
(also called Karhunen-Loeve transformation)

- **PCA** transforms the original input space into a lower dimensional space, by constructing dimensions that are linear combinations of the given features;
- The objective is to consider independent dimensions along which data have largest variance (i.e., greatest variability);

Geometric view

- Given set of datapoints in D dimensional space – find some transformation which will transform the points to lower dimensional space. U_d is *D x d* matrix with d orthonormal column vectors

$$x_i = x_0 + U_d y_i$$

- y_i are the new coordinates of x_i in d-dimensional space
- Derivation on the board – see handout for more details

Statistical view

- Given multivariate random variable *x* and set of sample points x_i , find d uncorrelated linear components of *x* such that variance of the components is maximized

$$y_i = u_i^T x$$

- Such that

$$u_i^T u_i = 1 \quad and \quad Var(y_1) \geq Var(y_2) \cdots$$

- Derivation on the board

Principal Component Analysis -- PCA

- **PCA** enables transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components;

- The first principal component accounts for as much of the variability in the data as possible;

- Each succeeding component (orthogonal to the previous ones) accounts for as much of the remaining variability as possible.

Curse of Dimensionality

- One way to deal with dimensionality is to assume that we know the form of the probability distribution.
- For example, a Gaussian model in N dimensions has N + N(N-1)/2 parameters to estimate.
- Requires $O(N^2)$ data to learn reliably. This may be practical.

Dimension Reduction

- One way to avoid the curse of dimensionality is by projecting the data onto a lower-dimensional space.
- Techniques for dimension reduction:
- Principal Component Analysis (PCA)
- Multi-dimensional Scaling.
- Independent Component Analysis.

Principal Component Analysis

- PCA is the most commonly used dimension reduction technique.
- (Also called the Karhunen-Loeve transform).
- Data samples $x_1, \cdots, x_N$
- Compute the mean $\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$
- Computer the covariance:

$$\Sigma_x = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

Principal Component Analysis

- Compute the eigenvalues λ and eigenvectors e of the matrix Σ_x
- Solve $\Sigma_x x = \lambda x$
- Order them by magnitude:

$$\lambda_1 \geq \lambda_2 \geq .\lambda_N.$$

- PCA reduces the dimension by keeping direction e such that $\lambda < T$.

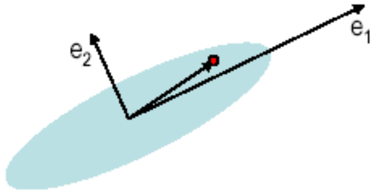
Principal Component Analysis

- For many datasets, most of the eigenvalues are negligible and can be discarded.

The eigenvalue λ measures the variation
In the direction e

Example:

$$\lambda_1 \neq 0, \lambda_2 = 0.$$



Principal Component Analysis

- How to get uncorrelated components which Capture most of the variance
- Project the data onto the selected eigenvectors:

$$y_i = e_i^T (x_i - \bar{x})$$

- If we consider first M eigenvectors we get new lower dimensional representation

$$[y_1, \cdots, y_M]$$

- Proportion covered by first M eigenvalues

$$\frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$$

PCA Example

- The images of an object under different lighting lie in a low-dimensional space.
- The original images are 256x 256. But the data lies mostly in 3-5 dimensions.
- First we show the PCA for a face under a range of lighting conditions. The PCA components have simple interpretations.
- Then we plot $\frac{\sum_{i=1}^{M}\lambda_i}{\sum_{i=1}^{N}\lambda_i}$ as a function of M for several objects under a range of lighting.

PCA on Faces.

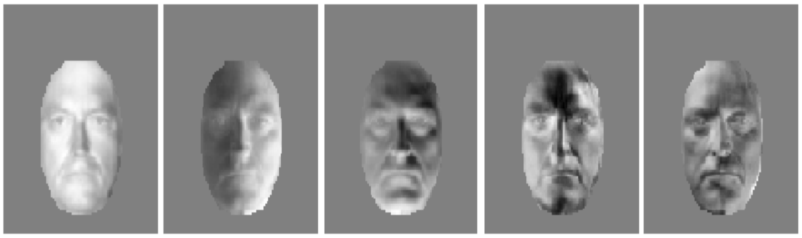
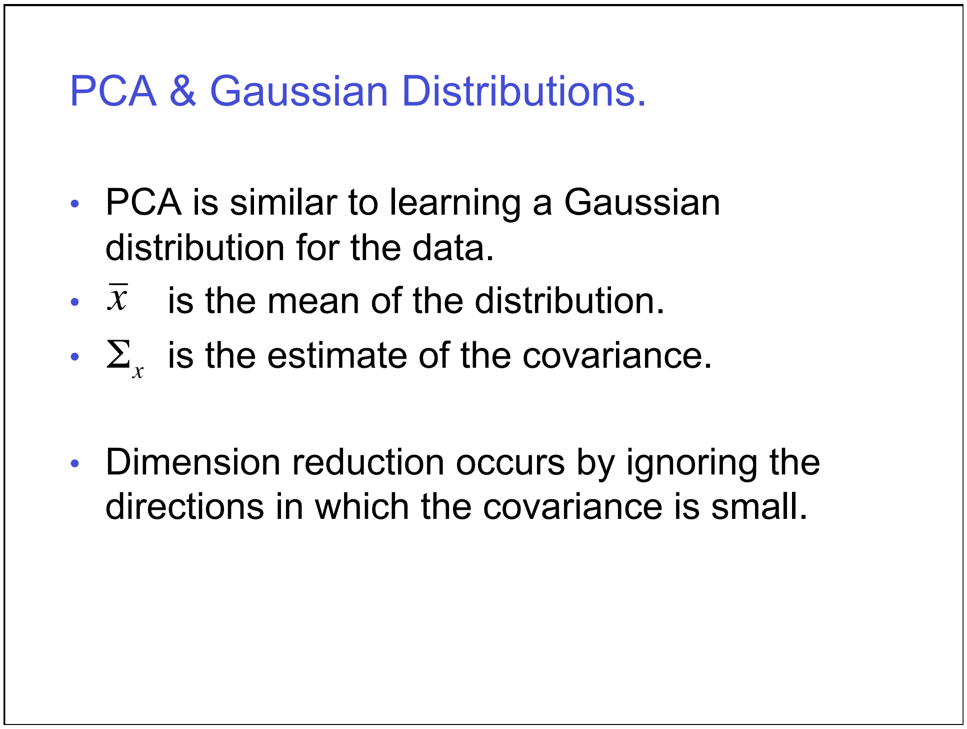
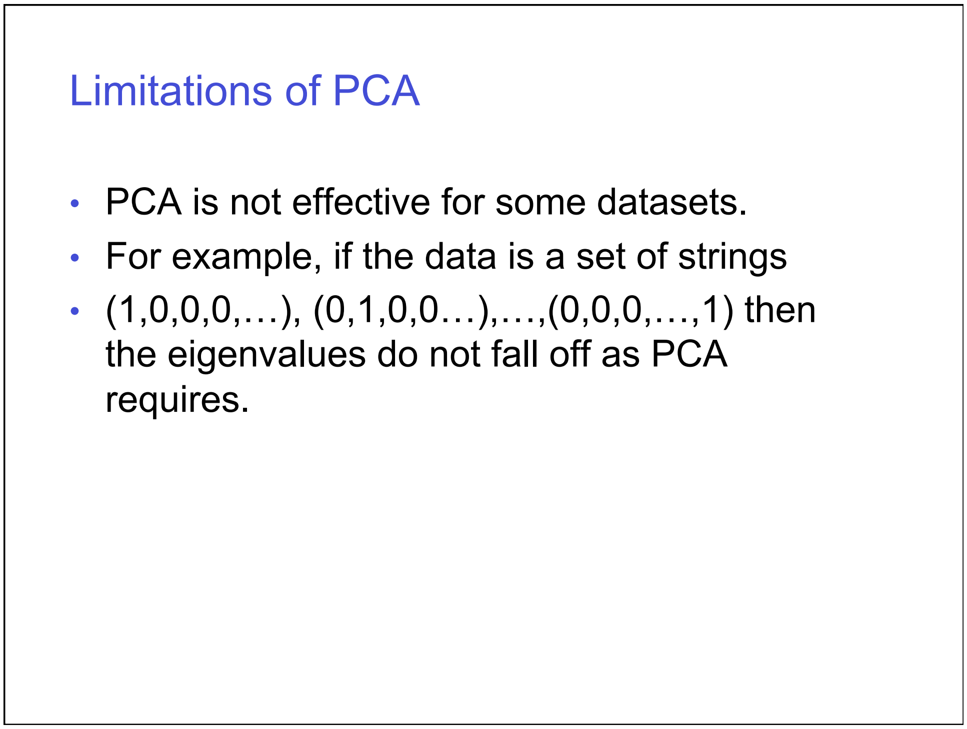


Figure 4: The eigenvectors calculated from the sparse set for the human face. Note that the images were only lit from the right so the eigenvectors are not perfectly symmetric. Observe also that the first three eigenvectors appear to be images of the face illuminated from three orthogonal lighting conditions in agreement with the orthogonal lighting conjecture.



PCA & Gaussian Distributions.

- PCA is similar to learning a Gaussian distribution for the data.
- $\overline{x}$ is the mean of the distribution.
- Σ_x is the estimate of the covariance.
- Dimension reduction occurs by ignoring the directions in which the covariance is small.

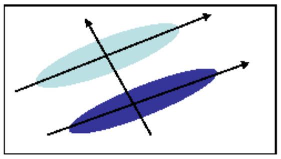


Limitations of PCA

- PCA is not effective for some datasets.
- For example, if the data is a set of strings
- (1,0,0,0,…), (0,1,0,0…),…,(0,0,0,…,1) then the eigenvalues do not fall off as PCA requires.

PCA and Discrimination

- PCA may not find the best directions for discriminating between two classes.
- Example: suppose the two classes have 2D Gaussian densities as ellipsoids.
- 1st eigenvector is best for representing the probabilities.
- 2nd eigenvector is best for discrimination.



Principal Component Analysis -- PCA

- Statistical view of PCA
- PCA finds *n* linearly transformed components so that they explain the maximum amount of variance
- See hand out/blackboard how to compute the largest principal component
- We can define PCA in an intuitive way using a recursive formulation:

Principal Component Analysis -- PCA

- Suppose data are first centered at the origin (i.e., their mean is ***0***);
- We define the direction of the first principal component, say $\boldsymbol{w}_1$, as follows

$$u_1 = \arg \max_{\|u\|=1} E[(u^T x)^2]$$

where u_1 is of the same dimensionality $\boldsymbol{q}$ as the data vector $\boldsymbol{x}$

- Thus: the first principal component is the projection on the direction along which the variance of the projection is maximized.

Principal Component Analysis -- PCA

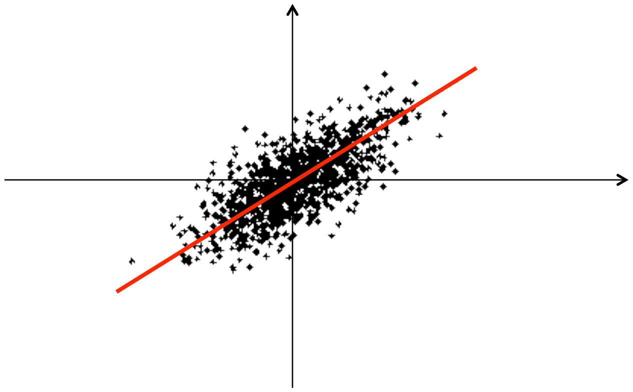
- Having determined the first *k-1* principal components, the *k*-th principal component is determined as the principal component of the data residual:

$$\boldsymbol{u}_k = \arg \max_{\|\boldsymbol{w}\|=1} E\{[\boldsymbol{u}^T (\boldsymbol{x} - \sum_{i=1}^{k-1} \boldsymbol{u}_i \boldsymbol{u}_i^T \boldsymbol{x})]^2\}$$

- The principal components are then given by:

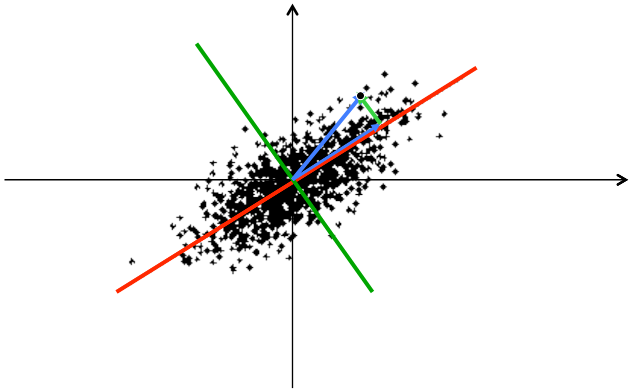
$$y_i = \boldsymbol{u}_i^T \boldsymbol{x}$$

Simple illustration of PCA



First principal component of a two-dimensional data set.

Simple illustration of PCA



Second principal component of a two-dimensional data set.

PCA – Geometric interpretation

- PCA computes new coordinates of points, i.e. the rotates the data (centered at the origin) in such a way that the maximum variability of the data is aligned with the axes.)

PCA – How to compute the principal components

Let $\boldsymbol{w}$ be the direction of the first principal component, with $\|\boldsymbol{w}\| = 1$

$s_i = \boldsymbol{w}^T \boldsymbol{x}_i$ is the projection of $\boldsymbol{x}_i$ along $\boldsymbol{w}$

$$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{w}^T \boldsymbol{x}_i$$

Variance of data along $\boldsymbol{w}$:

$$\frac{1}{N}\sum_{i=1}^{N} \left(s_i - \bar{s}\right)^2 =$$

$$\frac{1}{N}\sum_{i=1}^{N} \left(\boldsymbol{w}^T \boldsymbol{x}_i - \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{w}^T \boldsymbol{x}_j\right)^2$$

PCA – How to compute the principal components

$$\frac{1}{N}\sum_{i=1}^{N}\left(s_i-\bar{s}\right)^2 =$$

$$\frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{w}^T\boldsymbol{x}_i-\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{w}^T\boldsymbol{x}_j\right)^2 =$$

$$\frac{1}{N}\sum_{i=1}^{N}\left[\boldsymbol{w}^T\left(\boldsymbol{x}_i-\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{x}_j\right)\right]^2 =$$

$$\frac{1}{N}\sum_{i=1}^{N}\left[\boldsymbol{w}^T\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)\right]^2 =$$

$$\frac{1}{N}\sum_{i=1}^{N}\left[\boldsymbol{w}^T\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)^T\boldsymbol{w}\right] =$$

$$\boldsymbol{w}^T\left\{\boxed{\frac{1}{N}\sum_{i=1}^{N}\left[\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)^T\right]}\right\}\boldsymbol{w} \;=\; \boldsymbol{w}^T\Sigma\boldsymbol{w}$$

Sample covariance matrix

PCA – How to compute the principal components

Thus: the variance of data along direction $\boldsymbol{w}$ can be written as

$$\boldsymbol{w}^T\Sigma\boldsymbol{w}$$

Our objective is to find $\boldsymbol{w}$ such that

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} \boldsymbol{w}^T\Sigma\boldsymbol{w}$$

with the constraint $\boldsymbol{w}^T\boldsymbol{w} = 1$

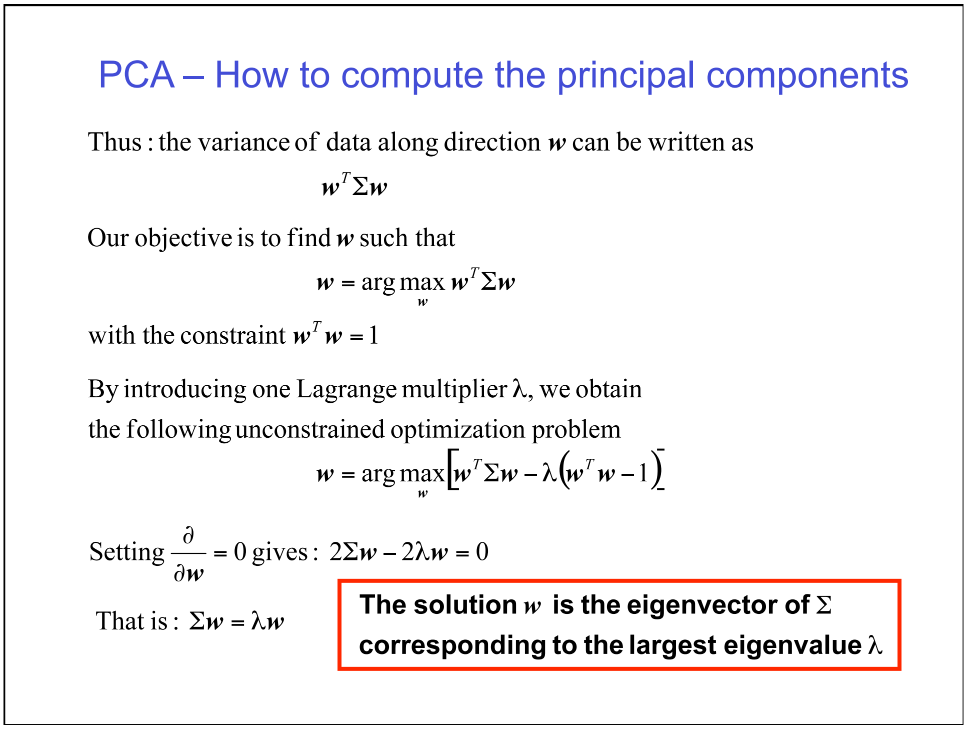
By introducing one Lagrange multiplier λ, we obtain the following unconstrained optimization problem

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}}\left[\boldsymbol{w}^T\Sigma\boldsymbol{w}-\lambda\left(\boldsymbol{w}^T\boldsymbol{w}-1\right)\right]$$

Setting $\frac{\partial}{\partial \boldsymbol{w}} = 0$ gives: $2\Sigma\boldsymbol{w}-2\lambda\boldsymbol{w} = 0$

That is: $\Sigma\boldsymbol{w} = \lambda\boldsymbol{w}$

Our problem is reduced to an eigenvalue problem



PCA – How to compute the principal components

Thus : the variance of data along direction $\boldsymbol{w}$ can be written as

$$\boldsymbol{w}^T \Sigma \boldsymbol{w}$$

Our objective is to find $\boldsymbol{w}$ such that

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} \boldsymbol{w}^T \Sigma \boldsymbol{w}$$

with the constraint $\boldsymbol{w}^T \boldsymbol{w} = 1$

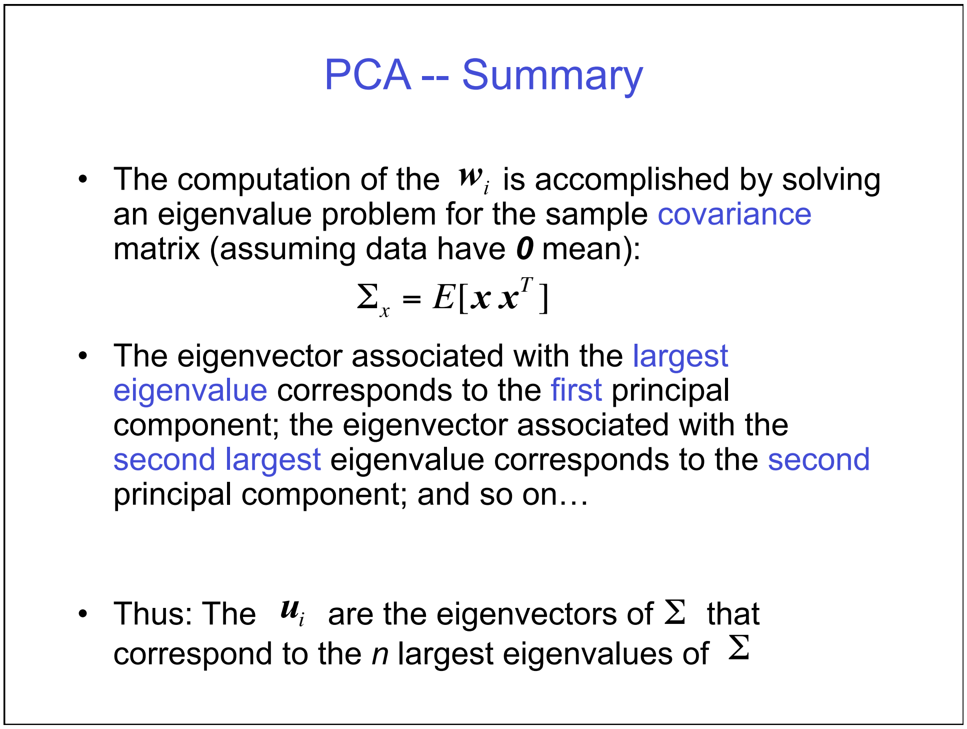
By introducing one Lagrange multiplier λ, we obtain the following unconstrained optimization problem

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} \left[\boldsymbol{w}^T \Sigma \boldsymbol{w} - \lambda\left(\boldsymbol{w}^T \boldsymbol{w} - 1\right) \right]$$

Setting $\frac{\partial}{\partial \boldsymbol{w}} = 0$ gives : $2\Sigma \boldsymbol{w} - 2\lambda \boldsymbol{w} = 0$

That is : $\Sigma \boldsymbol{w} = \lambda \boldsymbol{w}$

The solution w is the eigenvector of Σ corresponding to the largest eigenvalue λ



PCA -- Summary

- The computation of the $\boldsymbol{w}_i$ is accomplished by solving an eigenvalue problem for the sample covariance matrix (assuming data have ***0*** mean):

$$\Sigma_x = E[\boldsymbol{x}\,\boldsymbol{x}^T]$$

- The eigenvector associated with the largest eigenvalue corresponds to the first principal component; the eigenvector associated with the second largest eigenvalue corresponds to the second principal component; and so on…

- Thus: The $\boldsymbol{u}_i$ are the eigenvectors of Σ that correspond to the *n* largest eigenvalues of Σ

PCA -- In practice

- The basic goal of PCA is to reduce the dimensionality of the data. Thus, one usually chooses:

$$n << q$$

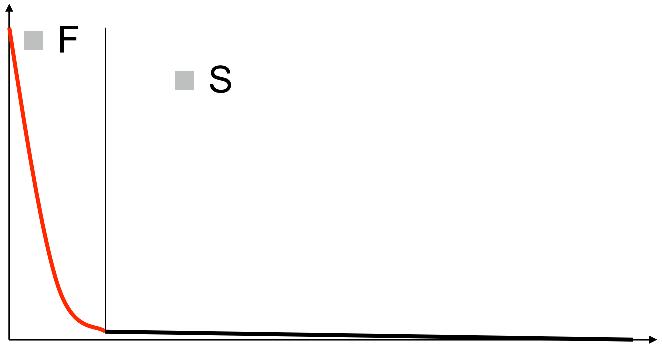
- But how do we select the number of components *n* ?

Determining the number of components

- Plot the eigenvalues – each eigenvalue is related to the amount of variation explained by the corresponding axis (eigenvector);
- If the points on the graph tend to level out (show an “elbow” shape), these eigenvalues are usually close enough to zero that they can be ignored.

- In general: Limit the variance accounted for.

Critical information lies in low dimensional subspaces



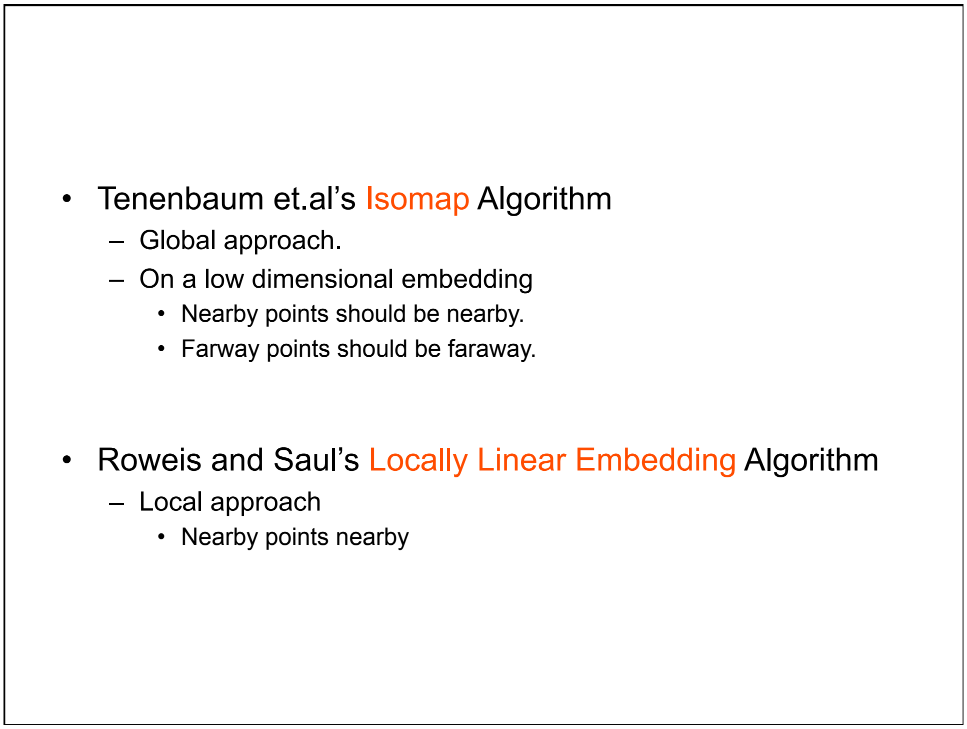
■ A typical eigenvalue spectrum and its division into two orthogonal subspaces

Applications

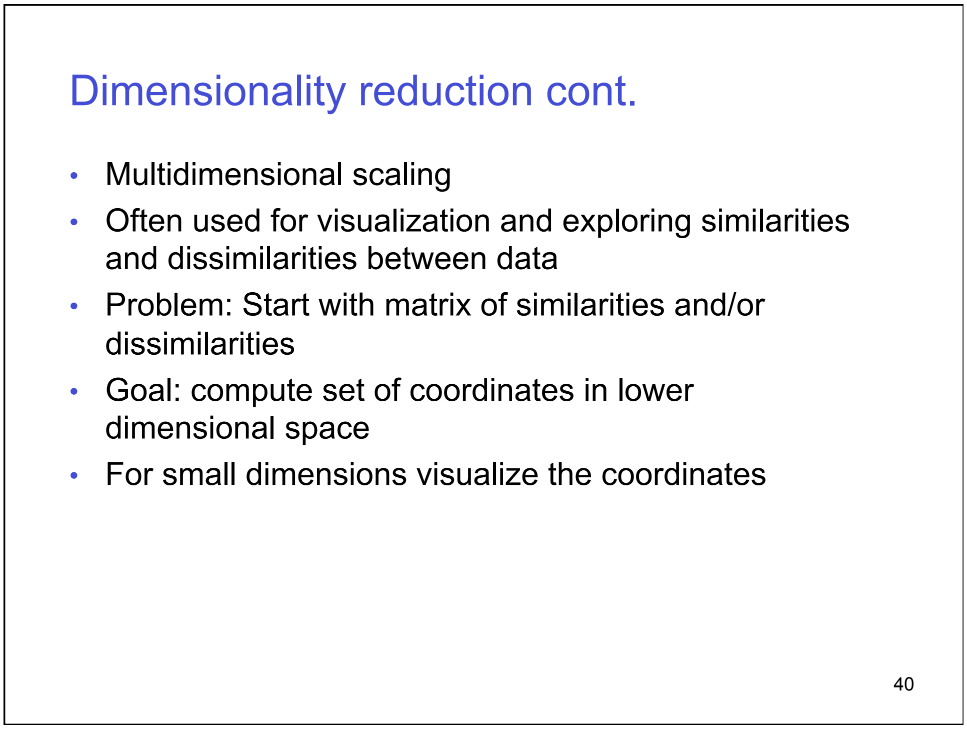
- Need to analyze large amounts multivariate data.
  - Human Faces.
  - Speech Waveforms.
  - Global Climate patterns.
  - Gene Distributions.

- Difficult to visualize data in dimensions just greater than three.

- Discover compact representations of high dimensional data.
  - Visualization.
  - Compression.
  - Better Recognition.
  - Probably meaningful dimensions.



- Tenenbaum et.al’s Isomap Algorithm
  - Global approach.
  - On a low dimensional embedding
    - Nearby points should be nearby.
    - Farway points should be faraway.
- Roweis and Saul’s Locally Linear Embedding Algorithm
  - Local approach
    - Nearby points nearby



Dimensionality reduction cont.

- Multidimensional scaling
- Often used for visualization and exploring similarities and dissimilarities between data
- Problem: Start with matrix of similarities and/or dissimilarities
- Goal: compute set of coordinates in lower dimensional space
- For small dimensions visualize the coordinates

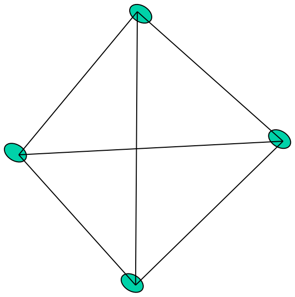
Multidimensional scaling

- Given (dis) similarity between more general objects
- Find coordinates where distances are preserved
- Example: given systems A … to … F rate their similarity at 1 to 10 scale
- For three systems it is easy to rank them in 1D to see the similarities
- With many systems more dimensions are needed
- Key is to define distance/similarity measure
- This depends of the type of data

Non-linear dimensionality reduction

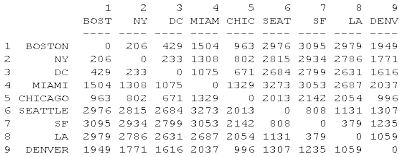
- MDS relies of distance measurements
- Discover true, linear structure of the data
- PCA finds low-dimensional embedding
- MDS finds embedding which preserves distances
- If distances are Euclidean MDS is equivalent to PCA

- Multidimensional scaling
- Here we are given pairwise distances instead of the actual data points.
- First convert the pairwise distance matrix into the dot product matrix XX^T
- After that same as PCA.

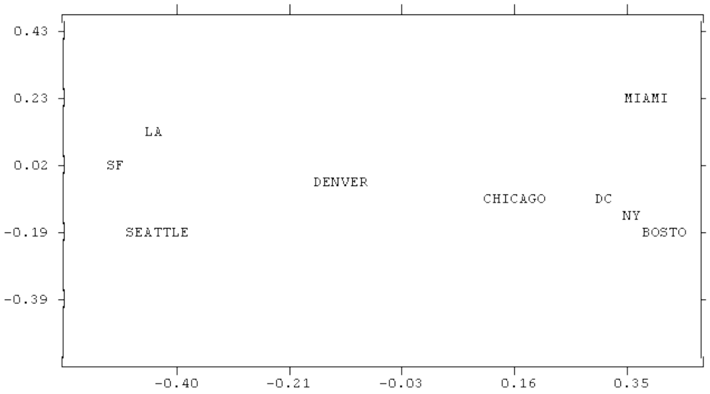


If we preserve the pairwise distances do we preserve the structure??

Example Multidimensional scaling



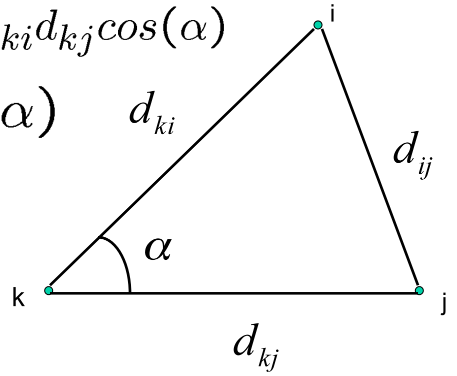
```
                  1    2    3    4    5    6    7    8    9
               BOST   NY   DC MIAM CHIC SEAT   SF   LA DENV
               ---- ---- ---- ---- ---- ---- ---- ---- ----
1   BOSTON        0  206  429 1504  963 2976 3095 2979 1949
2       NY      206    0  233 1308  802 2815 2934 2786 1771
3       DC      429  233    0 1075  671 2684 2799 2631 1616
4    MIAMI     1504 1308 1075    0 1329 3273 3053 2687 2037
5  CHICAGO      963  802  671 1329    0 2013 2142 2054  996
6  SEATTLE     2976 2815 2684 3273 2013    0  808 1131 1307
7       SF     3095 2934 2799 3053 2142  808    0  379 1235
8       LA     2979 2786 2631 2687 2054 1131  379    0 1059
9   DENVER     1949 1771 1616 2037  996 1307 1235 1059    0
```



Distances and inner products

$$d_{ij}^2 = d_{ki}^2 + d_{kj}^2 - 2d_{ki}d_{kj}cos(\alpha)$$

$$b_{ij} = d_{ki}d_{kj}cos(\alpha)$$



$$b_{ij} = \frac{1}{2}(d_{ki}^2 + d_{kj}^2 - d_{ij}^2)$$

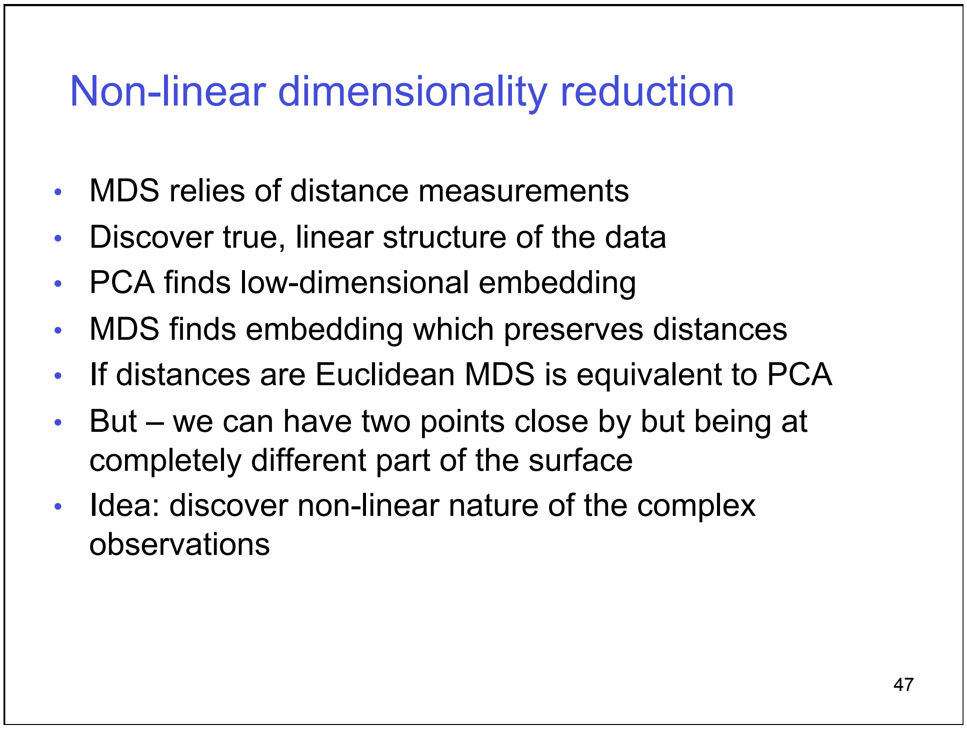
From distances to inner products

- MDS—origin as one of the points and orientation arbitrary
- How to compute the matrix of inner products $B = XX^T$
- Can be computed from distances $b_{ij} = x_i^T x_j$

$$b_{ij}^* = -\frac{1}{2}\left[d_{ij}^2 - \frac{1}{N}\sum_{l=1}^{N} d_{il}^2 - \frac{1}{N}\sum_{m=1}^{N} d_{mj}^2 + \frac{1}{N^2}\sum_{o=1}^{N}\sum_{p=1}^{N} d_{op}^2\right]$$

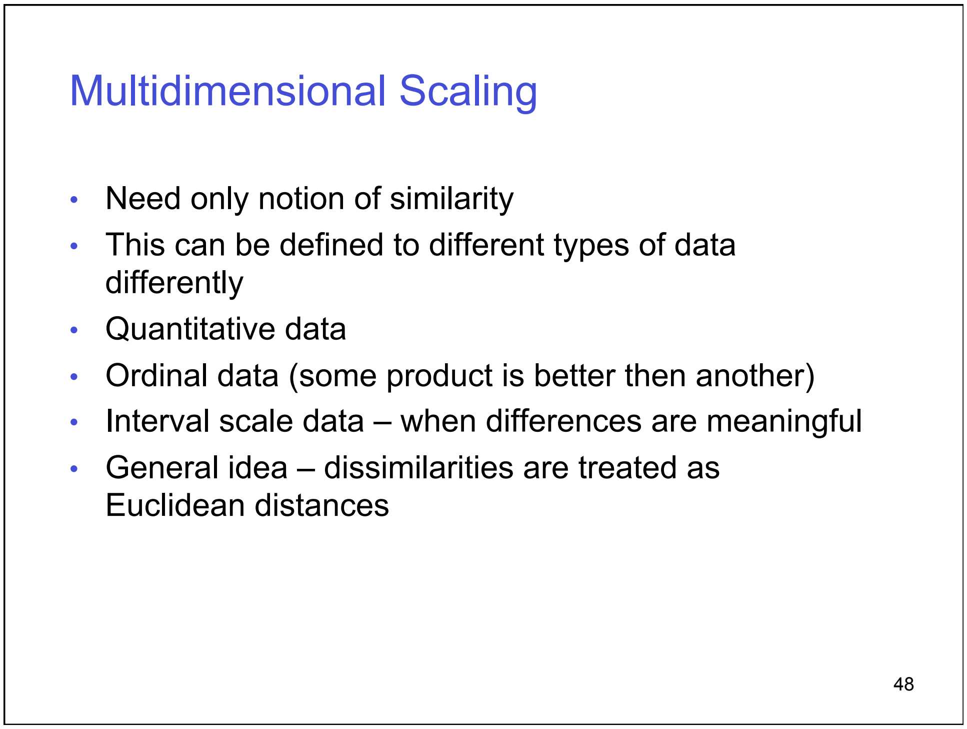
- In matrix form

$$B = \frac{1}{2}(I - u^T u)D(I - uu^T) \qquad u = \frac{1}{\sqrt{n}}(1,\cdots,1)$$



Non-linear dimensionality reduction

- MDS relies of distance measurements
- Discover true, linear structure of the data
- PCA finds low-dimensional embedding
- MDS finds embedding which preserves distances
- If distances are Euclidean MDS is equivalent to PCA
- But – we can have two points close by but being at completely different part of the surface
- Idea: discover non-linear nature of the complex observations

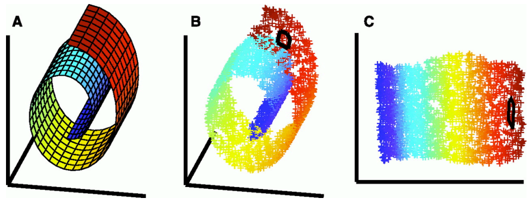


Multidimensional Scaling

- Need only notion of similarity
- This can be defined to different types of data differently
- Quantitative data
- Ordinal data (some product is better then another)
- Interval scale data – when differences are meaningful
- General idea – dissimilarities are treated as Euclidean distances

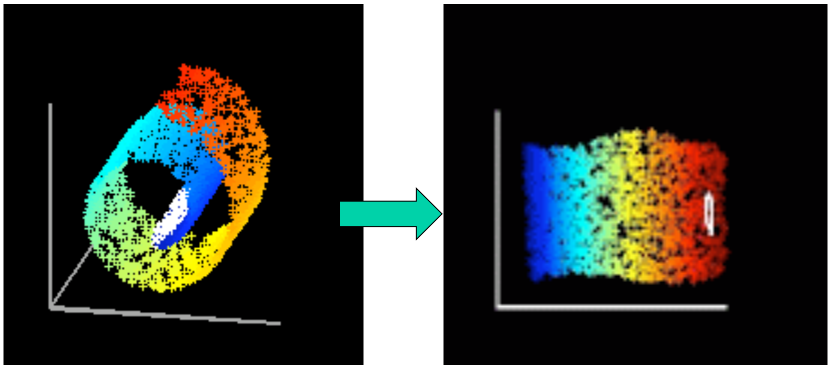
Non-linear dimensionality reduction

- Estimate the geodesic distance between faraway points.
- For neighboring points Euclidean distance is a good approximation to the geodesic distance
- Combine advantages of PCA and MDS
- For farway points estimate the distance by a series of short hops between neighboring points.
  - Find shortest paths in a graph with edges connecting neighboring data points



- Manifold is a topological space which is locally Euclidean.”

Fit Locally , Think Globally

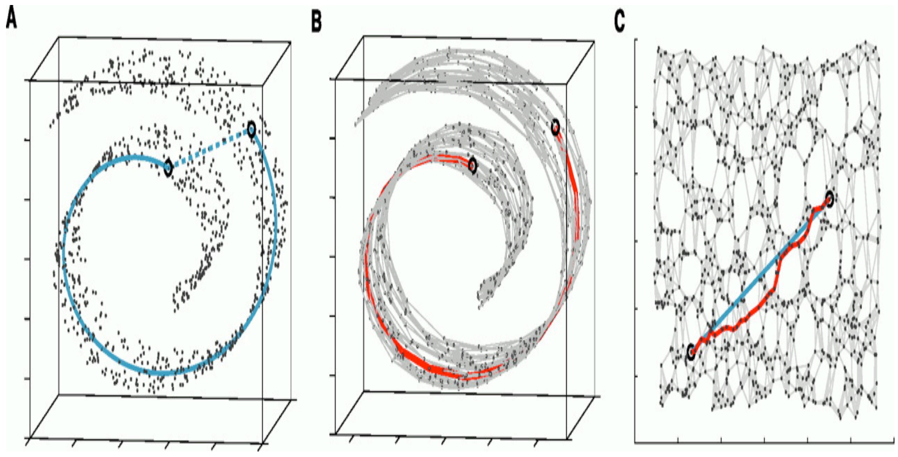


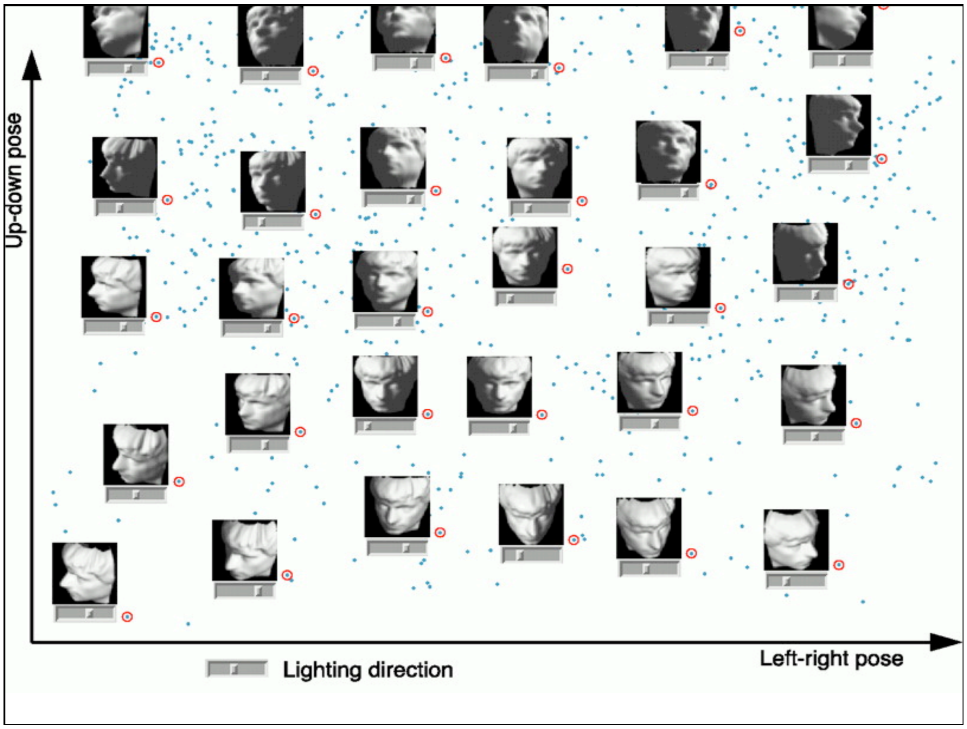
ISOMAP algorithms

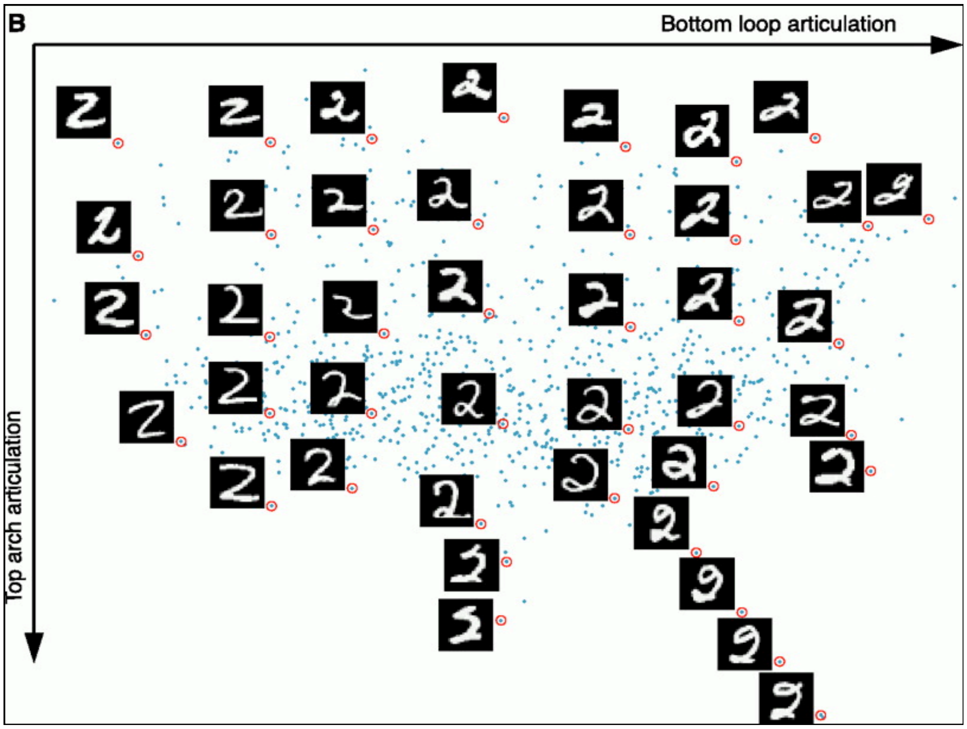
- Determine the neighbors.
  - All points in a fixed radius.
  - K nearest neighbors
- Construct a neighborhood graph.
  - Each point is connected to the other if it is a K nearest neighbor.
  - Edge Length equals the Euclidean distance
- Compute the shortest paths between two nodes
  - Floyd’s Algorithm
  - Dijkstra’s ALgorithm
- Construct a lower dimensional embedding.

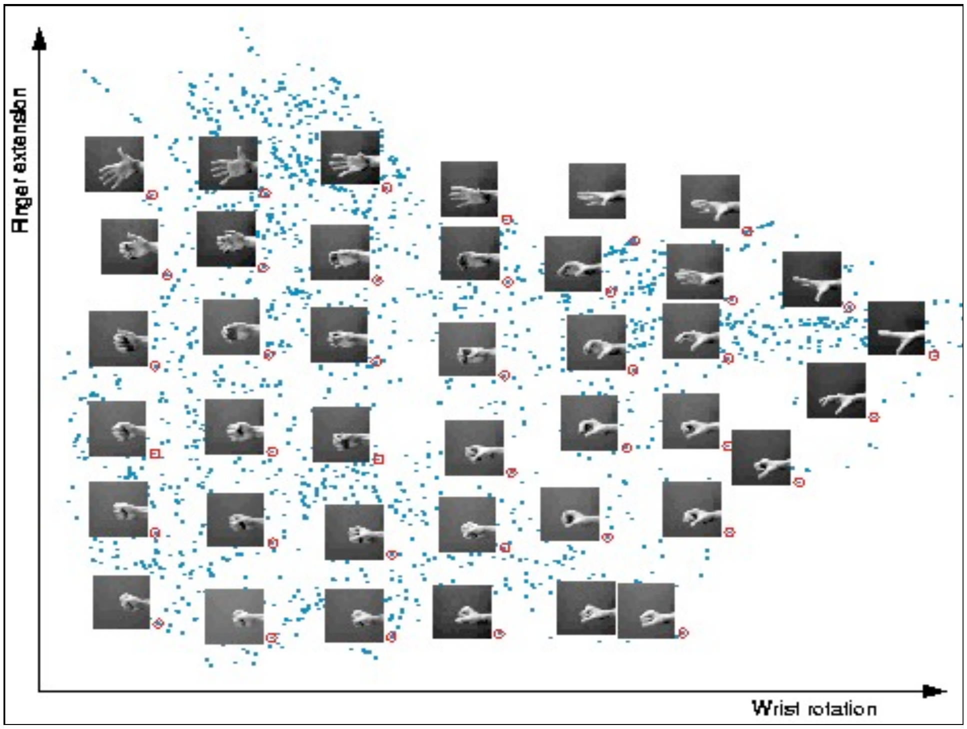
  Classical MDS

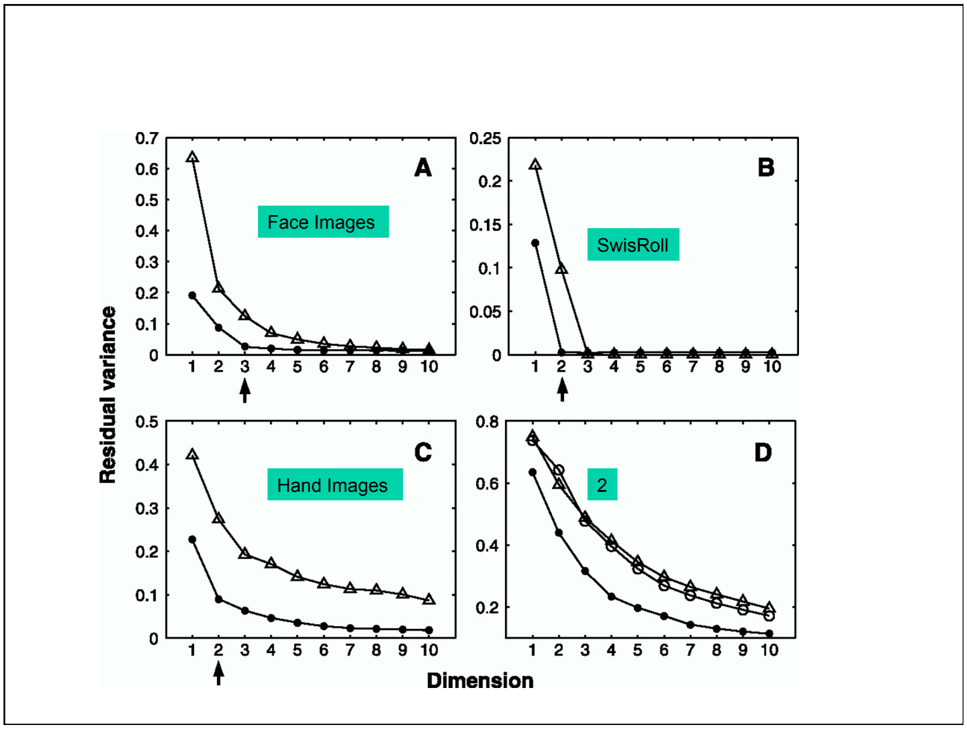
  See http://isomap.stanford.edu/ for more details



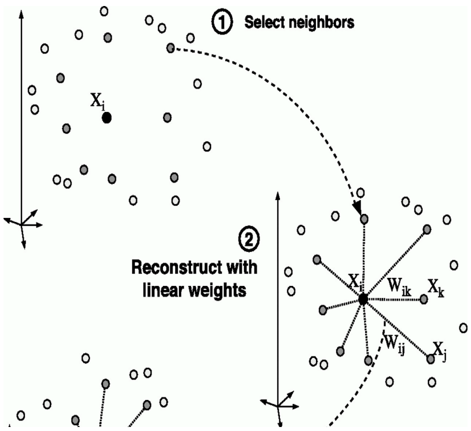

Up-down pose
Left-right pose
Lighting direction


B
Bottom loop articulation
Top arch articulation


Finger extension
Wrist rotation


Residual variance
Dimension
A
Face Images
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 2 3 4 5 6 7 8 9 10
B
SwisRoll
0.25
0.2
0.15
0.1
0.05
0
1 2 3 4 5 6 7 8 9 10
C
Hand Images
0.5
0.4
0.3
0.2
0.1
0
1 2 3 4 5 6 7 8 9 10
D
2
0.8
0.6
0.4
0.2
1 2 3 4 5 6 7 8 9 10

Locally Linear Embedding



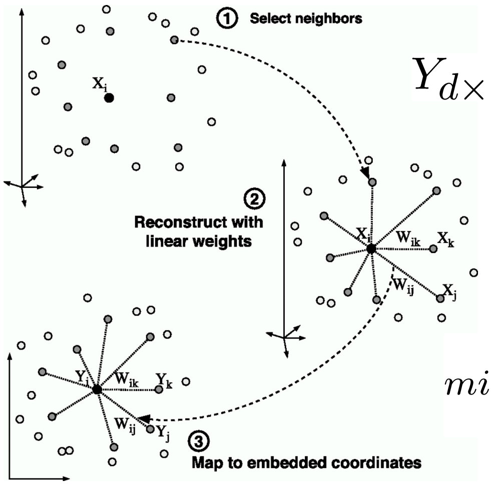
- We expect each data point and its
- neighbours to lie on or close
- to a locally linear patch of the
- manifold.
- Each point can be written as a linear combination of its neighbors.
- The weights chosen to minimize the reconstruction error.

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

Locally Linear Embedding

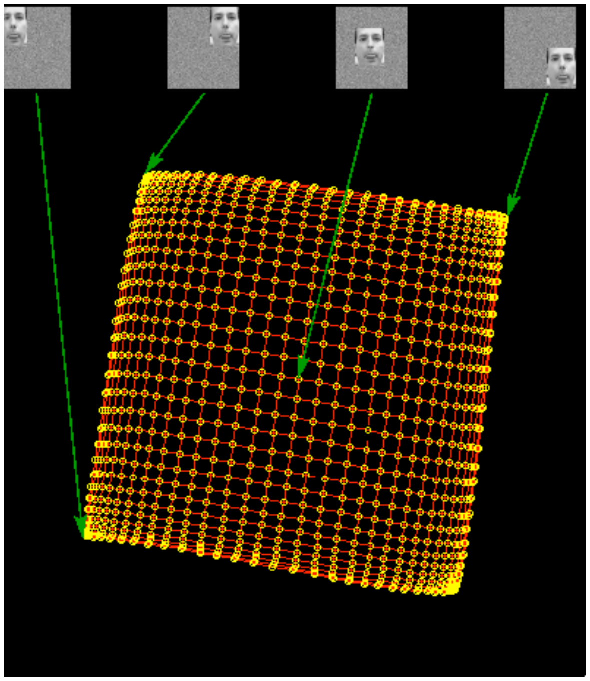
- The weights that minimize the reconstruction errors are invariant to rotation, rescaling and translation of the data points.
  - Invariance to translation is enforced by adding the constraint that the weights sum to one.
- The same weights that reconstruct the datapoints in D dimensions should reconstruct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

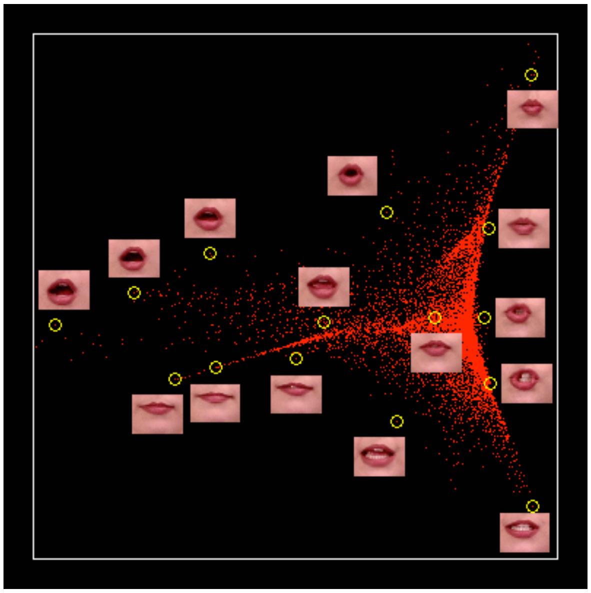
Locally Linear Embedding

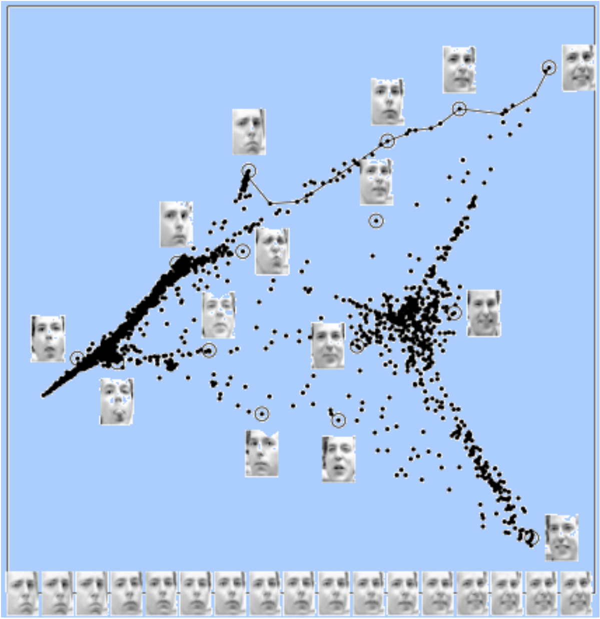


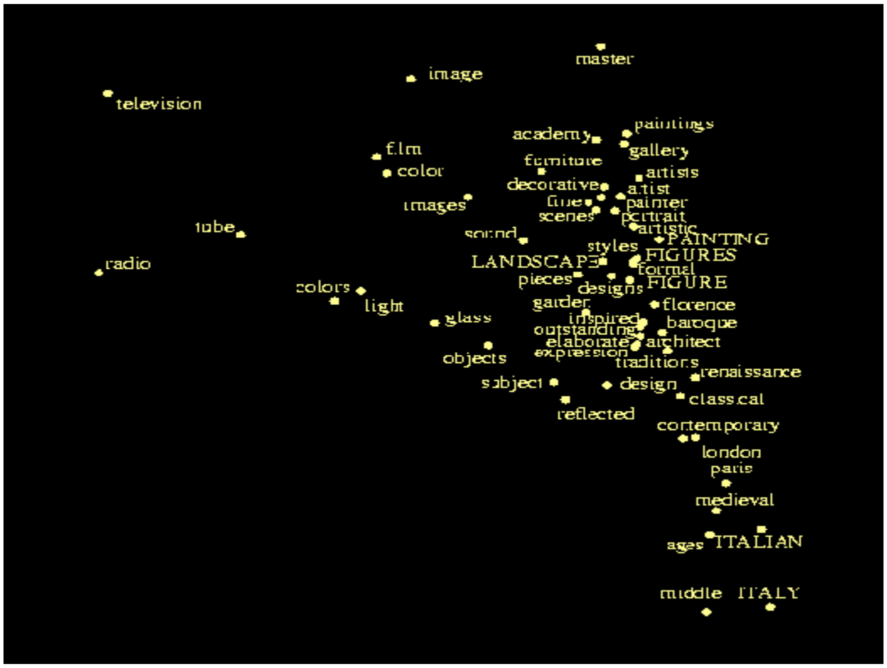
$$Y_{d\times N} = [Y_1|Y_2|\ldots|Y_N]$$

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









Summary..

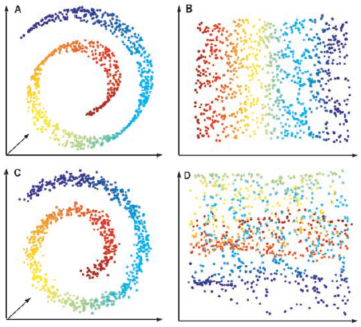
ISOMAP	LLE
Do MDS on the geodesic distance matrix.	Model local neighborhoods as linear a patches and then embed in a lower dimensional manifold.
Global approach	Local aproach
Dynamic programming approaches	Computationally efficient..sparse matrices
Convergence limited by the manifold curvature and number of points.	Good representational capacity

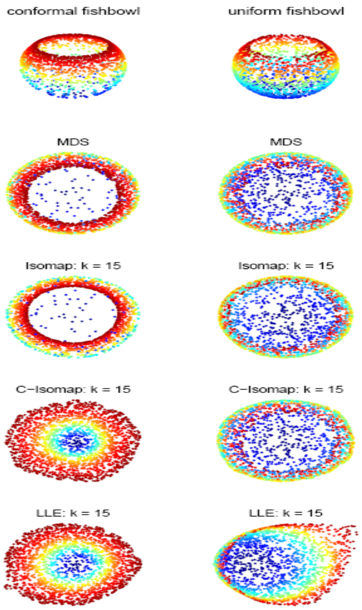
Unstable?

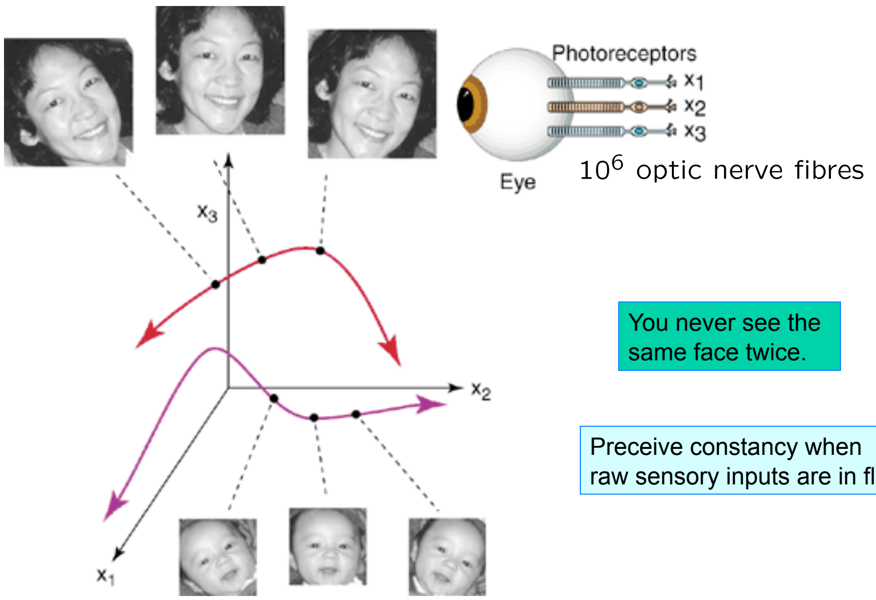
Only free parameter is

How many neighbours?

- How to choose neighborhoods.
  - Susceptible to short-circuit errors if neighborhood is larger than the folds in the manifold.
  - If small we get isolated patches.



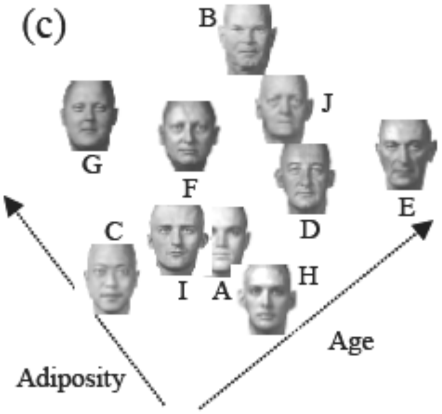




You never see the same face twice.

Preceive constancy when raw sensory inputs are in flux..

- Instead of pairwise distances we can use paiwise “dissimilarities”.
- When the distances are Euclidean MDS is equivalent to PCA.
- Eg. Face recognition, wine tasting
- Can get the significant cognitive dimensions.


(c)
B
J
G
F
D
E
C
I
A
H
Age
Adiposity