PA 2018 - 07

Manifold Learning:

Goal/Purpose:

Reduce the dimensionality of the data while preserving its structure

Example:

consider a noisy plane in 3-D:

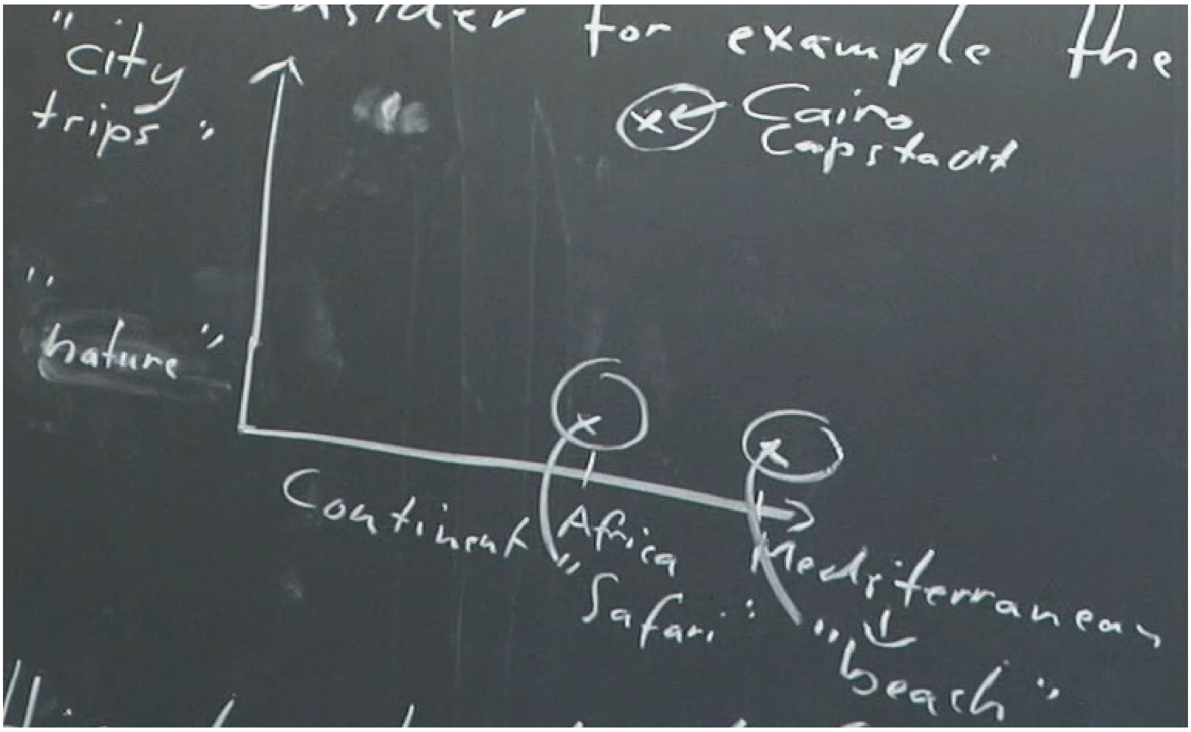
Noisy: unimportant

Plane: is the structure to preserve

###
In many real-world tasks, the structure to preserve, and the low-dim. manifold, is not so super-sharply defined.
Consider for example the manifold of tourist pictures.

Semantic Image Classification

From millions of pixels in millions of images to very few (<10) dimensions → rather extreme task.



In this class, I would like to see manifold learning as the task of finding a lower-dimensional representation of structure, essentially by ignoring the noise.

Why should we reduce the dimensionality?

- to avoid the "Curse of Dimensionality"

Curse of Dimensionality :
  Distance metrices lead to "wash out" in higher dimensional spaces, i.e. to lose their discriminative power.

To see this, consider N features $x_1, x_2, \dots x_N \in R^d$, where $0 \leq x_{i,k} \leq 1$,where k^{th} component of i^{th} Vector
To capture a fraction r of uniformly distributed features, wee need to consider a fraction $r * V$ of the Volume V of the feature space

This corresponds to a d-dimensional hypercube with edge length

$$e_d(r) = r^{\frac{1}{d}} =^d \sqrt{r} \tag{1}$$

For example, to find 1% of the features in 10-dim. space,

$$e_{10}(0.01) = (0.01)^{\frac{1}{10}} = 0.63 \tag{2}$$

to find10% in 10-dim. space:$e_{10}(0.1) = (0.1)^{\frac{1}{10}} = 0.8$

the **median** distance of the nearest neighbor to the origin of a d-dimensional space with N sampler is :

$$d(d, N) = (1 - (\frac{1}{2})^{\frac{1}{N}})^{\frac{1}{d}} \tag{3}$$

- d(10,5000) = 0.52

Most of the data points are close to the boundary

PCA (Principal Component Analyses):

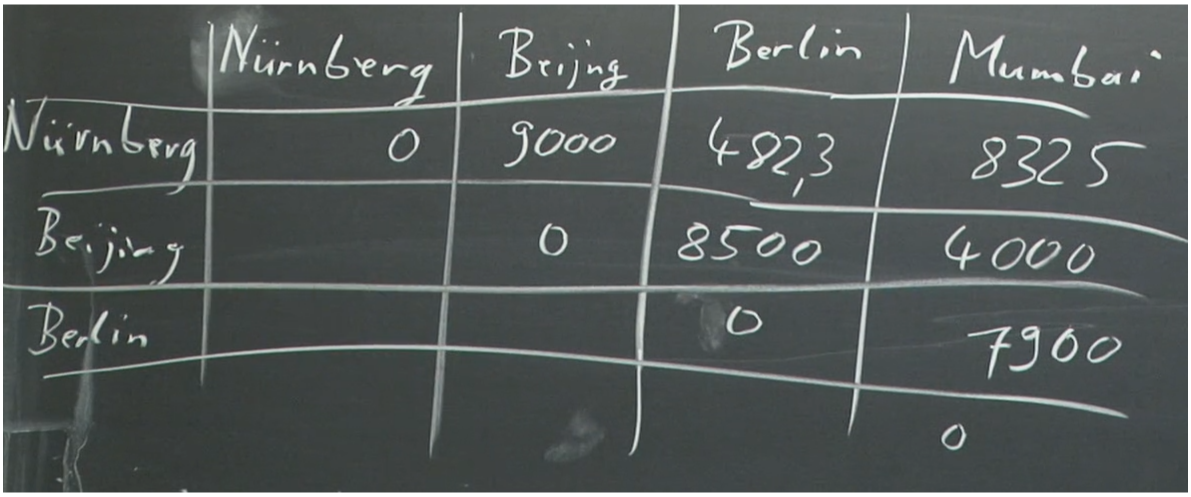
Find a linear mapping $\Phi : R^d -> R^{d1}, d^{'} << d$ that maximizes the variance in each dimension

- objective function: $J = \sum_{i,j=1}^{N} (\Phi * x_i - \Phi * x_j)^T * (\Phi * x_i - \Phi * x_j) + \lambda(\Phi^T \Phi - 1)$
- PCA requires an Eigenvalue decomposition of the covariance matrix PCA generates a low dim orthogonal basis.

Multidimensional Scaling (MDS)

Goal:

To compute a low-dimensional representation of data points where we only know the distances (or dissimilarities) between them.



- can we reconstruct a map from the distances?
  (- note that there are close links to kernel PCA!!!)
  see the elements of statistical learning, search for MDSE

Let $X = (x_1, \; x_2, x_3) \in R^{d \times N}$ denote the feature vectors (as usual)
Set $B = X^T X$
Let furthermore denote $D^2 = [d_{i,j}^2]_{i,j \in 1 \ldots N}$ where

$$d_{i,j}^2 = (x_i - x_j)^T (x_i - x_j) \tag{4}$$

(Euclidean distance, could be exchanged)
Task: given D^2, compute X

$$d_{i,j}^2 = (x_i - x_j)^T (x_i - x_j) = x_i^T x_i + x_j^T x_j - 2 x_i^T x_j \tag{5}$$

Assume that $x_i, \; \ldots, x_N$ are zero-mean. i.e. $\sum_{i=1}^{N^D} = 0$

The distance matrix is

$$D^2 = diag(X^T x) * l^T + l * diag(X^T X)^T - 2 * X^T X, \tag{6}$$

where $l = (1.///.l)^T \in R^N$
"Magic matrix"/Centering matrix $C = (I - \frac{1}{N} l * l^T)$

- Multiplying D^2 by the centering matrix from left & right and weighting the result by $-\frac{1}{2}$ yields:

$$-\frac{1}{2} C D^2 C = -\frac{1}{2}(I - \frac{1}{N} l * l^T) * (diag(X^T X) l^T) + l * diag(X^T X) - 2 * X^T X) * (I - \frac{1}{N} * l * l^T) \tag{7}$$

Term (1): = 0
Term (2): = 0
Term (3):
$-\frac{1}{2}(I - \frac{1}{N} l * l^T) * -2 * X^T X) * (I - \frac{1}{N} * l * l^T) = I * X^T - \frac{1}{N}(l * l^T) X^T * (X - \frac{1}{N} X * l * l^T) \rightarrow X^T X = B$

- Factorize B to obtain X via SBD:
  $B = U^T \Sigma V$
  B is symmetric $\Rightarrow U^T \Sigma^{\frac{1}{2}} * \Sigma^{\frac{1}{2}} * V$