

Advanced Learning for text and Graph data

M. VAZIRGIANNIS

<https://www.lix.polytechnique.fr/~mvazirg/>

DASCIM, LIX

<https://www.lix.polytechnique.fr/dascim/>

March 2015

Course objective

- The course aims to familiarize students with *machine learning* and *data mining* methods towards design and development of solutions for data sets (graph, text) that are characterized by complexity and large volume (*Bigdata*).

Course Syllabus

- **Data Preprocessing:** Feature selection, - dimensionality reduction (SVD, PCS, MDS, LDA)
 - **Learning in Graphs:**
 - graph generators (random, preferential attachment, stochastic block models, kronecker graphs)
 - graph kernels and classification,
 - clustering / community mining (louvain, modularity, degeneracy),
 - epidemics / diffusion, privacy
 - **Text Mining:** Feature extraction (measures), Indexing, Retrieval functions (tf, idf, BM25), Ad hoc retrieval, Filtering, classification
 - **Bigdata:** an introduction (Mapreduce), No SQL databases (Hbase), Spark, large graph management.
-

Suggested references

Machine Learning

- Bayesian Reasoning and Machine Learning, David Barber c 2010
- Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, ISBN-13: 978-0387310732

Text Web Mining

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. (<http://nlp.stanford.edu/IR-book/>)
- Web Data Mining, Exploring Hyperlinks, Contents, and Usage Data, Bing Liu, Second Edition, July 2011, <http://www.cs.uic.edu/~liub/WebMiningBook.html>

Graph Mining

- Practical Graph Mining With R, Nagiza F. Samatova, et.al., Chapman & Hall/CRC Data Mining and Knowledge Discovery Series (2013)
[http://www.csc.ncsu.edu/faculty/samatova/practical-graph-mining-with-R/
PracticalGraphMiningWithR.html](http://www.csc.ncsu.edu/faculty/samatova/practical-graph-mining-with-R/PracticalGraphMiningWithR.html)

Bigdata

- Frontiers in Massive Data Analysis, http://www.nap.edu/catalog.php?record_id=18374

Course Logistics

Total Duration: 18 hours (9 lectures, 9 labs)

Evaluation: Assignment

E-learning platform:

<http://moodle.lix.polytechnique.fr/moodle/course/view.php?id=5>

Guest access key: ALTEGRAD2015

For full access enrolment we **need your emails** please

- Go to: <http://www.lix.polytechnique.fr/dascim/teaching/>
 - Locate the line:
 - [Graph and Text Mining](#) – joint course for the masters MVA and Mathbigdata – students to follow fill their contact data [here](#)
 - **Fill the form linked at the word** [here](#)
-

Team



- Fragkiskos Malliaros
(graph/text mining labs)



- Apostolos Papadopoulos
(bigdata, spark)

Outline

Distance Measures

Data Exploration and Preprocessing

Dimensionality Reduction

Feature Selection

Distance Measures

Data mining techniques are based on similarity or distance measures between objects.

Similarity or distance between data points can be expressed as:

Explicit similarity measurement for each pair of objects

Similarity obtained indirectly based on vector of object attributes.

A distance $d(i, j)$ is a metric iff

1. $d(i, j) \geq 0$ for all i, j and $d(i, j) = 0$ iff $i = j$

2. $d(i, j) = d(j, i)$ for all i and j

3. $d(i, j) \leq d(i, k) + d(k, j)$ for all i, j and k

Distance

- Notation: n objects with p attributes

$$x(i) = (x_1(i), x_2(i), \dots x_p(i))$$

- Most common distance metric is *Euclidean* distance:

$$d_E(i, j) = (\sum (x_k(i) - x_k(j))^2)^{1/2}$$

- Makes sense in the case where the different measurements are proportional; each variable measured in the same units.
- If the measurements are different, say length and weight, it is not clear – need for standardization

Weighted Euclidean distance

Finally, if we have some idea of the relative importance of each variable, we can weight them:

$$d_E(i, j) = \left(\sum w_k (x_k(i) - x_k(j))^2 \right)^{1/2}$$

Other Distance Metrics

Minkowski or L_λ metric:

$$d_E(i, j) = \left(\sum_{k=1}^p (x_k(i) - x_k(j))^\lambda \right)^{1/\lambda}$$

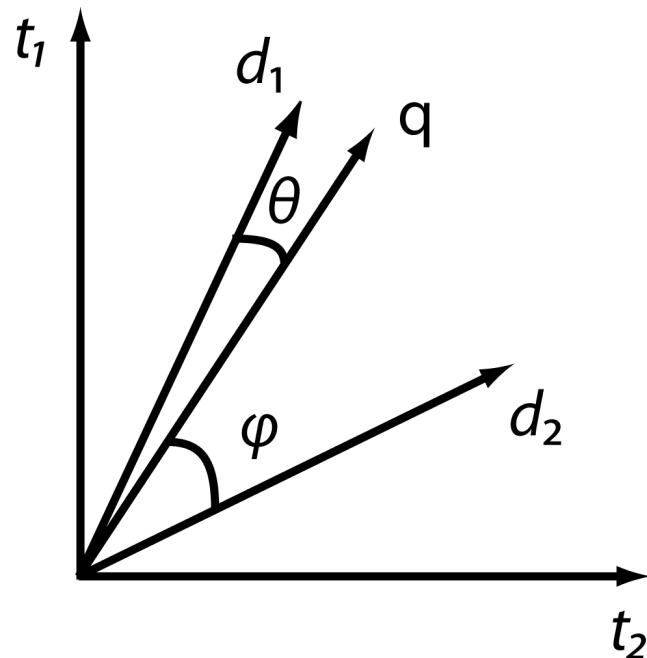
Manhattan, city block or L_1 metric:

$$L_1 \quad d_E(i, j) = \sum_{k=1}^p |x_k(i) - x_k(j)|$$

$$L_\infty \quad d_E(i, j) = \max_k |x_k(i) - x_k(j)|$$

Cosine based similarity

$$sim(q, d) = \frac{q \cdot d}{|q||d|} = \frac{\sum_{k=1}^p w_{k,q} \cdot w_{k,d}}{\sqrt{\sum_{k=1}^p w_{k,q}^2} \cdot \sqrt{\sum_{k=1}^p w_{k,d}^2}}$$



Distance metrics – Nominal values / text

Nominal variables

Number of matches divided by number of dimensions

A	A	B	B	C	B	B	C	C	A
A	B	B	A	C	B	B	C	C	C

7/10

- Edit (Levenshtein) distance

“kitten → sitten (substitution of "s" for "k")

sitten → sittin (substitution of "i" for "e")

sittin → sitting (insertion of "g" at the end)”

Exploratory Data Analysis

Methods not including formal statistical modeling and inference

- Detection of mistakes

- Checking of assumptions

- Preliminary selection of appropriate models

- Determining relationships among the explanatory variables, and

- Assessing the direction and rough size of relationships between explanatory and outcome variables (i.e. demographics – purchase)

Useful information about the data

- Min and Max values

- Mean Value

- Standard Deviation

- Number of instances per value (for nominal data)

- Percentage of missing values

- Data distribution

Data Quality

Individual measurements

Random noise in individual measurements

Variance (precision)

Bias

Random data entry errors

Noise in label assignment (e.g., class labels in medical data sets)

Systematic errors

E.g., all ages > 99 recorded as 99

More individuals aged 20, 30, 40, etc than expected

Missing information

Missing at random

Questions on a questionnaire that people randomly forget to fill in

Missing systematically

Questions that people don't want to answer

Patients who are too ill for a certain test

Data Quality

Ideal case = random sample from population of interest

Real case = often a biased sample of some sort

Key point: patterns or models from training data are valid on future (test) data
only if they are *generated from the same probability distribution*

Examples of non-randomly sampled data

Medical study where subjects are all students

Geographic dependencies

Temporal dependencies

Stratified samples

E.g., 50% healthy, 50% ill

Hidden systematic effects

E.g., market basket data the weekend of a large sale in the store

E.g., Web log data during finals week

Standardization

When variables are not commensurate standardize them dividing by the sample standard deviation. This makes them all equally important.

The estimate for the standard deviation of x_k :

$$\hat{\sigma}_k = \left(\frac{1}{n} \sum_{i=1}^n (x_k(i) - \bar{x}_k)^2 \right)^{\frac{1}{2}}$$

where \bar{x}_k is the sample mean:

$$\bar{x}_k = \frac{1}{n} \sum_{i=1}^n x_k(i)$$

Is standardization always a good idea?

hint: think of extremely skewed data and outliers, e.g., Bill Gates income.

Dependence among Variables

Covariance and correlation measure linear dependence

Assume variables X and Y and n objects taking on values $x(1), \dots, x(n)$ and $y(1), \dots, y(n)$.

Sample **covariance** of X and Y is:

$$\text{Cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x(i) - \bar{x})(y(i) - \bar{y})$$

The covariance is a measure of how X and Y vary together.

large and positive if large values of X are associated with large values of Y, and small X \Rightarrow small Y

Sample correlation coefficient

- Covariance depends on ranges of X and Y
- Standardize dividing with standard deviation

Sample **correlation** coefficient

$$\rho(X, Y) = \frac{\sum_{i=1}^n (x(i) - \bar{x})(y(i) - \bar{y})}{\left(\sum_{i=1}^n (x(i) - \bar{x})^2 \sum_{i=1}^n (y(i) - \bar{y})^2 \right)^{\frac{1}{2}}}$$

Outline

Distance Measures

Data Exploration and Preprocessing

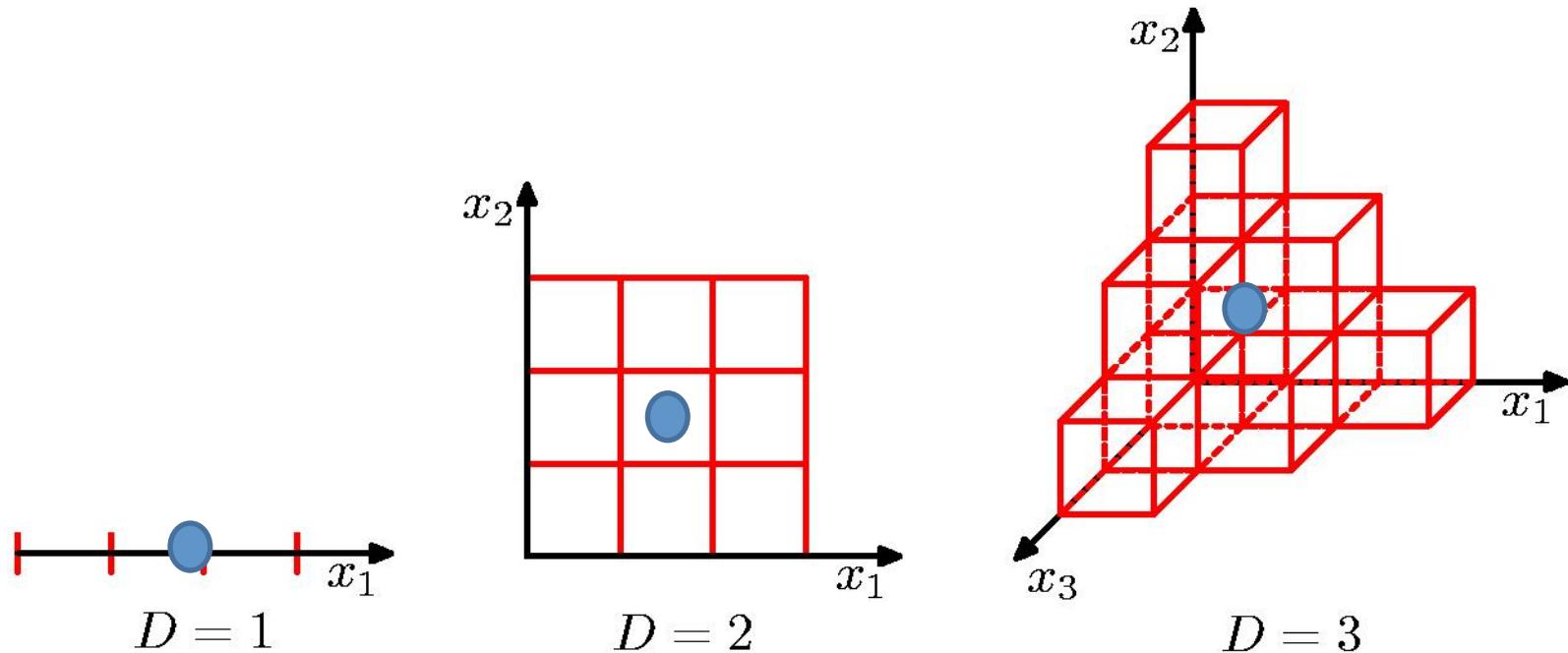
Dimensionality Reduction

Feature Selection

Curse of Dimensionality

- Some dimensions do not contribute to the data representation.
- Subsets of the dimensions may be highly correlated.
- Nearest neighbor is distorted in a high dimensional space
- Low dimension intuitions do not apply to high dimensions
- Empty space phenomenon

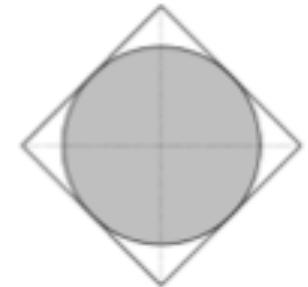
Curse of Dimensionality – k-NN



Assuming k-nn

- d^3 neighbors in a d dimensional space
- Distance computations are increasingly complex
- Empty space – same data embedded in a space of d^3 cells

Empty space phenomenon



Hyper sphere within a hyper rectangle

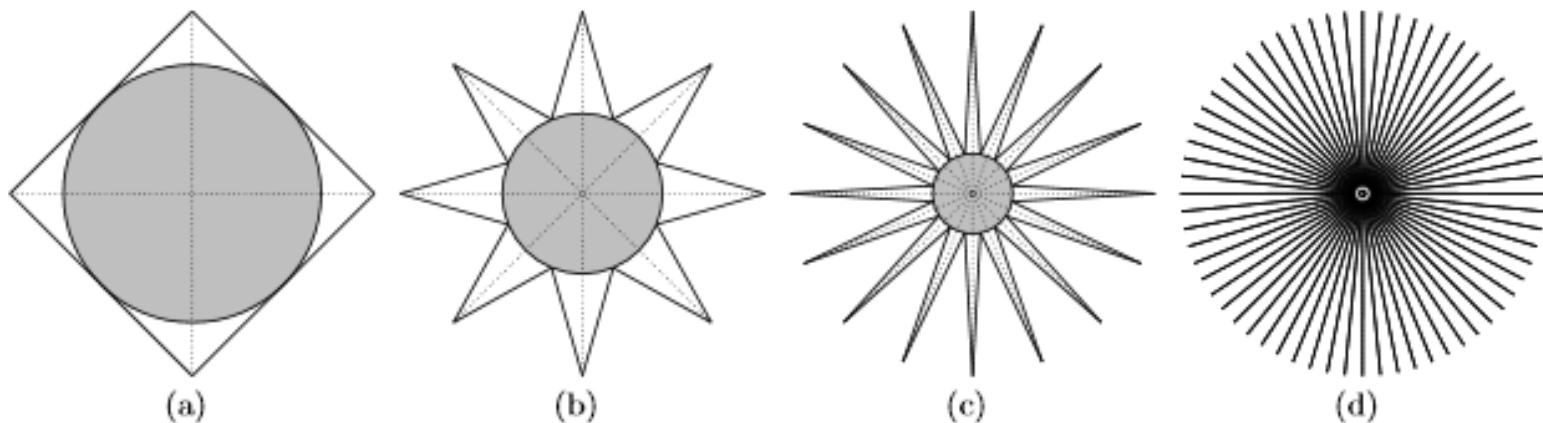
Respective volumes: $V(S) = \frac{2r^d \pi^{\frac{d}{2}}}{d\Gamma(\frac{d}{2})}$, $V(R) = (2r)^d$

The fraction of the sphere within the rectangle becomes insignificant with d increasing:

$$\lim_{d \rightarrow \infty} \left(\frac{\pi^{\frac{d}{2}}}{d 2^{d-1} \Gamma(\frac{d}{2})} \right) = 0$$

- the normal distribution in high dimensions
- longest/shortest distances converge.
- clustering becomes infeasible

Inscription of hyper sphere in a hypercube



The radius of the inscribed circle accurately reflects the difference between the volume of the hypercube and the inscribed hypersphere in d-dimensions.

Dim. Reduction – Linear Algorithms

Matrix Factorization methods

- Principal Components Analysis (PCA)
 - Singular Value Decomposition (SVD)
 - Multidimensional Scaling (MDS)
 - Linear Discriminant Analysis (LDA)
-

Low Rank Approximation

Data: $X = \{x_i \in R^{mxn} | x_i \text{ columns of } X\}$

Goal: approximate $X = UV^T$,

$U \in R^{mxr}$, $V \in R^{nxr}$, , $r \ll n$

- each data vector x_i : $x_i \sim Uv_i^T$, v_i is the i-th column of V .

Geometric interpretation:

- each data vector $x_i \in R^m$, $x_i \sim Uv_i^T$, is approximated by its projection to an r-dimensional space spanned by the column vectors of U
- $Y = UV^T$ the approximation matrix, max rank r

Frobenius distance

$$|X|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2} = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2}$$

Minimizing the frobenius distance can be considered as likelihood estimation

SOME CONTRIBUTIONS TO DIMENSIONALITY REDUCTION, Wei Tong, Ph.D. thesis,
2010, Michigan State University

<http://www.ece.uprm.edu/~domingo/teaching/ciic8996/SOME%20CONTRIBUTIONS%20TO%20DIMENSIONALITY%20REDUCTION.pdf>

Dim. Reduction–Eigenvectors

A $n \times n$ matrix

- eigenvalues λ : $|A - \lambda I| = 0$
 - Eigenvectors x : $Ax = \lambda x$
 - Matrix rank: # linearly independent rows or columns
 - A real symmetric table A $n \times n$ can be expressed as: $A = U \Lambda U^T$
 - U 's columns are A 's eigenvectors
 - Λ 's diagonal contains A 's eigenvalues
 - $A = U \Lambda U^T = \lambda_1 x_1 x_1^T + \lambda_2 x_2 x_2^T + \dots + \lambda_n x_n x_n^T$
 - $x_1 x_1^T$ represents projection via x_1 (λ_i eigenvalue, x_i eigenvector)
 - XX^T vs. $X^T X$
-

Singular Value Decomposition (SVD)

Eigen values and eigenvectors decomposition is applied to square matrices. For non square matrices we apply **Singular Value Decomposition.**

Let \mathbf{X} a $m \times n$ table, $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$

\mathbf{U} : orthogonal $m \times m$, its columns are the eigenvectors of $\mathbf{X}\mathbf{X}^T$.

\mathbf{U}, \mathbf{V} define orthogonal basis: $\mathbf{U}^T\mathbf{U} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$

Σ : $m \times n$ contains A's singular values (square roots of $\mathbf{X}\mathbf{X}^T$ eigenvalues)

\mathbf{V} : $n \times n$, its columns are the eigenvectors of $\mathbf{X}^T\mathbf{X}$

Singular Value Decomposition (SVD) - I

PROOF:

$$X = U\Sigma V^T, X^T = V\Sigma^T U^T \rightarrow XX^T = U\Sigma(V^T V)\Sigma^T U^T \rightarrow XX^T = U\Sigma\Sigma^T U^T$$

Similarly: $\rightarrow X^T X = U\Sigma(V^T V)\Sigma^T U^T$ therefore $X^T X = V\Sigma^T\Sigma U^T$

Hence: U : eigenvectors of XX^T , V : eigenvectors of $X^T X$ and Σ sqrt of MM^T (or $M^T M$) eigenvalues

Let X a $m \times n$ table, $X = U\Sigma V^T$ then an r rank approximation of X is:

$$Y = U_{m \times r} \text{diag}(\sigma_1, \dots, \sigma_r) {V_{n \times r}}^T$$

Singular Value Decomposition (SVD) - II

Matrix approximation

$$X_k = U_k \Sigma_k V_k^T$$

The best rank r approximation Y' of a matrix X . (minimizing the Frobenius norm)

$$\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 = \text{trace}(AA^H) = \sum_{i=1}^{\min\{m, n\}} \sigma_i^2$$

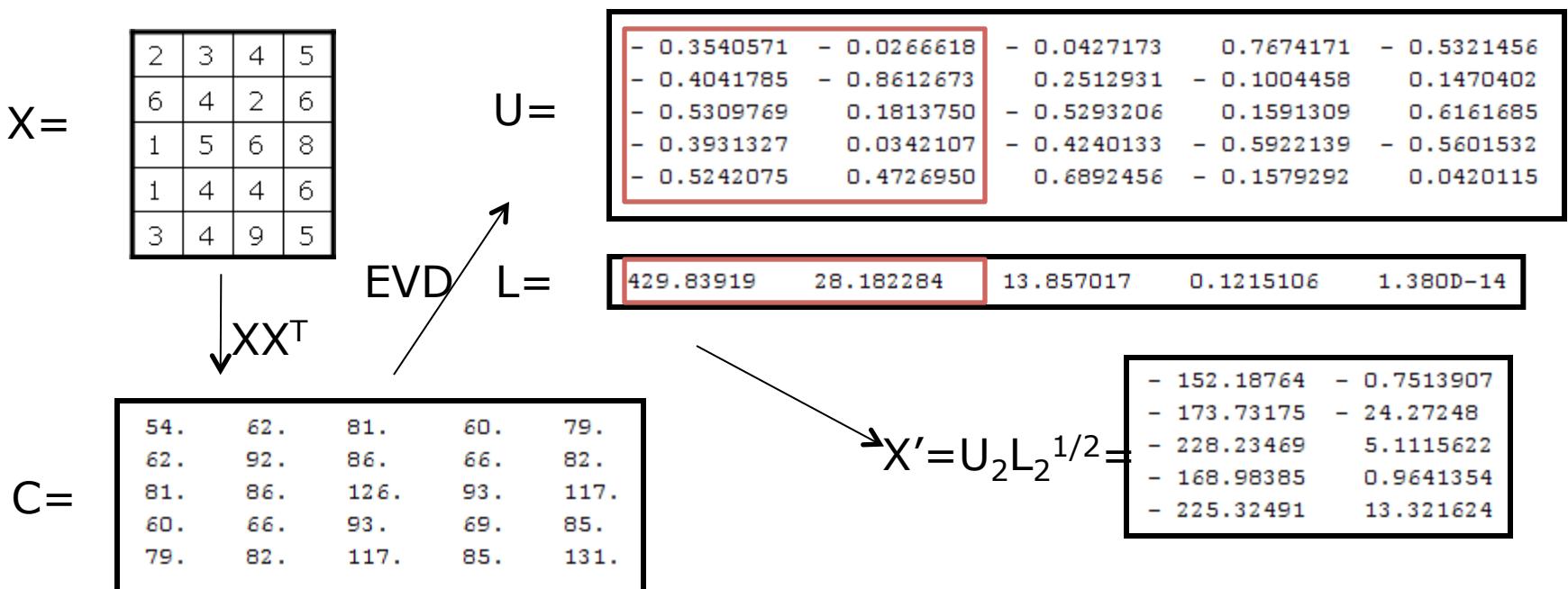
where A^H transpose of A , σ_i are the singular values of A , and the trace function is used.

Multidimensional Scaling (MDS)

- Decomposition of the data similarity matrix: XX^T
- Aim to minimize the stress:
 - Stress = $\sum(d_{ij} - d'_{ij})^2 / \sum d_{ij}^2$
 - Complexity $O(N^3)$ (N :number of vectors)
- Result:
 - A new depiction of the data in a lower dimensional space.
- Implement usually by:
 - Eigen decomposition of the inner product matrix
 - projection on the k eigenvectors corresponding to the k largest eigenvalues.

Multidimensional Scaling

- Data is given as rows in X
 - $C = XX^T$ (inner product of x_i with x_j)
 - Eigen decomposition of $C' = ULU^{-1}$
 - Eventually $X' = U_k L_k^{1/2}$, where k is the projection dimension



Principal Components Analysis

The main concept behind *Principal Components Analysis* is dimensionality reduction, maintaining as much as possible data's variance.

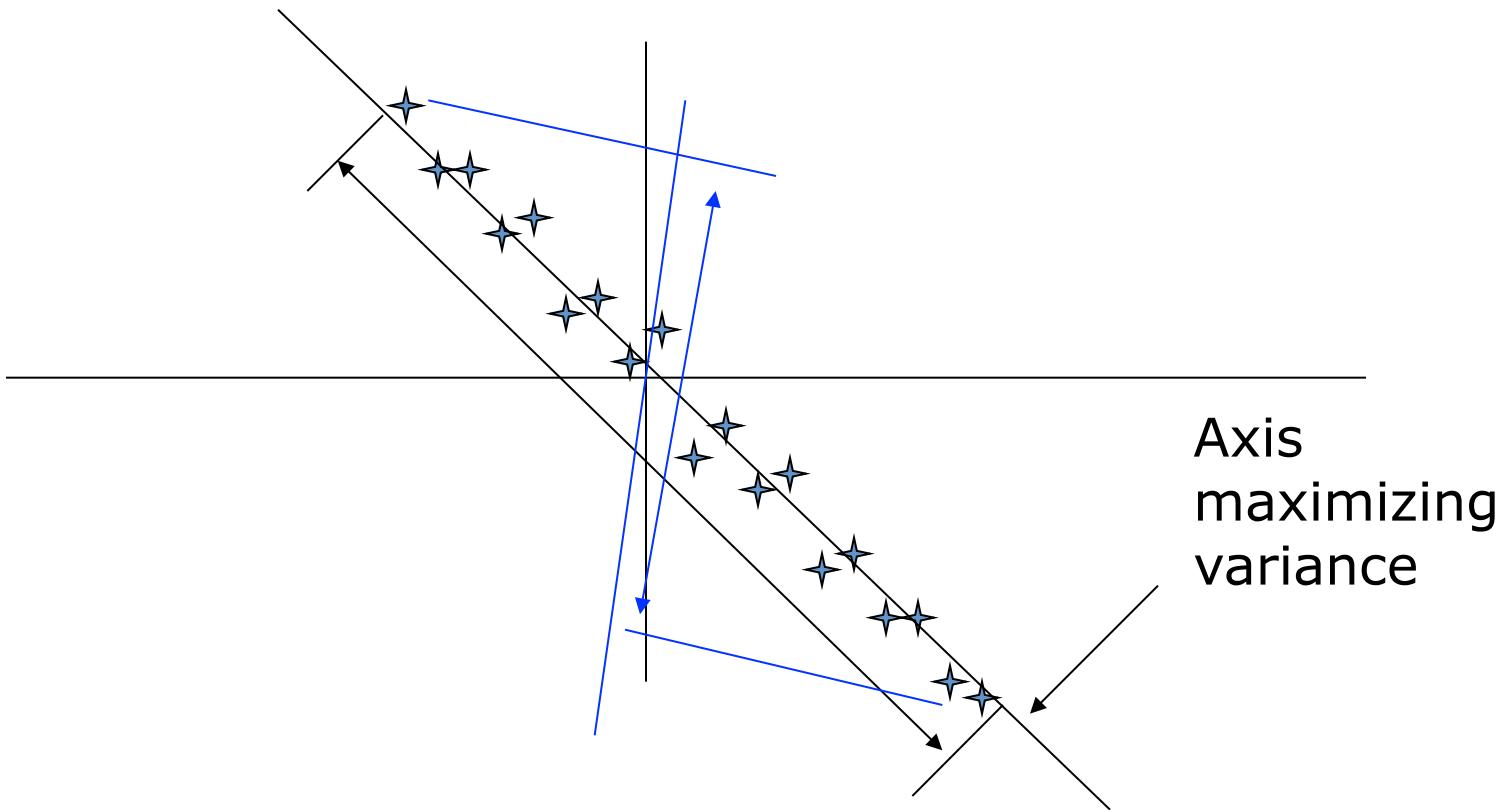
variance: $V(X) = \sigma^2 = E[(X - \mu)^2]$

Let N objects, with mean value, m , it is approximated as:

$$\frac{1}{N} \sum_{i=1}^N (x_i - m)^2,$$

Sample of N objects with unknown mean value: $\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$,

Dimensionality reduction based on variance maintenance



Covariance Matrix

Let Matrix $X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$ where X_i vectors

covariance matrix Σ is the matrix whose (i, j) entry is the covariance

$$\Sigma = \begin{bmatrix} E[(X_1 - \mu_1)(X_1 - \mu_1)] & E[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & E[(X_1 - \mu_1)(X_n - \mu_n)] \\ E[(X_2 - \mu_2)(X_1 - \mu_1)] & E[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & E[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n - \mu_n)(X_1 - \mu_1)] & E[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & E[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}.$$

Also: $\text{cov}(X) = X^T X'$, where $X' = X - M$

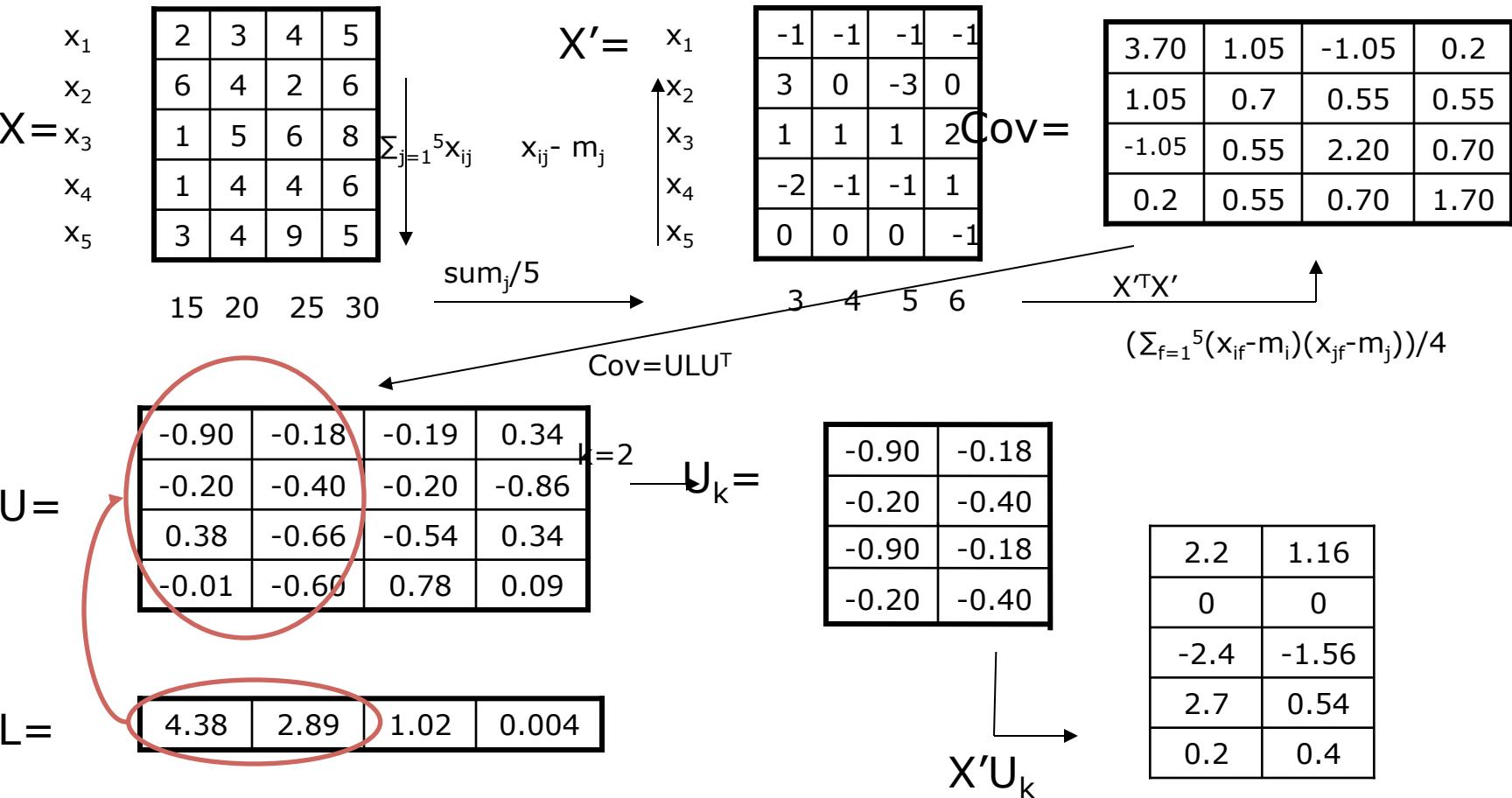
Principal Components Analysis (PCA)

- **The basic idea of PCA is the maximization of the covariance.**
 - Variance: Depicts the maximum deviation of a random variable from the mean.
 - $\sigma^2 = \sum_{i=1}^n ((x_i - \mu_i)^2 / n)$
- **Method:**
 - Assumption: Data is described by p variables and contained as rows in matrix X_{pxn}
 - We subtract mean values from columns. $X' = (X - M)$
 - Calculate covariance matrix $W = X'^T X'$

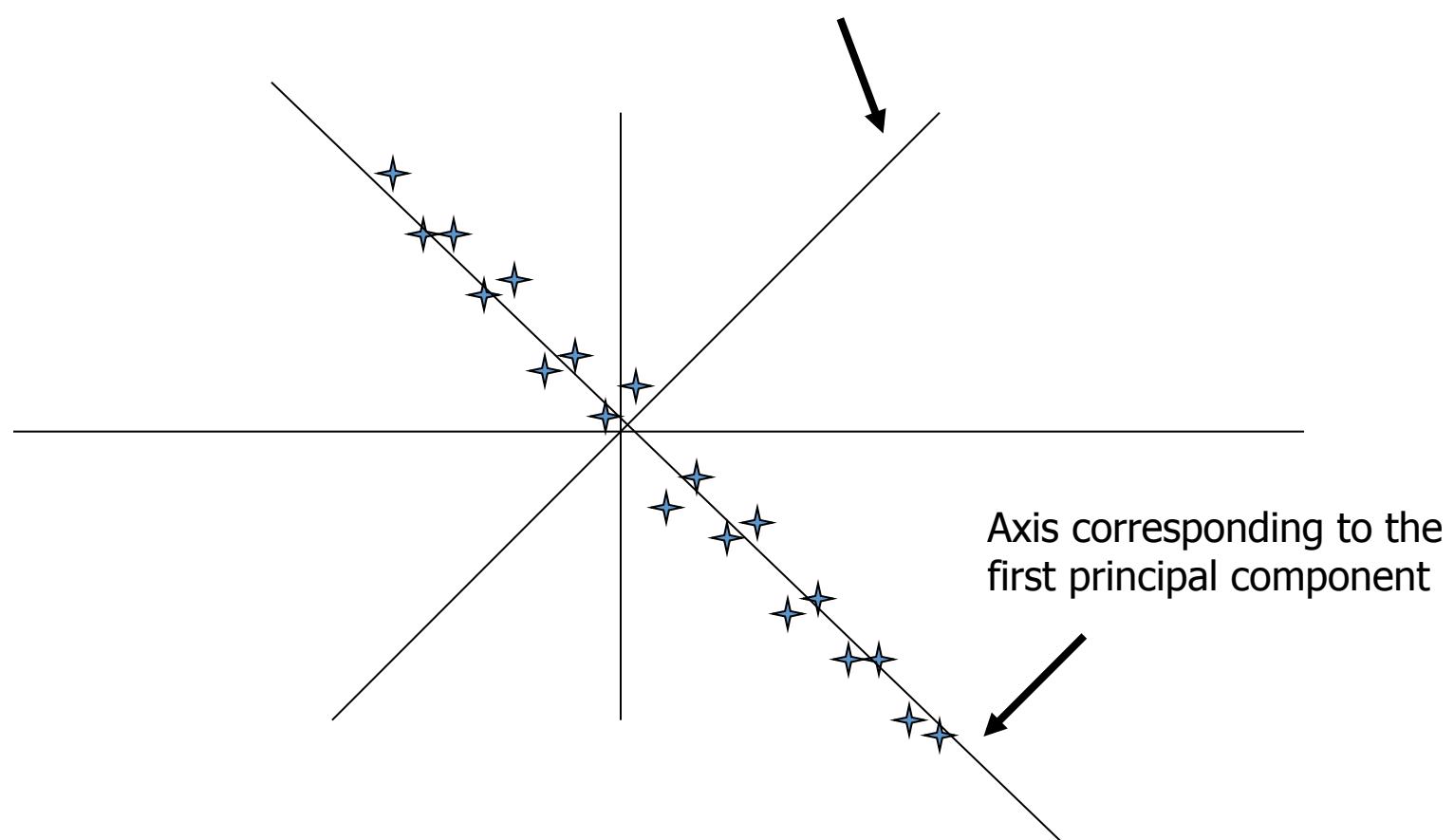
Principal Components Analysis (PCA) – (2)

- **Calculation of covariance matrix (W)**
 - A matrix nxn, in each cell of which (i,j) we have the covariance of X_i, X_j .
- **Calculate eigenvalues and eigenvectors of W ($X, D = UAU^T$)**
- **Retain k largest eigenvalues and corresponding eigenvectors**
 - k is an input parameter
 - There is an input parameter and k is calculate by
$$\sum_{j=k+1}^p \lambda_j / \sum_{j=1}^p \lambda_j > 85\%$$
- **Projection : $A'X_k$**

Principal Components Analysis



PCA, example



PCA Synopsis & Applications

- Preprocessing step preceding the application of data mining algorithms (such as clustering).
 - Data Visualization & Noise reduction.
 - It is a dimensionality reduction method
 - Nominal complexity $O(np^2+p^3)$
 - n: number of data points
 - p: number of initial space dimensions
 - The new space maintains sufficiently the data variance.
-

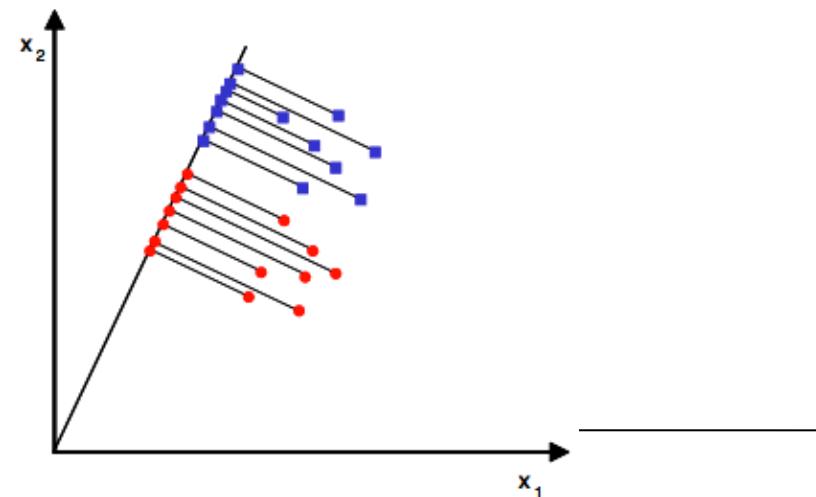
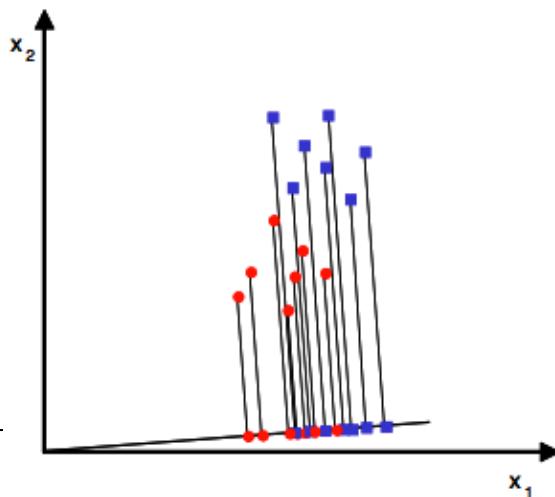
Linear discriminant analysis

Linear discriminant analysis, two classes

Linear discriminant analysis, C classes

LDA – two classes

- LDA aims to project data in a lower dimensional space that preserves as much of the class discriminatory information as possible
- Assume $X = (x_1, x_2, \dots, x_n)$ d-dimensional that belong to either of the classes C1 and C2.
- We search for $y = w^T X$ that best separates the data of C1 and C2.

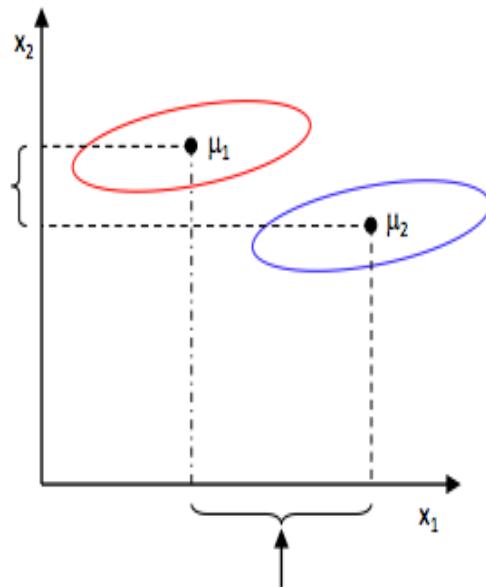


LDA – two classes

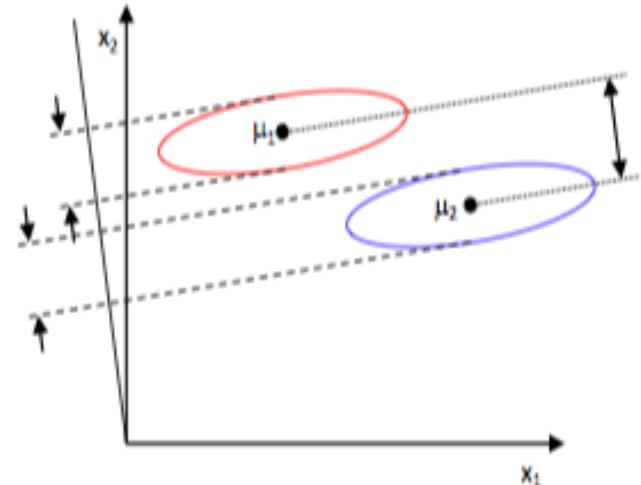
The projection should

- maximize the distance of the class centers
- minimize the in class variance

This axis yields better class separability →



This axis has a larger distance between means



LDA two classes – Fischer’s criterion

Distance between projected classes centers $\hat{\mu}_1 \hat{\mu}_2$

$$|\hat{\mu}_1 - \hat{\mu}_2| = \frac{1}{|C_1|} \sum_{C_1} y_i^{C_1} - \frac{1}{|C_2|} \sum_{C_2} y_i^{C_2} = \frac{1}{|C_1|} \sum_{C_1} w^T x_i^{C_1} - \frac{1}{|C_2|} \sum_{C_2} w^T x_i^{C_2}$$

Within class variance $\hat{S}_i = \sum_{y \in C_i} (y - \hat{\mu}_i)^2$

Total *within class scatter* for the projected data

$$\hat{S}_1 + \hat{S}_2$$

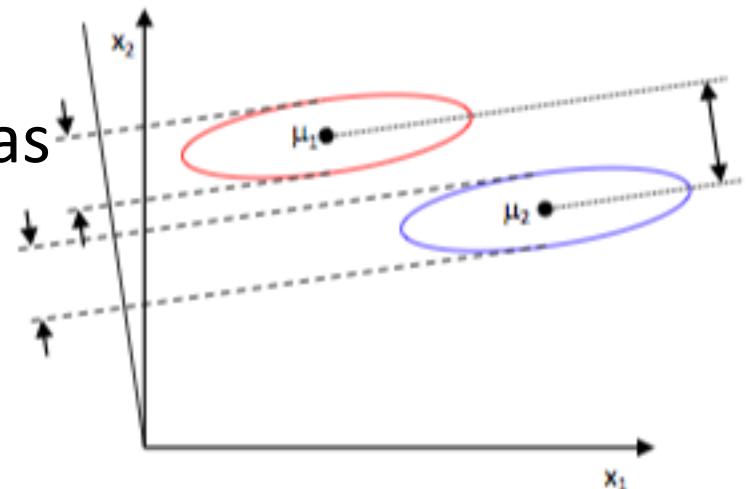
Fischer's linear discriminant

linear function of $y = w^T X$

maximizing:
$$J(w) = \frac{|\hat{\mu}_1 - \hat{\mu}_2|^2}{\hat{S}_1 + \hat{S}_2}$$

Therefore, searching for a projection where same class points are

- projected very close to each
- projected means are as farther as possible



Optimization – within class scatter

Within class scatter matrix in feature space x

$$S_w = S_1 + S_2$$

Where $S_i = \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T$

Within class scatter matrix in projected space y

$$\hat{S}_1^2 + \hat{S}_2^2 = w^T S_w w$$

Where $\hat{S}_i = w^T S_i w$

between class scatter in projected space y

$$|\hat{\mu}_1 - \hat{\mu}_2|^2 = w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = w^T S_B w$$

Fishers Criterion optimization

$$J(W) = \frac{w^T S_B w}{w^T S_W w}$$

To find maximum

$$\frac{d}{dw} J(W) = 0 \implies S_w^{-1} S_B w - Jw = 0$$

Solving the generalized eigenvalue problem:

$$S_w^{-1} S_B w = Jw$$

We find:

$$w^* = \operatorname{argmax} \left[\frac{w^T S_B w}{w^T S_W w} \right] = S_w^{-1} (\mu_1 - \mu_2)$$

An example[1]

- Compute the LDA projection for the following 2D dataset

$$X_1 = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$$

$$X_2 = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$$

SOLUTION (by hand)

- The class statistics are

$$S_1 = \begin{bmatrix} .8 & -.4 \\ 2.64 & \end{bmatrix} \quad S_2 = \begin{bmatrix} 1.84 & -.04 \\ & 2.64 \end{bmatrix}$$

$$\mu_1 = [3.0 \ 3.6]^T; \quad \mu_2 = [8.4 \ 7.6]^T$$

- The within- and between-class scatter are

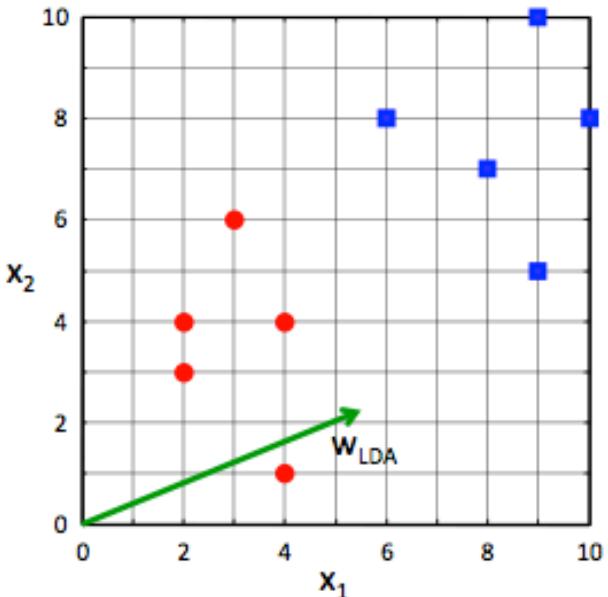
$$S_B = \begin{bmatrix} 29.16 & 21.6 \\ 16.0 & \end{bmatrix} \quad S_W = \begin{bmatrix} 2.64 & -.44 \\ & 5.28 \end{bmatrix}$$

- The LDA projection is then obtained as the solution of the generalized eigenvalue problem

$$S_W^{-1} S_B v = \lambda v \Rightarrow |S_W^{-1} S_B - \lambda I| = 0 \Rightarrow \begin{vmatrix} 11.89 - \lambda & 8.81 \\ 5.08 & 3.76 - \lambda \end{vmatrix} = 0 \Rightarrow \lambda = 15.65$$
$$\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 15.65 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 5.08 \\ .91 \end{bmatrix} \begin{bmatrix} 8.81 \\ 3.76 - \lambda \end{bmatrix} = \begin{bmatrix} .91 \\ .39 \end{bmatrix}$$

- Or directly by

$$w^* = S_W^{-1}(\mu_1 - \mu_2) = [-.91 \ - .39]^T$$



LDA C classes [1]

Fisher's LDA generalizes gracefully for C-class problems

- Instead of one projection y , we will now seek $(C - 1)$ projections $[y_1, y_2, \dots, y_{C-1}]$ by means of $(C - 1)$ projection vectors w_i arranged by columns into a projection matrix $W = [w_1 | w_2 | \dots | w_{C-1}]$:

$$y_i = w_i^T x \Rightarrow y = W^T x$$

Derivation

- The within-class scatter generalizes as

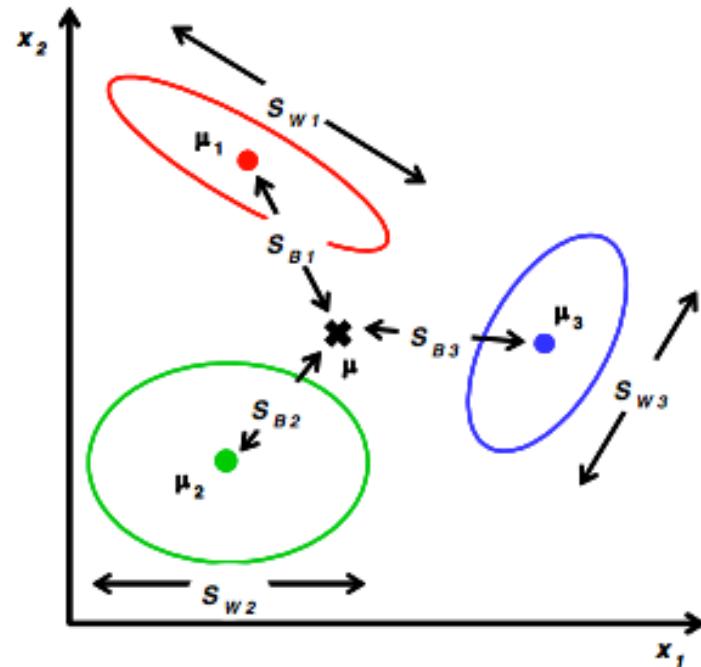
$$S_W = \sum_{i=1}^C S_i$$

- where $S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$
and $\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$

- And the between-class scatter becomes

$$S_B = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

- where $\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{i=1}^C N_i \mu_i$



- Matrix $S_T = S_B + S_W$ is called the total scatter

LDA C classes

mean vector and scatter matrices for the projected samples

$$\begin{aligned}\tilde{\mu}_i &= \frac{1}{N_i} \sum_{y \in \omega_i} y & \tilde{S}_W &= \sum_{i=1}^C \sum_{y \in \omega_i} (y - \tilde{\mu}_i)(y - \tilde{\mu}_i)^T \\ \tilde{\mu} &= \frac{1}{N} \sum_{\forall y} y & \tilde{S}_B &= \sum_{i=1}^C N_i (\tilde{\mu}_i - \tilde{\mu})(\tilde{\mu}_i - \tilde{\mu})^T\end{aligned}$$

Thus generalizing the two class problem:

$$\begin{aligned}\tilde{S}_W &= W^T S_W W \\ \tilde{S}_B &= W^T S_B W\end{aligned}$$

Searching for a projection maximizing the ratio of *between-class/within-class scatter*.

projection is no longer a scalar ($C - 1$ dimensions), use the determinant of the scatter matrices to obtain a scalar objective function

$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \frac{|W^T S_B W|}{|W^T S_W W|}$$

LDA C classes

seek the projection matrix W^* maximizing this ratio

optimal projection matrix W^* is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the following generalized eigenvalue problem

$$W^* = [w_1^* | w_2^* | \dots | w_{C-1}^*] = \arg \max \frac{|W^T S_B W|}{|W^T S_W W|} \Rightarrow (S_B - \lambda_i S_W) w_i^* = 0$$

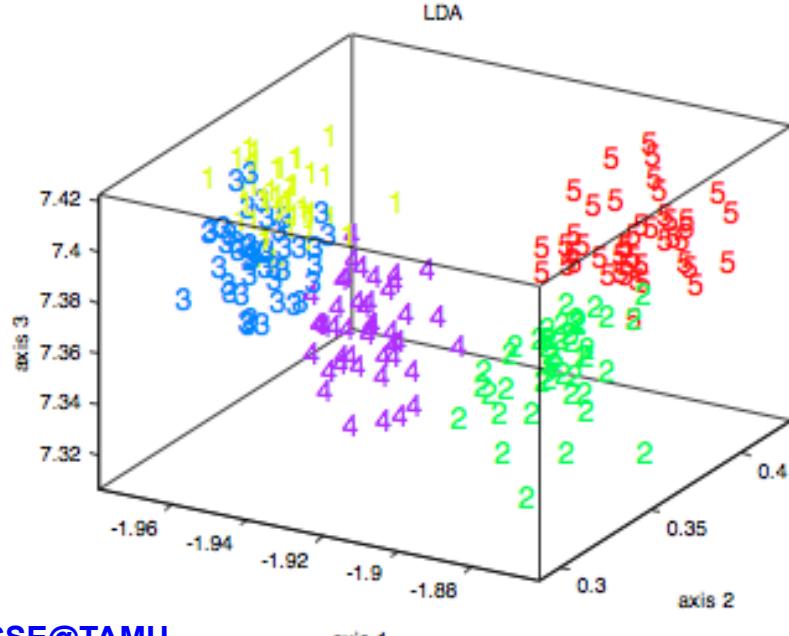
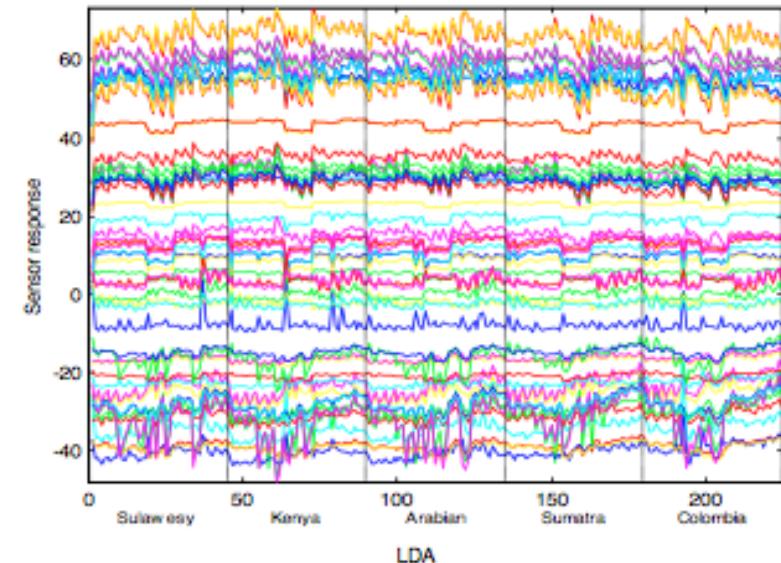
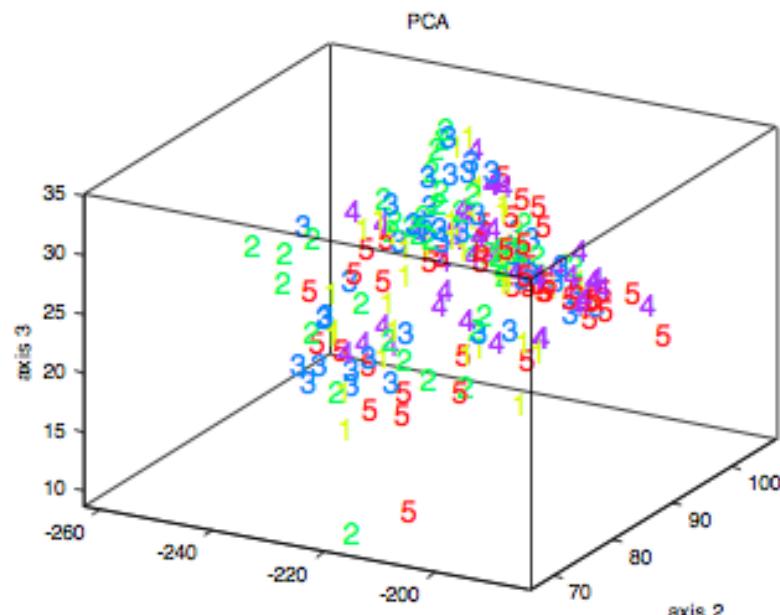
LDA vs. PCA

This example illustrates the performance of PCA and LDA on an odor recognition problem

- Five types of coffee beans were presented to an array of gas sensors
- For each coffee type, 45 “sniffs” were performed and the response of the gas sensor array was processed in order to obtain a 60-dimensional feature vector

Results

- From the 3D scatter plots it is clear that LDA outperforms PCA in terms of class discrimination
- This is one example where the discriminatory information is not aligned with the direction of maximum variance



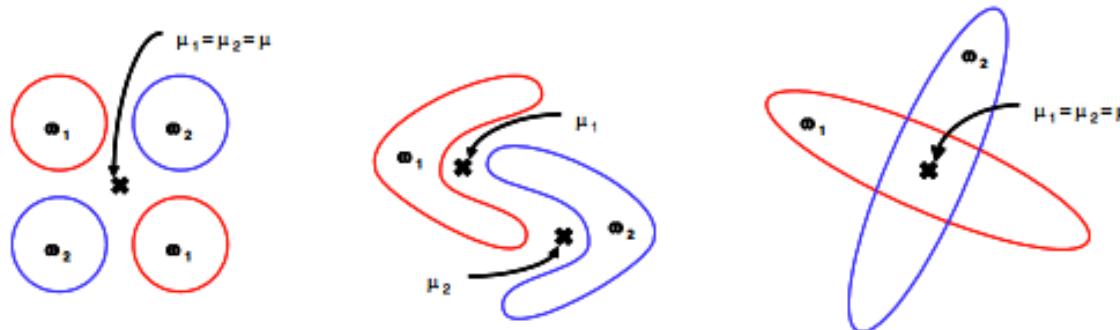
Limitations of LDA

LDA produces at most $C - 1$ feature projections

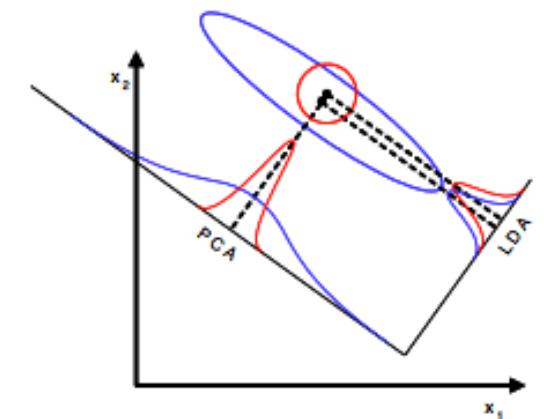
- If the classification error estimates establish that more features are needed, some other method must be employed to provide those additional features

LDA is a parametric method (it assumes unimodal Gaussian likelihoods)

- If the distributions are significantly non-Gaussian, the LDA projections may not preserve complex structure in the data needed for classification



LDA will also fail if discriminatory information is not in the mean but in the variance of the data



Feature selection

Select the “best” features (subset of the original one)

Filter methods:

rank the features individually according to some criteria (information gain, χ^2 , etc.) and take the top-k or eliminate redundant features (correlation)

Wrapper methods:

evaluate each subset using some data mining algorithm; use heuristics for the exploration of the subset space (forward/backward search, etc.)

Embedded methods:

feature selection is part of the data mining algorithm

Filter methods - Information Gain (IG)

For a random variable X (class) its entropy

$$H = - \sum_{i=1}^c P(x_i) \times \log(P(x_i)) , c \text{ classes}$$

“High Entropy”: X is from a uniform distribution – lack on information

“Low Entropy”: X is from varied (peaks and valleys) distribution – rich in information content

Let variable A (feature), $\text{IG}(X, A)$ represents the reduction in entropy (~ gain in Information) of X achieved by learning the state of A:

$$\text{IG}(X, A) = H(X) - H(X|A)$$

Filter methods - Chi-squared test (χ^2)

Test of independence between a class X and a feature A

$$\chi^2(A) = \sum_{i=1}^v \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}} , v \text{ values, } c \text{ classes}$$

O_{ij} : observed frequency of class j for feature A (value i)

E_{ij} : the expected frequency

(# of samples with value i) x (# of samples with class j)

$$E_{ij} = \frac{\text{(# of samples with value i) x (# of samples with class j)}}{\text{# of samples in total}}$$

Finding the k best variables

Find the subset of k variables that predicts best:

This is a generic problem when p is large
(arises with all types of models, not just linear regression)

Models with different complexity..

- p models with a **single** variable
- $p(p-1)/2$ models with **2 variables**, etc
- ...
- 2^p possible models in total

Best k set is not the same as the *best k individual* variables

What does “best” mean here?

Search Problem

How can we search over all 2^p possible models?
exhaustive search is clearly infeasible

Heuristic search is used to search over model space:

- Forward search (greedy)
- Backward search (greedy)
- Branch and bound techniques

Variable selection problem in several data mining algorithms

- Outer loop that searches over variable combinations
- Inner loop that evaluates each combination

Forward model selection

- Start with the variable the lowest **p-value** (i.e. value with the highest evidence for rejecting the null hypothesis)
- add in each repetition the variable with the *highest F-test* value:

$$F = \frac{\frac{RSS_1 - RSS_2}{\rho_2 - \rho_1}}{\frac{RSS_2}{n - \rho_2}}$$

- Assume two models ρ_2, ρ_1 with $|\rho_2| > |\rho_1|$
- Repeat until $F\text{-value} < \text{threshold}_f$ (or $p\text{-value} > \text{threshold}_p$)
- RSS_i the residual sum of squares - the error induced by the model:

$$F = \sum_1^n (y_i - f(x_i))^2$$

with y_i real value and $f(x_i)$ predicted by models containing p_i .

Backward Elimination

- start with the full model
- drop the predictor that produces the smallest F value (or highest p-value)
- Continue until $F\text{-value} < \text{threshold}_f$
(or $p\text{-value} > \text{threshold}_p$)
- Sometimes constraint $N > p$

Results Evaluation metrics

Confusion matrix:

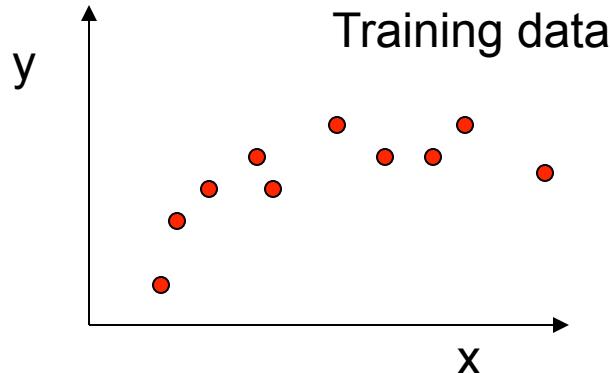
		Actual class	
		True Positive	False Positive
Predicted class	True Negative	False Negative	

$$\text{Precision} \quad \frac{TP}{TP + FP}$$

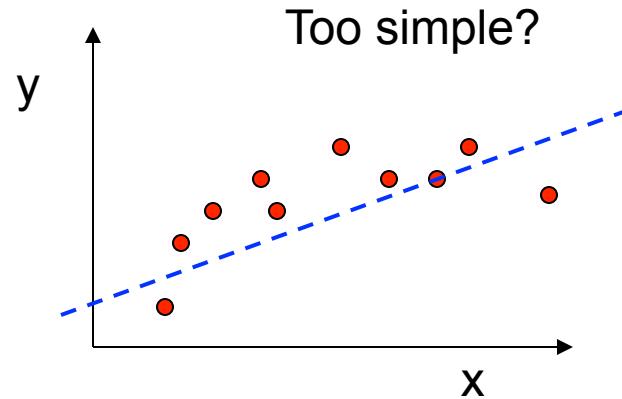
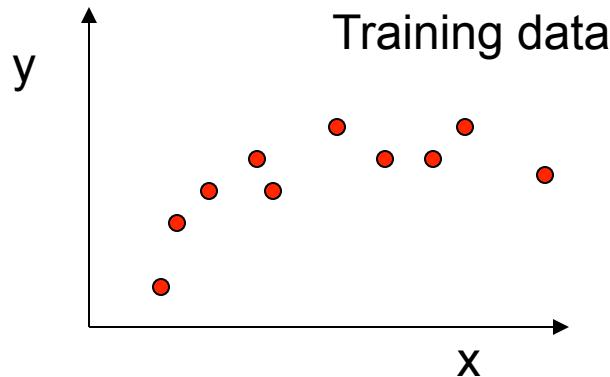
$$\text{Recall} \quad \frac{TP}{TP + FN}$$

$$\text{Matching coefficient} \quad \frac{TP + TN}{TP + TN + FP + FN}$$

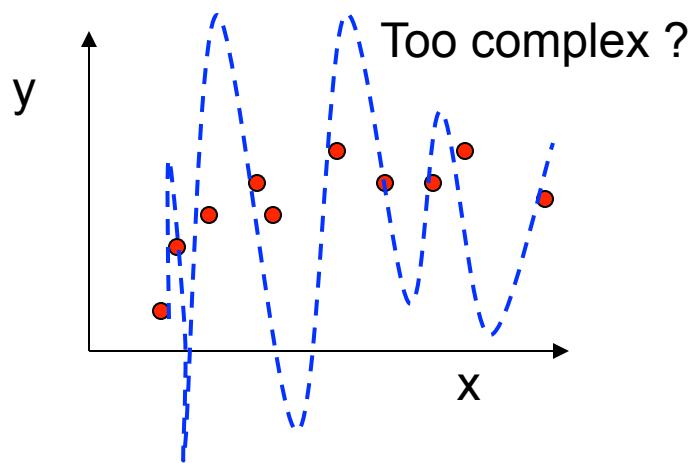
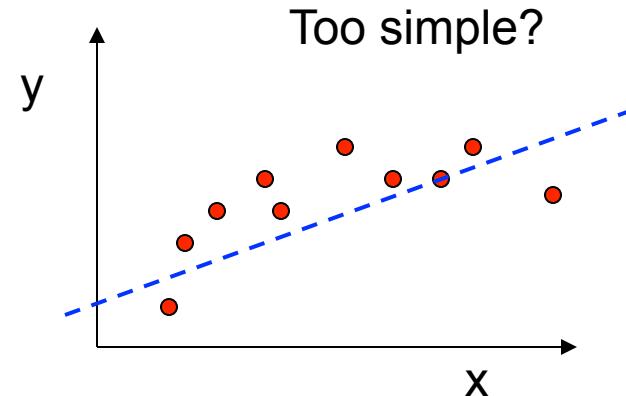
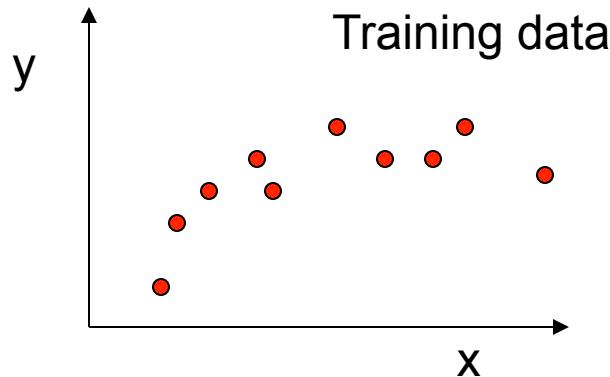
Complexity versus Goodness of Fit



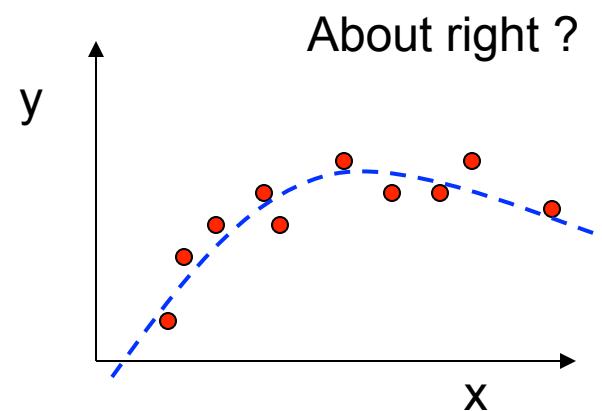
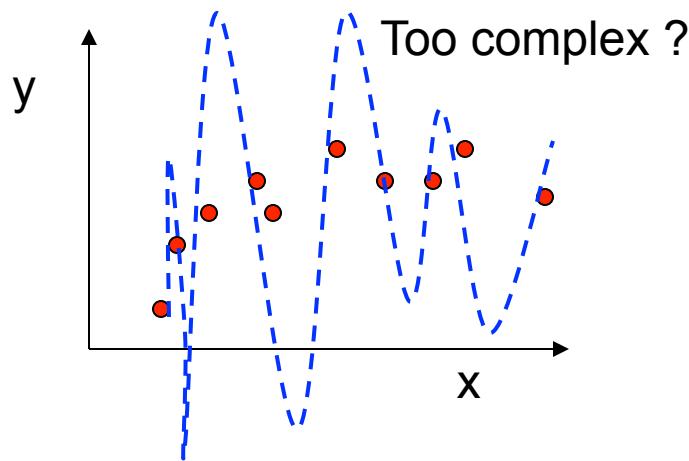
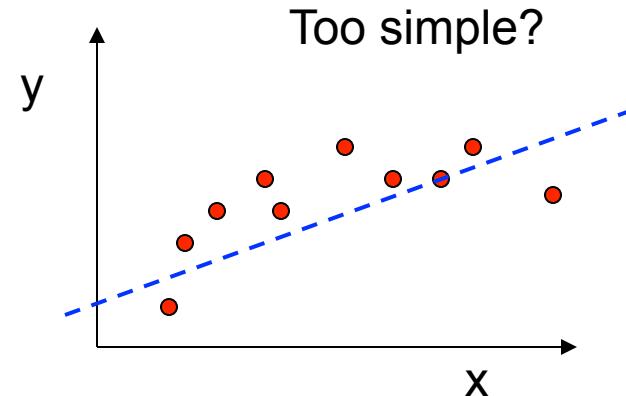
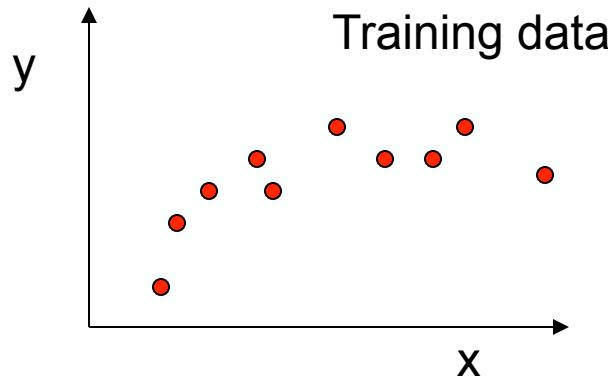
Complexity versus Goodness of Fit



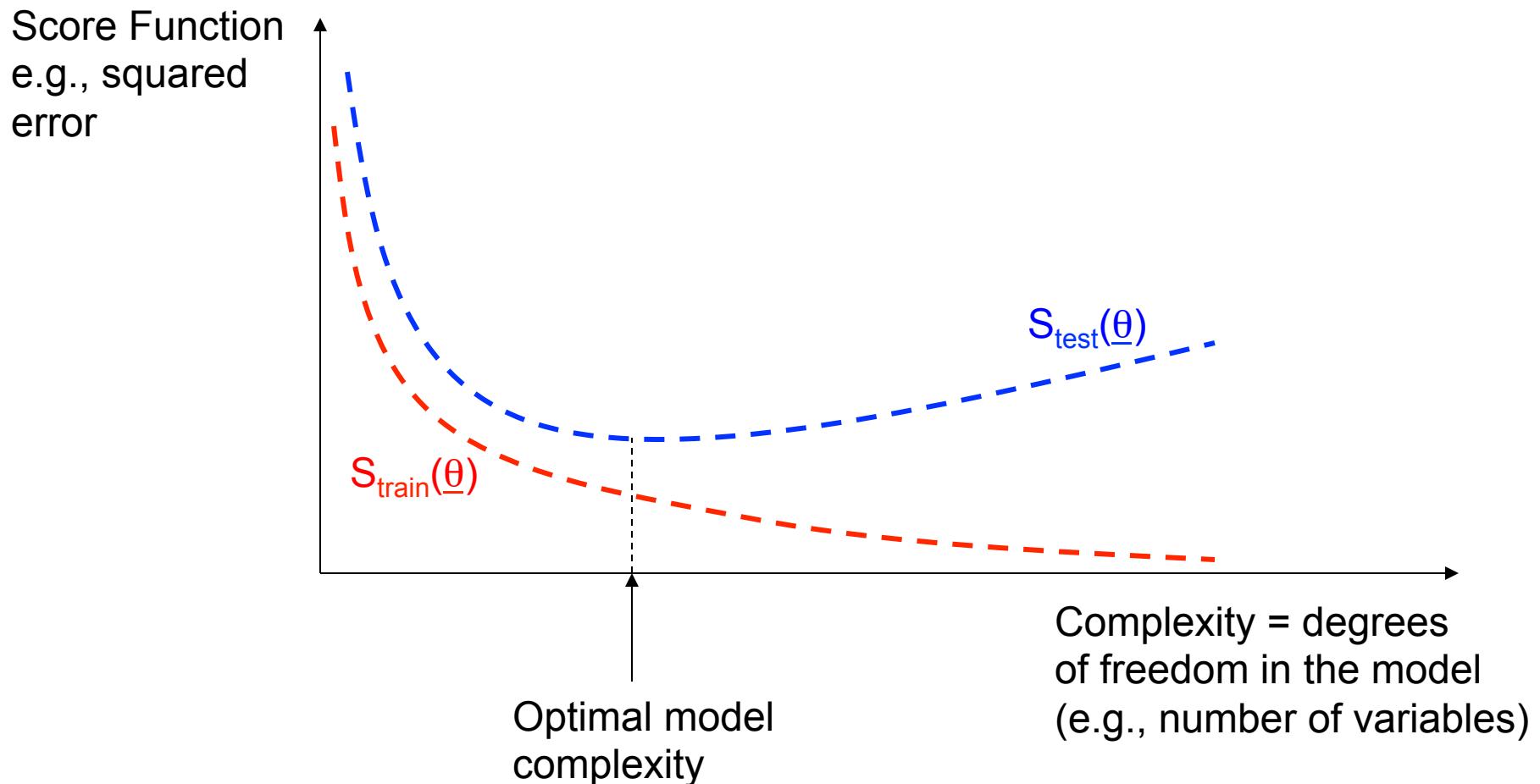
Complexity versus Goodness of Fit



Complexity versus Goodness of Fit



Complexity and Generalization



Useful References

- **Principles of Data Mining**, David J. Hand, Heikki Mannila and Padhraic Smyth
MIT Press 2001
- **T. Hastie, R. Tibshirani, and J. Friedman**, **Elements of Statistical Learning**,
Springer Verlag, 2001
- **Dash, Manoranjan, and Huan Liu**. "Feature selection for classification." *Intelligent data analysis* 1.1-4 (1997): 131-156.
- **N. R. Draper and H. Smith**, **Applied Regression Analysis, 2nd edition**,
Wiley, 1981 (the “bible” for classical regression methods in statistics)
- **An introduction to variable and feature selection**, **Isabelle Guyon, André Elisseeff**, The Journal of Machine Learning Research archive Volume 3, 3/1/2003,
pp. 1157-1182
- **Mohammed J. Zaki, course notes, High Dimesional Notes**
<http://www.cs.rpi.edu/~zaki/www-new/uploads/Dmcourse/Main/chap6.pdf>