ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό

Ενότητα # 2: Dimesionality Reduction and Feature Selection

Διδάσκων: Μιχάλης Βαζιργιάννης

Τμήμα: Προπτυχιακό Πρόγραμμα Σπουδών “Πληροφορικής”

BY NC ND

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται … .

Σκοποί ενότητας

Εισαγωγή και εξοικείωση με τις μεθόδους Pre-processing, Exploration, Feature selection, Dimensionality reduction, feature extraction and evaluation.

Περιεχόμενα ενότητας

- Pre-processing
- Exploration
- Feature selection
- Dimensionality reduction
- Feature extraction and evaluation

ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Distance Measures

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό, **Ενότητα # 2:** Dimesionality Reduction and Feature Selection

Διδάσκων: Μιχάλης Βαζιργιάννης, **Τμήμα:** Προπτυχιακό Πρόγραμμα Σπουδών "Πληροφορικής"

BY NC ND

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

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 measurements

$$\mathbf{x(i)} = (\mathbf{x}_1(\mathbf{i}), \mathbf{x}_2(\mathbf{i}), \ldots, \mathbf{x}_p(\mathbf{i}))$$

- Most common distance metric is *Euclidean* distance:

$$d_E(i, j) = \left(\sum_{k=1}^{p} (x_k(i) - x_k(j))^2 \right)^{\frac{1}{2}}$$

- Makes sense in the case where the different measurements are commensurate; 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:

$$d_{WE}(i, j) = \left(\sum_{k=1}^{p} w_k (x_k(i) - x_k(j))^2 \right)^{\frac{1}{2}}$$

Other Distance Metrics

- Minkowski or L_λ metric:

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

- Manhattan, city block or L_1 metric:

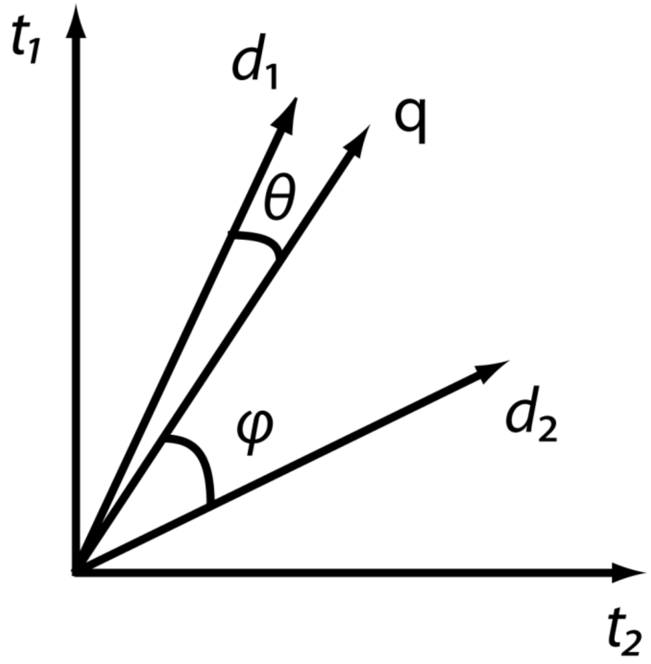
$$d(i, j) = \sum_{k=1}^{p} \left| x_k(i) - x_k(j) \right|$$

$$d(i, j) = \max_{k} \left| x_k(i) - x_k(j) \right|$$

- L_∞

Cosine based similarity

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



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

- Edit (Levenshtein) distance
  - “**k**itten → **s**itten (substitution of "s" for "k")
  - sitt**e**n → sitt**i**n (substitution of "i" for "e")
  - sittin → sittin**g** (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 :

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

where $\overline{x_k}$ is the sample mean:

$$\overline{\mathrm{x}}_{\mathrm{k}} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x}_{\mathrm{k}}(\mathrm{i})$$

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), …, x(n) and y(1), …, y(n).
- Sample covariance of X and Y is:

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

- 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

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

Results Evaluation metrics

- Confusion matrix:

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

- Precision $\frac{TP}{TP+FP}$
- Recall $\frac{TP}{TP+FN}$
- Matching coefficient $\frac{TP+TN}{TP+TN+FP+FN}$

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 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), IG(X, A) represents the reduction in entropy (~ gain in Information) of X achieved by learning the state of A:

  - IG(X,A)=H(X)–H(X|A)

Filter methods - Chi-squared test (χ2)

- 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 values, c classes

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

  E_{ij}: the expected frequency

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

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:

$$F = \frac{\dfrac{RSS_1 - RSS_2}{r_2 - r_1}}{\dfrac{RSS_2}{n - r_2}}$$

- *Assume two models* p_2, p_1 with $|p_2| > |p_1|$
- Repeat until F-value < threshold_f (or p-value > threshold_p)
- *RSSi* the residual sum of squares - the error induced by the model:

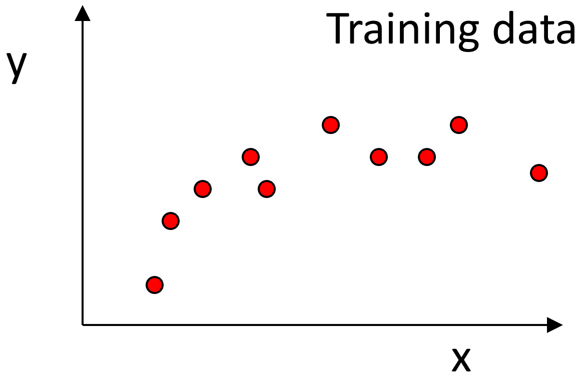
$$F = \sum_1^n (y_i - f(x_i))^2$$

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

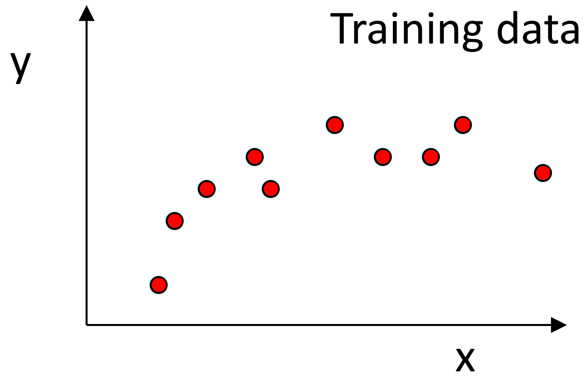
Backward Elimination

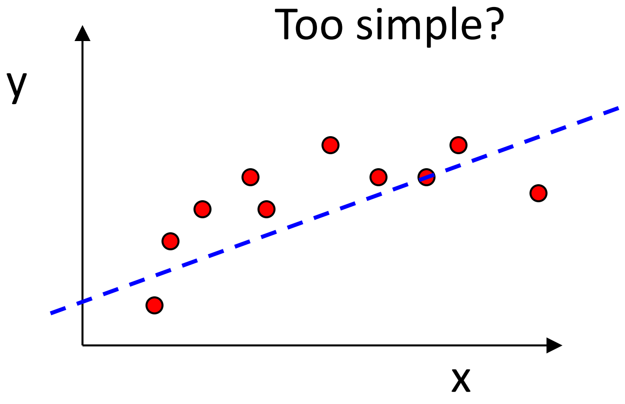
- start with the full model
- drop the predictor that produces the smallest F value (or highest p-value)
- Continue until F-value < threshold_f
  - (or p-value > threshold_p)
- Sometimes constraint N>p

Complexity versus Goodness of Fit

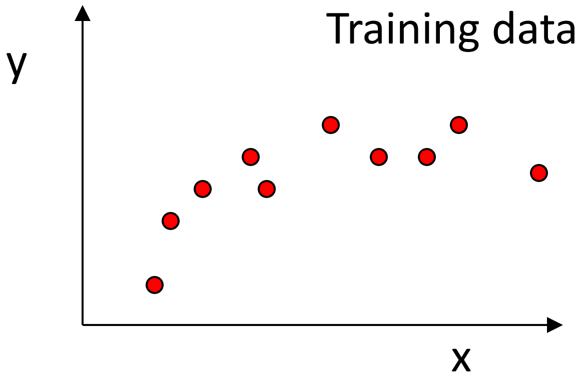


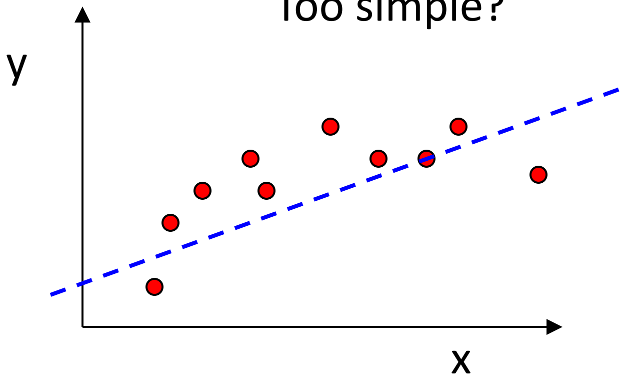
Complexity versus Goodness of Fit

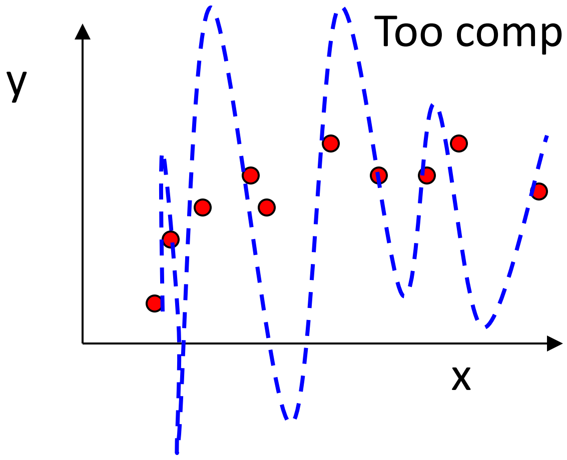




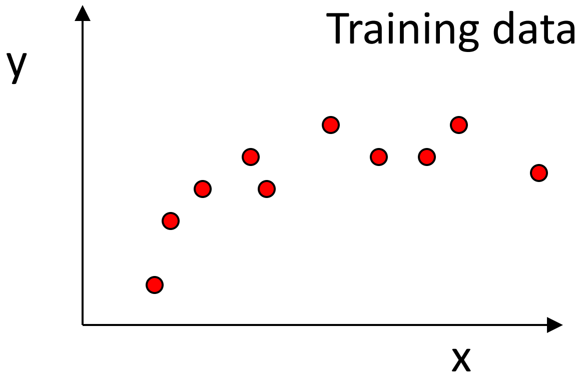
Complexity versus Goodness of Fit

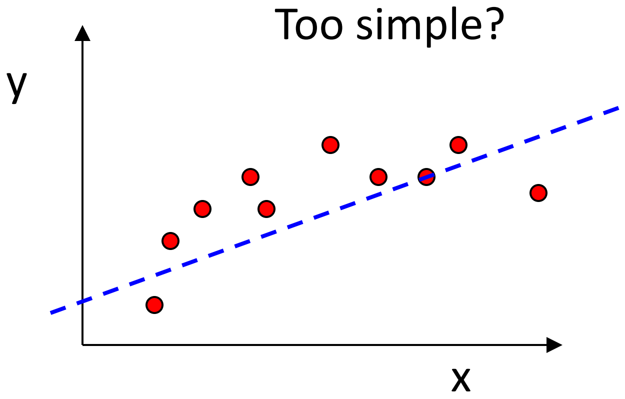

Training data

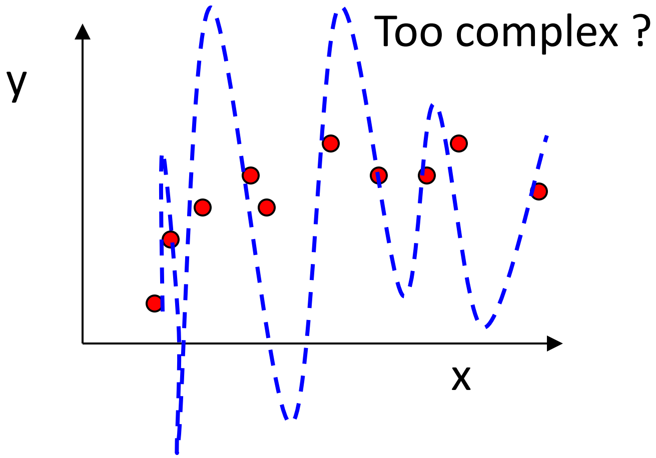

Too simple?

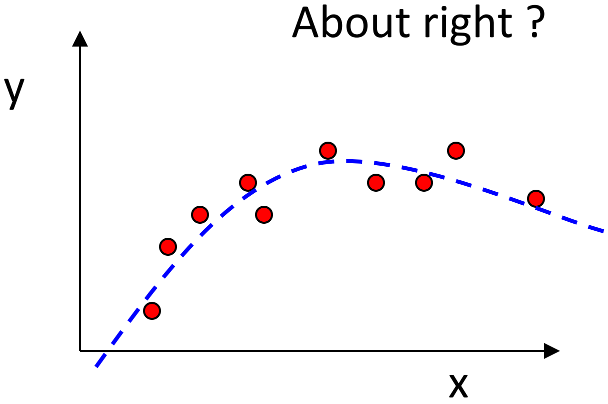

Too complex ?

Complexity versus Goodness of Fit

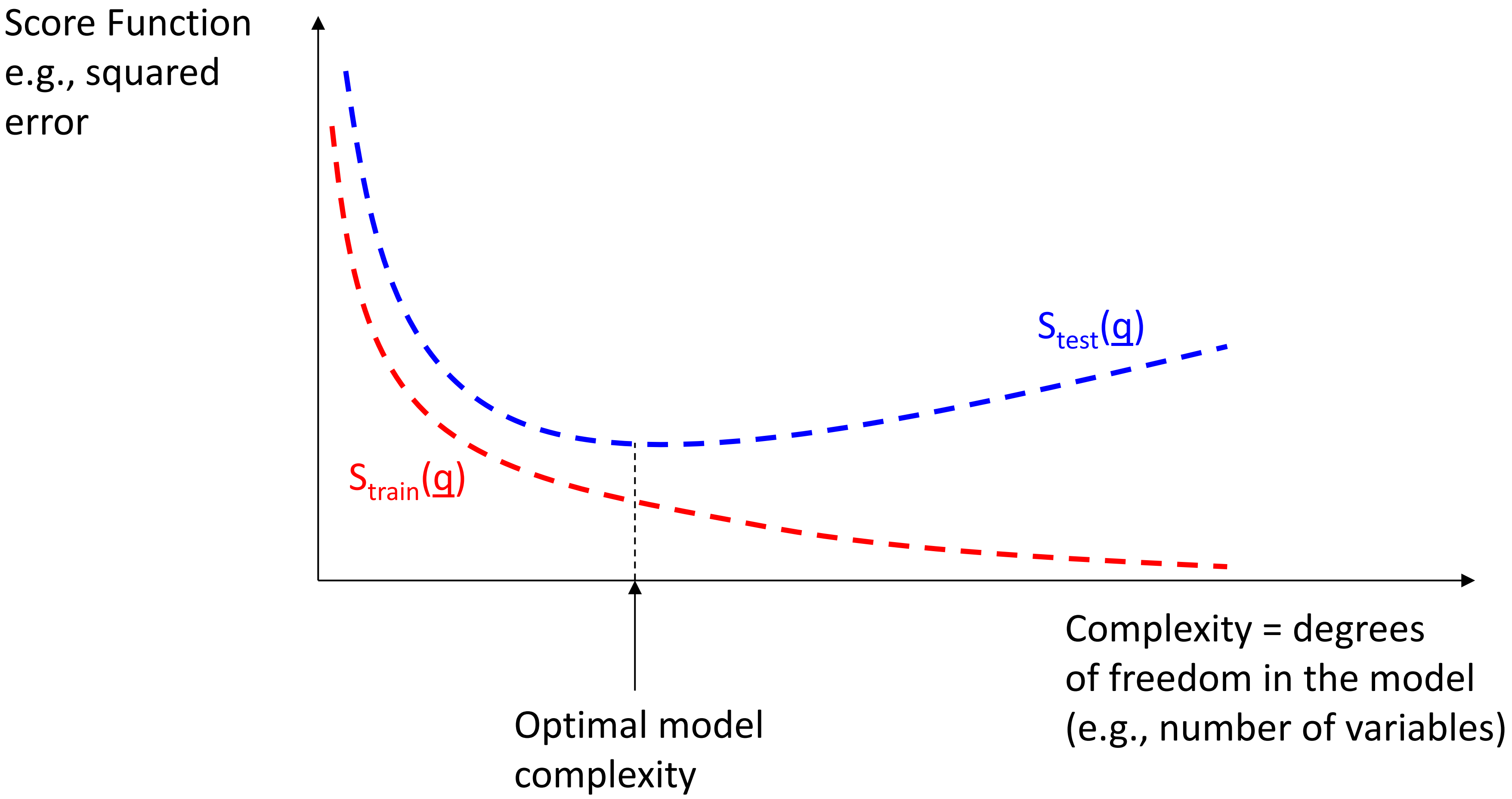

Training data


Too simple?


Too complex ?


About right ?

Complexity and Generalization



Optimal model complexity

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

ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Dimensionality reduction

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό, **Ενότητα # 2:** Dimesionality Reduction and Feature Selection

Διδάσκων: Μιχάλης Βαζιργιάννης, **Τμήμα:** Προπτυχιακό Πρόγραμμα Σπουδών “Πληροφορικής”

BY NC ND

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Data features

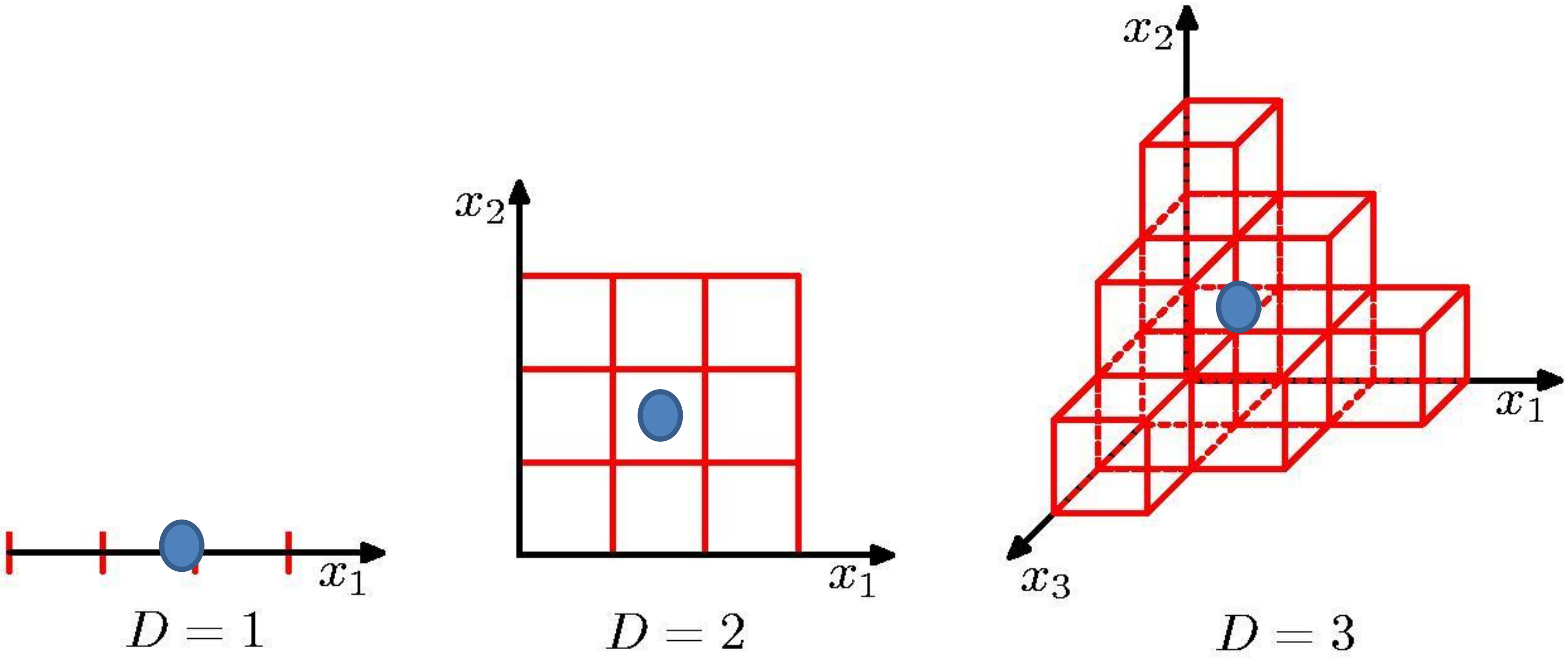
- Huge volume/ Dimensionality
- Heterogeneity
- Dynamism
  - Motion
  - Availability?
  - Frequent Updates
- Huge query loads
- Examples: Web, P2P systems, Image data

Curse of Dimensionality

- Some coordinates 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

- **2dk neighbors are needed for a d dimensional space**
- **Distance computations are increasingly complex**

Empty space phenomenon

Hyper sphere within a hyper rectangle

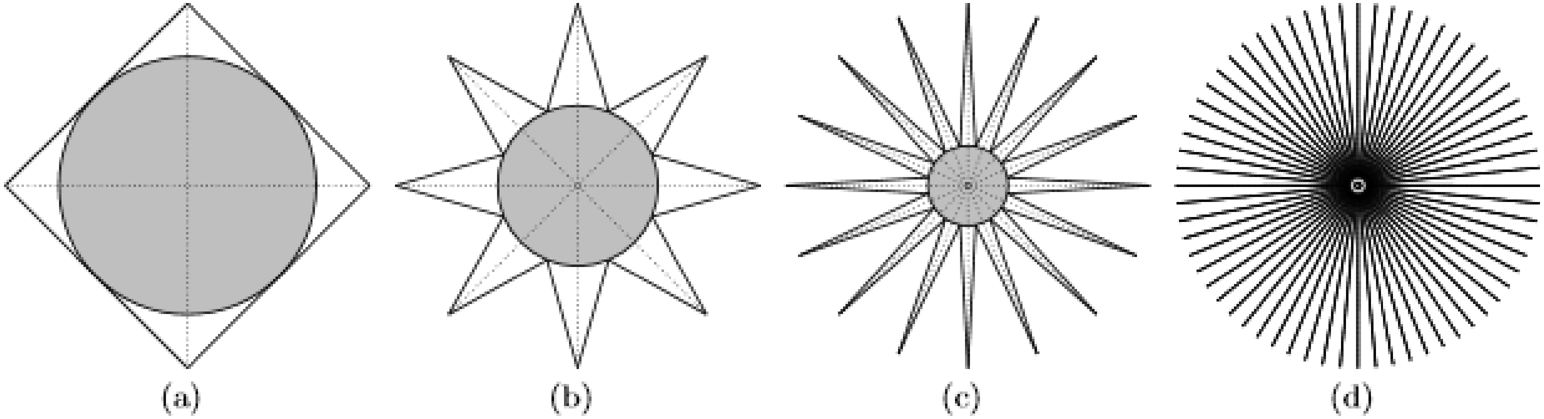
Respective Volumes $\mathbf{V}(\boldsymbol{S}) = {2r^d\pi^{\frac{d}{2}}}\Big/_{d\Gamma({}^{d}/_{2})}, \boldsymbol{V}(\boldsymbol{R}) = (\mathbf{2}\boldsymbol{r})^{\boldsymbol{d}}$

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

$$\lim_{d\to\infty}\left({\pi^{\frac{d}{2}}}\Big/_{d2^{d-1}\Gamma({}^{d}/_{2})}\right) = 0$$

- 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.

http://www.cs.rpi.edu/~zaki/www-new/uploads/Dmcourse/Main/chap6.pdf

Dim. Reduction – Linear Algorithms

- *Matrix Factorization methods*
- Principal Components Analysis (PCA)
- Singular Value Decomposition (SVD)
- Multidimensional Scaling (MDS)
- Non negative Matrix Factorization (NMF)
- Latent Semantic Indexing (LSI)

Low Rank Approximation

Data: $\boldsymbol{X} = \{\boldsymbol{x}_i \in \boldsymbol{R}^{mxn} | \boldsymbol{x}_i$ **columns of** $\boldsymbol{X}\}$

Goal: approximate $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^T$,

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

- each data vector $\boldsymbol{x}_i$: $\boldsymbol{x}_i \sim \mathrm{U}\boldsymbol{v}_i{}^T$, $\boldsymbol{v}_i$ is the i-th column of $\boldsymbol{V}$.

Geometric interpretation:

- each data vector $\boldsymbol{x}_i \in \boldsymbol{R}^m$, $_i\sim \mathrm{U}\boldsymbol{v}_i{}^T$, is approximated by its projection to an r-dimesional space spanned by the column vectors of U
- $\boldsymbol{Y} = \boldsymbol{U}\ \boldsymbol{V}^T$ the approximation matrix, max rank r

Frobenius distance

$$||\boldsymbol{X} - \boldsymbol{Y}||^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}(\boldsymbol{X}_{ij} - \boldsymbol{Y}_{ij})^2$$

- Minimizing the Frobenius distance can be considered as *maximum 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

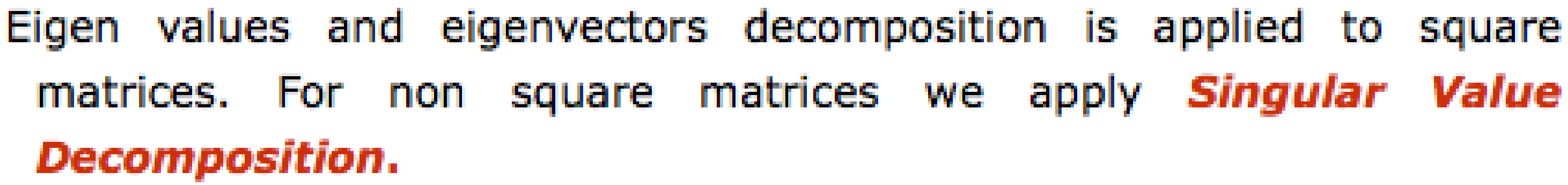
Dim. Reduction–Eigenvectors

A nxn matrix

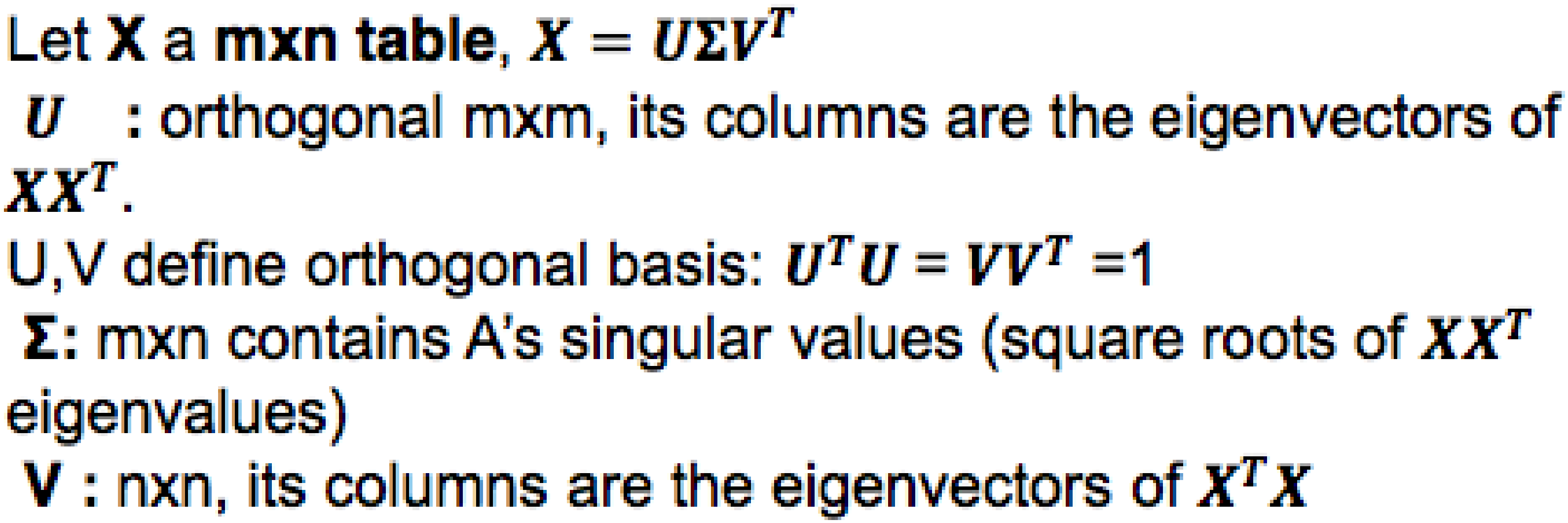
- eigenvalues λ: $|A-\lambda I|=0$
- Eigenvectors *x* : $Ax=\lambda x$
- Matrix rank: # linearly independent rows or columns
- A real symmetric table A nxn can be expressed as: $A=U\Lambda U^T$
- U's columns are A's eigenvectors
- Λ' s diagonal contains A's eigenvalues
- $A=U\Lambda U^T=\lambda_1 x_1 x^T{}_1+\lambda_2 x_2 x^T{}_2+\ldots+\lambda_n x_n x^T{}_n$
- $x_1 x^T{}_1$ 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 **X** a **mxn table**, $\boldsymbol{X = U\Sigma V^T}$

$\boldsymbol{U}$: orthogonal mxm, its columns are the eigenvectors of $\boldsymbol{XX^T}$.

U,V define orthogonal basis: $\boldsymbol{U^T U} = \boldsymbol{VV^T}$ =1

Σ: mxn contains A's singular values (square roots of $\boldsymbol{XX^T}$ eigenvalues)

V : nxn, its columns are the eigenvectors of $\boldsymbol{X^T X}$

Singular Value Decomposition (SVD) - I

PROOF:

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

Similarly: -> $X^T X = U\Sigma(V^T V)\Sigma^T U^T$ 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 **mxn table,** $\boldsymbol{X = U\Sigma V^T}$ then an r rank approximation of X is:

$$\boldsymbol{Y = U_{mxr}\,\mathrm{diag}(\sigma_1, \ldots \sigma_r) V_{nxr}{}^T}$$

Singular Value Decomposition (SVD) - II

Matrix approximation

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

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

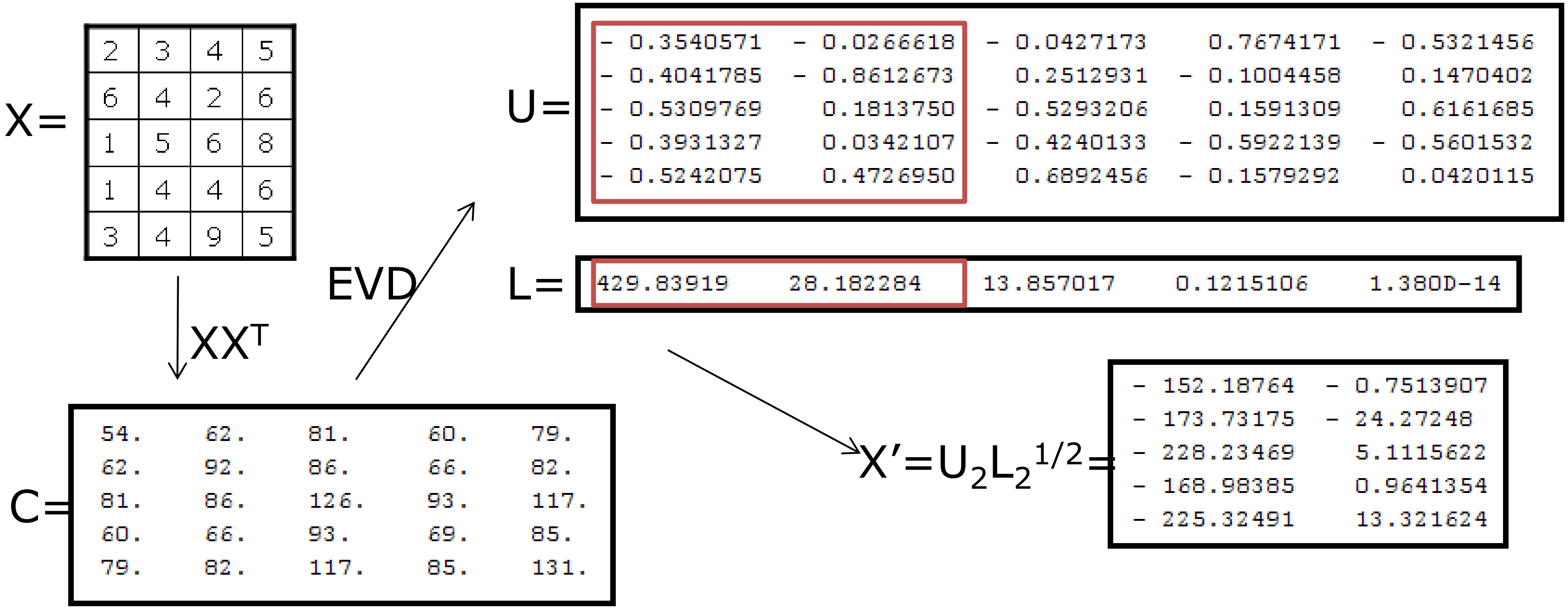
- where A^H transpose of *A*, σ_i are the singular values of *A*, and the trace function is used.
- The Frobenius norm is sub-multiplicative and is very useful for numerical linear algebra. This norm is often easier to compute than induced norms.

Multidimensional Scaling (MDS)

- Initially we depict vectors in random places
- Iteratively reposition them in order to minimize 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 and projection on the k eigenvectors that correspond 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



X=

2	3	4	5
6	4	2	6
1	5	6	8
1	4	4	6
3	4	9	5

C=

```
54.    62.    81.     60.    79.
62.    92.    86.     66.    82.
81.    86.    126.    93.    117.
60.    66.    93.     69.    85.
79.    82.    117.    85.    131.
```

U=

```
- 0.3540571  - 0.0266618  - 0.0427173    0.7674171  - 0.5321456
- 0.4041785  - 0.8612673    0.2512931  - 0.1004458    0.1470402
- 0.5309769    0.1813750  - 0.5293206    0.1591309    0.6161685
- 0.3931327    0.0342107  - 0.4240133  - 0.5922139  - 0.5601532
- 0.5242075    0.4726950    0.6892456  - 0.1579292    0.0420115
```

L=

```
429.83919    28.182284    13.857017    0.1215106    1.380D-14
```

$X'=U_2L_2^{1/2}=$

```
- 152.18764  - 0.7513907
- 173.73175  - 24.27248
- 228.23469    5.1115622
- 168.98385    0.9641354
- 225.32491    13.321624
```

Principal Components Analysis

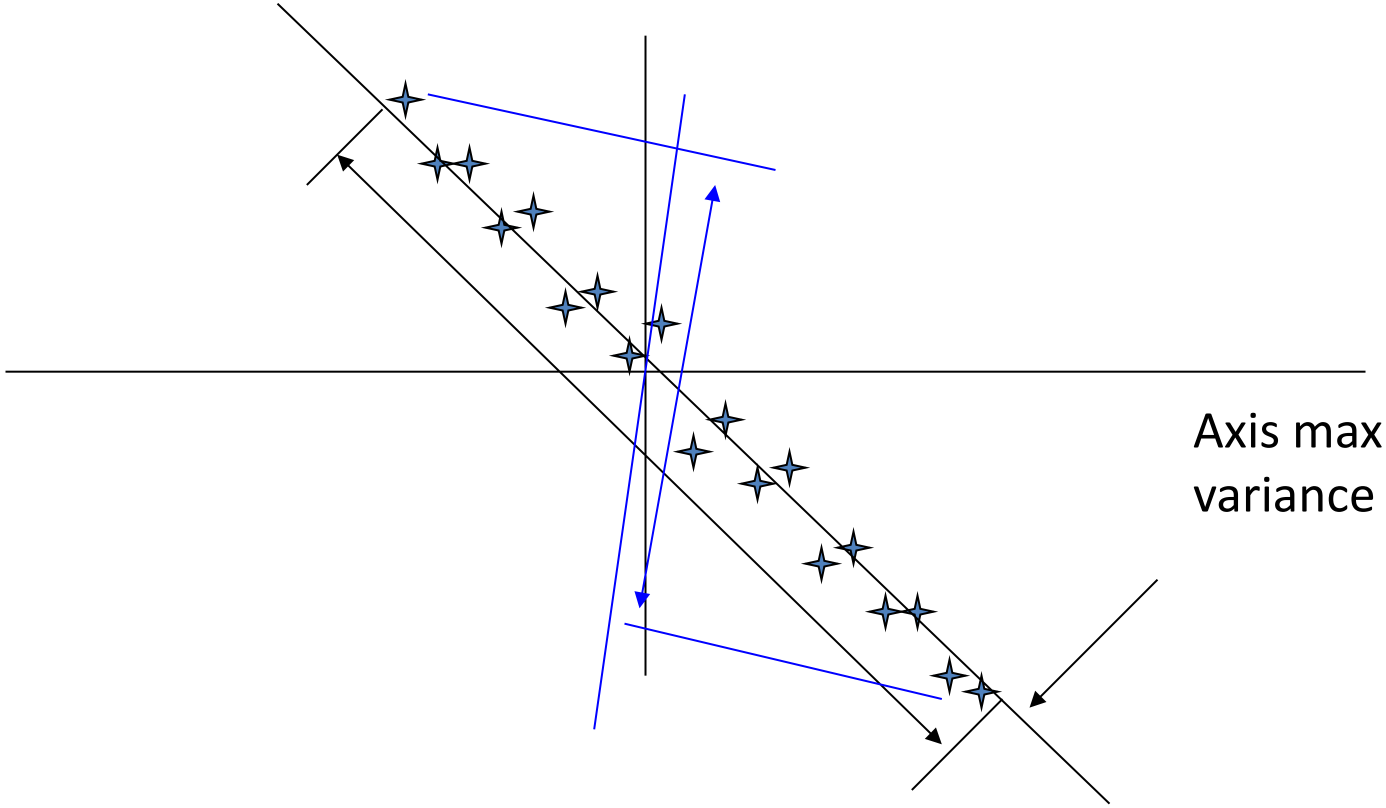
- The main concept behind *Principal Components Analysis* is dimensionality reduction, maintaining as much as possible data's variance.
- variance: $V(X)=\sigma^2=E[(X-\mu)^2]$
- Let *N* objects, with mean value, *m*, it is approximated as:

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

- Sample of *N* objects with unknown mean value:

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

Dimensionality reduction based on variance maintenance



Axis maximizing variance

Principal Components Analysis

- «A linear transformation that chooses a new coordinate system for the data set such that the greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on ...» (wikipedia)
- Let n dimensional data, with dimensions: $x_1,\ldots,x_n$
- The objective is to project the data to k dimensions via some linear decomposition:

  $y_1=a_1*x_1+\ldots+a_n*x_n$

  ………

  $y_k=b_1*x_1+\ldots+b_n*x_n$
- should maintain the variance of the original data

Covariance Matrix

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

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

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

- Also: 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 W(i,j) we have the covariance of X_i,X_j.

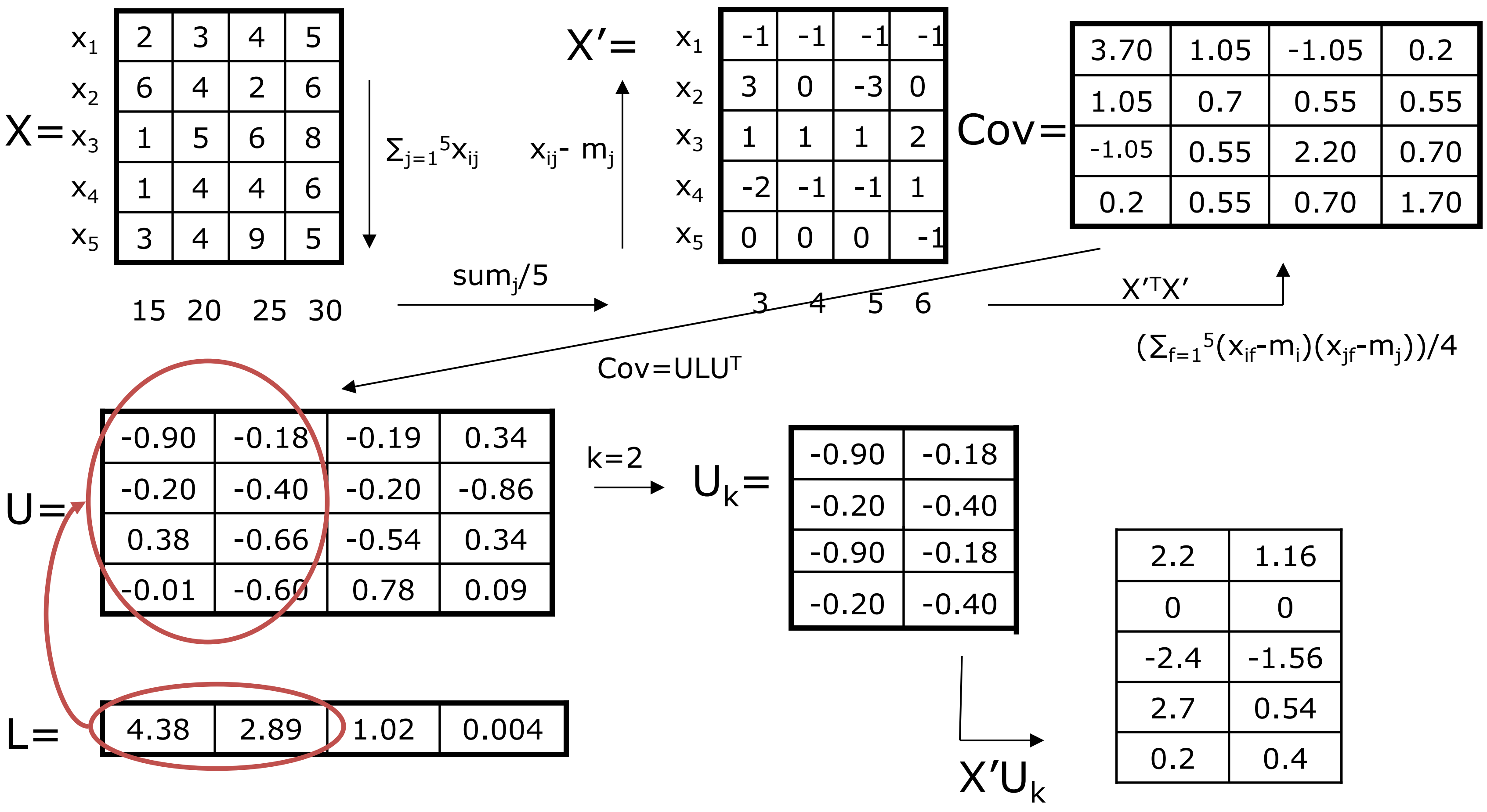
- Calculate eigenvalues and eigenvectors of W (X,D) = UAU^T

- Retain k largest eigenvalues and corresponding eigenvectors
  - k is an input parameter
  - There is an input parameter and k is calculate by

$$\sum^{p}_{j=k+1}\lambda_j / \sum^{p}_{j=1}\lambda_j > 85\%$$

- Projection : $A'X_k$

Principal Components Analysis



$$X=\begin{array}{c|cccc} x_1 & 2 & 3 & 4 & 5 \\ x_2 & 6 & 4 & 2 & 6 \\ x_3 & 1 & 5 & 6 & 8 \\ x_4 & 1 & 4 & 4 & 6 \\ x_5 & 3 & 4 & 9 & 5 \\ \hline & 15 & 20 & 25 & 30 \end{array}$$

$\Sigma_{j=1}^{5}x_{ij}$

$\text{sum}_j/5$

$3 \quad 4 \quad 5 \quad 6$

$x_{ij}-m_j$

$$X'=\begin{array}{c|cccc} x_1 & -1 & -1 & -1 & -1 \\ x_2 & 3 & 0 & -3 & 0 \\ x_3 & 1 & 1 & 1 & 2 \\ x_4 & -2 & -1 & -1 & 1 \\ x_5 & 0 & 0 & 0 & -1 \end{array}$$

X'^TX'

$(\Sigma_{f=1}^{5}(x_{if}-m_i)(x_{jf}-m_j))/4$

$$\text{Cov}=\begin{array}{|c|c|c|c|} 3.70 & 1.05 & -1.05 & 0.2 \\ 1.05 & 0.7 & 0.55 & 0.55 \\ -1.05 & 0.55 & 2.20 & 0.70 \\ 0.2 & 0.55 & 0.70 & 1.70 \end{array}$$

$\text{Cov}=ULU^T$

$$U=\begin{array}{|c|c|c|c|} -0.90 & -0.18 & -0.19 & 0.34 \\ -0.20 & -0.40 & -0.20 & -0.86 \\ 0.38 & -0.66 & -0.54 & 0.34 \\ -0.01 & -0.60 & 0.78 & 0.09 \end{array}$$

$$L=\begin{array}{|c|c|c|c|} 4.38 & 2.89 & 1.02 & 0.004 \end{array}$$

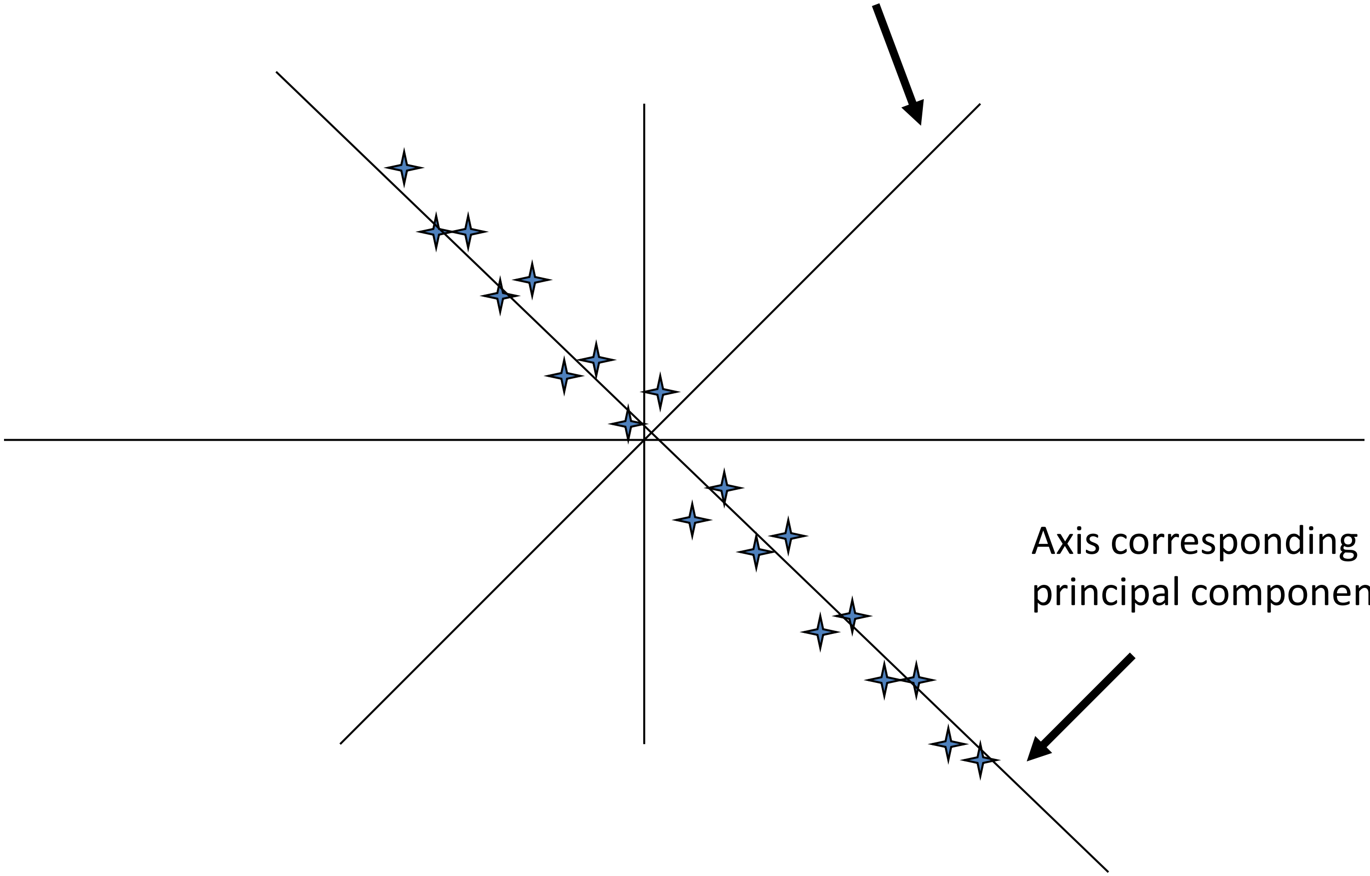
$k=2$

$$U_k=\begin{array}{|c|c|} -0.90 & -0.18 \\ -0.20 & -0.40 \\ -0.90 & -0.18 \\ -0.20 & -0.40 \end{array}$$

$X'U_k$

$$\begin{array}{|c|c|} 2.2 & 1.16 \\ 0 & 0 \\ -2.4 & -1.56 \\ 2.7 & 0.54 \\ 0.2 & 0.4 \end{array}$$

PCA, example



Axis corresponding to the second principal component

Axis corresponding to the first principal component

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.

Non Negative Matrix factorization (NMF)

- - Applying SVD results in factorized matrices with positive and negative elements may contradict the physical meaning of the result.
- Example:
- - X gray-scale image intensities, Y its SVD approximation
- - difficult to interpret the reconstructed matrix Y for a gray-scale image with negative elements.
- - *Nonnegative matrix factorization (NMF)*
- find the reduced rank *nonnegative factors* to approximate a given nonnegative data matrix.

Non Negative Matrix factorization (NMF)

Assume X mxn data matrix $(X_{ij} \geq 0), r \ll \min(m,n)$

Then NMF finds non negative matrices

$$U \in R^{mxr}, V \in R^{nxr}: X \approx UV^T$$

To find U, V is to minimize Euclidian Distance $X - UV^T$:

$$\min_{U,V} \quad f(U,V) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(X_{ij} \log \frac{X_{ij}}{(UV^\top)_{ij}} - X_{ij} + (UV^\top)_{ij} \right)$$

$$\text{s. t.} \quad U_{ia} \geq 0, V_{jb} \geq 0, \forall i, a, b, j.$$

NMF Algorithms

- Multiplicative: updating solutions U and V

$$V_{bj}^{\top} \leftarrow V_{bj}^{\top} \frac{\left(U^{\top} X\right)_{bj}}{\left(U^{\top} U V^{\top}\right)_{bj}} \qquad U_{ia} \leftarrow U_{ia} \frac{(XV)_{ia}}{(UV^{\top}V)_{ia}}$$

- Gradient descent algorithms

$$V_{bj}^{\top} \leftarrow V_{bj}^{\top} - \epsilon_V \frac{\partial f}{\partial V_{bj}^{\top}} \qquad U_{ia} \leftarrow U_{ia} - \epsilon_U \frac{\partial f}{\partial U_{ia}}$$

- ε_v and ε_υ are the step sizes.

SVD application - Latent Structure in documents

- Documents are represented based on the Vector Space Model
- Vector space model consists of the keywords contained in a document.
- In many cases baseline keyword based performs poorly – not able to detect synonyms.
- Therefore document clustering is problematic
- Example where of keyword matching with the query: “IDF in computer-based information look-up”

	access	document	retrieval	information	theory	database	indexing	computer
Doc1	x	x	x			x	x	
Doc2				x	x			x
Doc3			x	x				x

Indexing by Latent Semantic Analysis (1990) Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, Journal of the American Society of Information Science

Latent Semantic Indexing (LSI) -I

- Finding similarity with exact keyword matching is problematic.

- Using SVD we process the initial document-term document.

- Then we choose the k larger singular values. The resulting matrix is of order k and is the most similar to the original one based on the Frobenius norm than any other k-order matrix.

Latent Semantic Indexing (LSI) - II

- The initial matrix is SVD decomposed as: $A=ULV^T$
- Choosing the top-k singular values from L we have:

$$A_k=U_kL_kV_k^T ,$$

- L_k is square kxk containing the top-k singular values of the diagonal in matrix L,
- $U_{k,}$ the mxk matrix containing the first k columns in U (left singular vectors)
- $V_k^{T,}$ the kxn matrix containing the first k lines of V^T (right singular vectors)

Typical values for κ~200-300 (empirically chosen based on experiments appearing in the bibliography)

LSI capabilities

- - Term to term similarity: $A_k A_k^T = U_k L_k^2 U_k^T$
- Where Ak=UkLkVt
- - Document-document similarity: $A_k^T A_k = V_k L_k^2 V_k^T$
- - Term document similarity (as an element of the transformed – document matrix)
- - Extended query capabilities transforming initial query q to q_n

  $q_n = q^T U_k L_k^{-1}$
- - Thus q_n can be regarded a line in matrix V_k

LSI – an example

LSI application on a term – document matrix

C1: Human machine Interface for Lab ABC computer application

C2: A survey of user opinion of computer system response time

C3: The EPS user interface management system

C4: System and human system engineering testing of EPS

C5: Relation of user-perceived response time to error measurements

M1: The generation of random, binary unordered trees

M2: The intersection graph of path in trees

M3: Graph minors IV: Widths of trees and well-quasi-ordering

M4: Graph minors: A survey

- The dataset consists of 2 classes, 1st: "human – computer interaction" (c1-c5) 2nd: related to graph (m1-m4). After feature extraction the titles are represented as follows

Indexing by Latent Semantic Analysis (1990) Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, Journal of the American Society of Information Science

LSI – an example

	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	1	0	0	1	0	0	0	0	0
Interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	0	1	1	2	0	0	0	0	0
Response	0	1	0	0	1	0	0	0	0
Time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
Survey	0	1	0	0	0	0	0	0	1
Trees	0	0	0	0	0	1	1	1	0
Graph	0	0	0	0	0	0	1	1	1
Minors	0	0	0	0	0	0	0	1	1

LSI – an example

$A=ULV^T$

A=

1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	2	0	0	0	0	0
0	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	1
0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1	1

LSI – an example

$A=ULV^T$

$U=$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41	0	0	0
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11	0	0	0
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49	0	0	0
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01	0	0	0
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27	0	0	0
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05	0	0	0
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05	0	0	0
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17	0	0	0
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58	0	0	0
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23	0	0	0
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23	0	0	0
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18	0	0	0

LSI – an example

$A=ULV^T$

L=

3.34	0	0	0	0	0	0	0	0
0	2.54	0	0	0	0	0	0	0
0	0	2.35	0	0	0	0	0	0
0	0	0	1.64	0	0	0	0	0
0	0	0	0	1.50	0	0	0	0
0	0	0	0	0	1.31	0	0	0
0	0	0	0	0	0	0.85	0	0
0	0	0	0	0	0	0	0.56	0
0	0	0	0	0	0	0	0	0.36
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

LSI – an example

$A=ULV^T$

V=

0.20	-0.06	0.11	-0.95	0.05	-0.08	0.18	-0.01	-0.06
0.61	0.17	-0.50	-0.03	-0.21	-0.26	-0.43	0.05	0.24
0.46	-0.13	0.21	0.04	0.38	0.72	-0.24	0.01	0.02
0.54	-0.23	0.57	0.27	-0.21	-0.37	0.26	-0.02	-0.08
0.28	0.11	-0.51	0.15	0.33	0.03	0.67	-0.06	-0.26
0.00	0.19	0.10	0.02	0.39	-0.30	-0.34	0.45	-0.62
0.01	0.44	0.19	0.02	0.35	-0.21	-0.15	-0.76	0.02
0.02	0.62	0.25	0.01	0.15	0.00	0.25	0.45	0.52
0.08	0.53	0.08	-0.03	-0.60	0.36	0.04	-0.07	-0.45

LSI – an example

Choosing the 2 largest singular values we have

U_k=

0.22	-0.11
0.20	-0.07
0.24	0.04
0.40	0.06
0.64	-0.17
0.27	0.11
0.27	0.11
0.30	-0.14
0.21	0.27
0.01	0.49
0.04	0.62
0.03	0.45

L_k=

3.34	0
0	2.54

V_k^T=

0.20	0.61	0.46	0.54	0.28	0.00	0.02	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53

LSI (2 singular values)

A_k =

	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
Interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
Computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
User	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
System	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
Response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
Time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
Survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
Trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
Graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
Minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

LSI Example

- Query: “human computer interaction” retrieves documents: c_1,c_2, c_4 but *not* c_3 and c_5.
- If we submit the same query (based on the transformation shown before) to the transformed matrix we retrieve (using cosine similarity) all c_1-c_5 even if c_3 and c_5 have no common keyword to the query.
- According to the transformation for the queries we have:

Query transformation

	query
human	1
Interface	0
computer	1
User	0
System	0
Response	0
Time	0
EPS	0
Survey	0
Trees	0
Graph	0
Minors	0

q=

1
0
1
0
0
0
0
0
0
0
0
0

Query transformation

q^T=

1	0	1	0	0	0	0	0	0	0	0	0

U_k=

0.22	-0.11
0.20	-0.07
0.24	0.04
0.40	0.06
0.64	-0.17
0.27	0.11
0.27	0.11
0.30	-0.14
0.21	0.27
0.01	0.49
0.04	0.62
0.03	0.45

L_k=

0.3	0
0	0.39

q_n=$q^T U_k L_k$ =

0.138	-0.0273

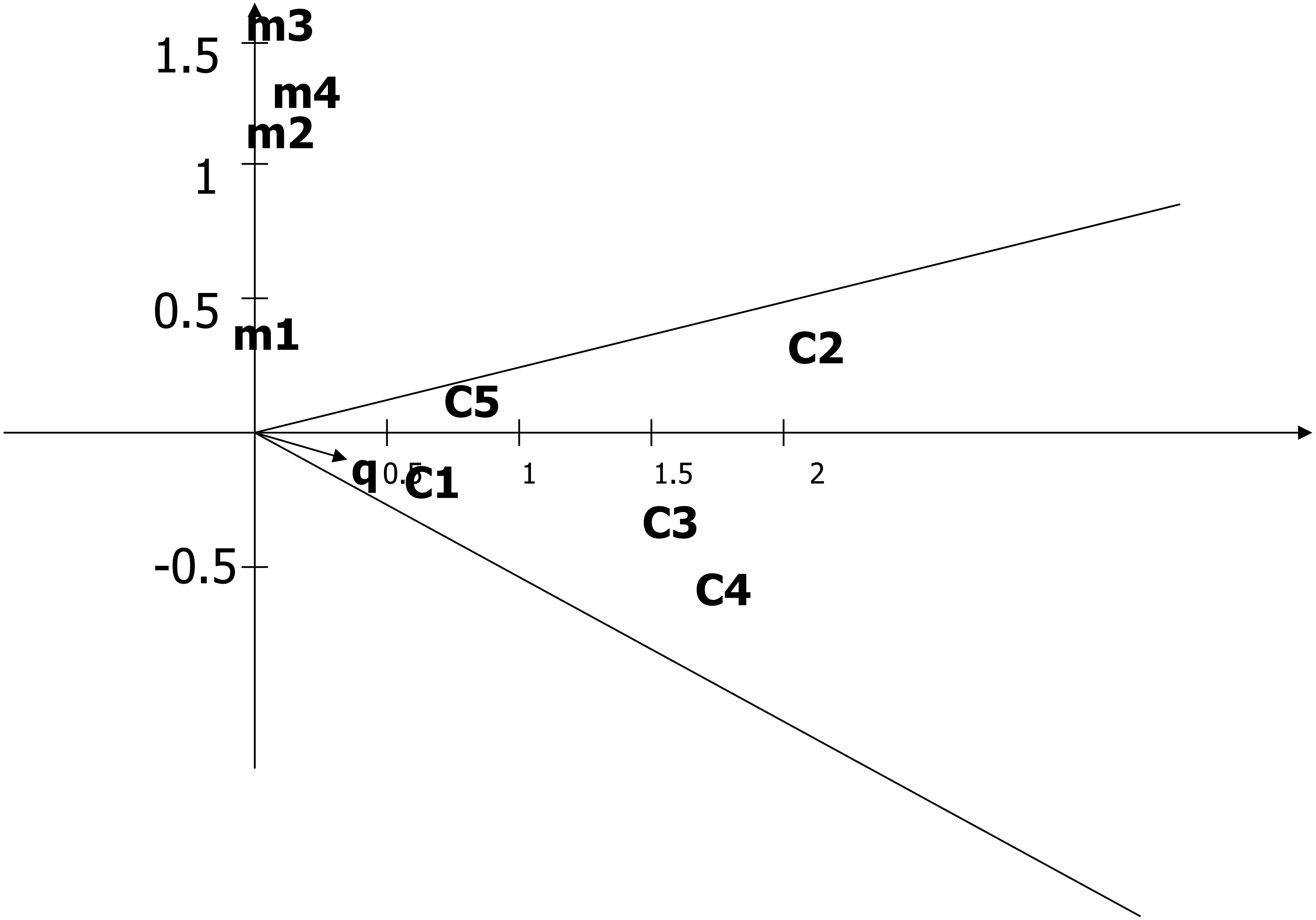
Query transformation

Map docs to the 2 dim space $V_k L_k =$

$$\begin{bmatrix} 0.20 & -0.06 \\ 0.61 & 0.17 \\ 0.46 & -0.13 \\ 0.54 & -0.23 \\ 0.28 & 0.11 \\ 0.00 & 0.19 \\ 0.01 & 0.44 \\ 0.02 & 0.62 \\ 0.08 & 0.53 \end{bmatrix} \begin{bmatrix} 3.34 & 0 \\ 0 & 2.54 \end{bmatrix} = \begin{bmatrix} 0.67 & -0.15 \\ 2.04 & 0.43 \\ 1.54 & -0.33 \\ 1.80 & -0.58 \\ 0.94 & 0.28 \\ 0.00 & 0.48 \\ 0.03 & 1.12 \\ 0.07 & 1.57 \\ 0.27 & 1.35 \end{bmatrix}$$

$$q_n L_k = \begin{bmatrix} 0.138 & -0.0273 \end{bmatrix} \begin{bmatrix} 3.34 & 0 \\ 0 & 2.54 \end{bmatrix} = \begin{bmatrix} 0.46 & -0.069 \end{bmatrix}$$

Query transformation



Query transformation

- Comparison of the transformed query to the new document vectors based on cosine similarity, where the similarity is computed as: Cos(x,y)=<x,y>/||x||.||y||

Where $x=(x_1,\ldots,x_n)$, $y=(y_1,\ldots,y_n)$

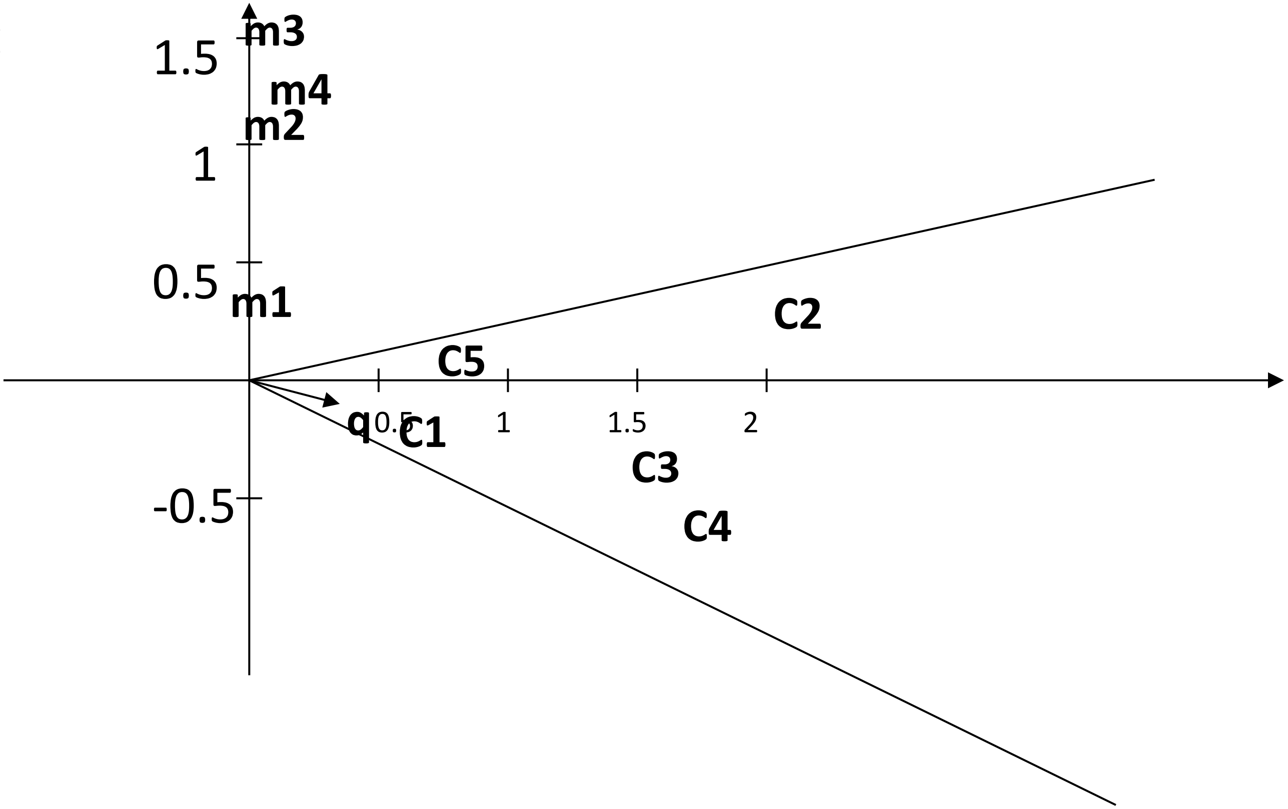
$<x,y>=x_1*y_1+\ldots+x_n*y_n$

||x||=sqrt(<x,x>)

Query transformation

- The cosine similarity matrix of query vector to the documents is:

	query
C1	0.99
C2	0.94
C3	0.99
C4	0.99
C5	0.90
M1	-0.14
M2	-0.13
M3	-0.11
M4	0.05



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τέλος Ενότητας # 2

Μάθημα: Εξόρυξη γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό, **Ενότητα # 2:** Dimesionality Reduction and Feature Selection

Διδάσκων: Μιχάλης Βαζιργιάννης**, Τμήμα:** Προπτυχιακό Πρόγραμμα Σπουδών "Πληροφορικής"

cc BY NC ND

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης