

# Dimensionality Reduction

Michalis Vazirgiannis

LIX

2011

# Data features

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

# Dimensionality Reduction -Objectives

- What is dimensionality reduction?
  - A methodology that attempts to project a set of high dimensional vectors to a lower dimensionality space while retaining metrics among them.
- Let a multidimensional data set
$$X = (x_1, \dots, x_n), x_i \in \mathbb{R}^d,$$
- Aim: find a “credible” mapping of the  $n$  vectors to  $\mathbb{R}^k$ ,  $k \ll d$
- Credible:
  - Maintain: variation / distances
- In a lower dimensional space clustering-structure is maintained and “amplified”
- similarity queries are much faster

# Why ?

- Why is it necessary?
  - Curse of dimensionality (exponentially increasing data to represent adequately a pattern)
  - Empty space phenomenon (longest/shortest distances converge).
  - Clustering becomes infeasible
  - In distributed environments: Transmitted data.
- Why is it feasible ?
  - Some coordinates do not contribute to the data representation.
  - Subsets of the dimensions may be highly correlated.
- When is it applied?
  - When the cost of dim. reduction application is worth the expected benefit.

# Dimensionality reduction – fundamentals...

- Dimensionality Reduction Methodology
  - $N$  vectors in  $\mathbb{R}^n$ .
  - Projection space is  $\mathbb{R}^k$ .
  - Must find a transformation  $W_{k \times n}$  such that :  $X_{(k)} = W_{(k \times n)} X_{(n)}$ .
- Linear dimensionality reduction algorithms
  - All data lay in a globally linear space.
- Non linear dimensionality reduction algorithms
  - All data lay in a locally linear subspace.

# Dim. Reduction – Linear Algorithms

- Principal Components Analysis (PCA)
  - Singular Value Decomposition (SVD)
  - Multidimensional Scaling (MDS)
  - Latent Semantic Indexing (LSI)
  - FastMap [Faloutsos and Lin 1995]
  - K-Landmarks [Magdalinos, et al. 2006]
-

# Dim. Reduction – Non Linear Algorithms

- Isomap [Tenenbaum et al., 2000],
- maximum variance unfolding [Weinberger and Saul, 2005, Sun et al., 2005],
- Locally linear embedding [Roweis and Saul, 2000, Saul and Roweis, 2003],
- Laplacian eigenmaps [Belkin and Niyogi, 2003].

# Linear Algebra

Basic Principles

# Dim. Reduction–Eigenvectors

A  $n \times n$  matrix

- eigenvalues  $\lambda$ :  $|A-\lambda I|=0$
- 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
- $\Lambda$ '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  $x_1$  ( $\lambda_i$  eigenvalue,  $x_i$  eigenvector)

# Linear Independence

- Linear Independence:
  - Given vectors  $x_1 \rightarrow, x_2 \rightarrow, \dots, x_n \rightarrow$  and equation  $\lambda_1 x_1 \rightarrow + \lambda_2 x_2 \rightarrow + \dots + \lambda_n x_n \rightarrow = 0$ , if the equation has one solution,  $\lambda_i = 0$ , then all vectors are linear independent. If there are more solutions then the vectors are linearly dependent.
    - Linear dependence: One of the vectors is the result of the linear combination of one or more of remaining vectors.
- Basis of vector space  $V$ :
  - The set of linear independent vectors from which all elements of vector space  $V$  are produced as their linear combinations.
    - i.e.  $B = \{e_1, e_2, \dots, e_n\}$  with  $e_i = \{0, 0, 0, \dots, 1, \dots, 0\}$  then  $B$  is the basis of  $\mathbb{R}^n$ .
    - The same stands for  $e_i = \{1, 1, \dots, 1, 0, \dots, 0\}$
- Dimension of vector space  $V$ :
  - The cardinality of its basis,  $\dim V = n$ 
    - Notice: A space  $V$  may have more than one basis. However, all basis are of the same cardinality.

# Metric Space

- Given a non-empty set and a function  $d: X \times X \rightarrow \mathbb{R}: (x \rightarrow, y \rightarrow) \rightarrow d(x \rightarrow, y \rightarrow)$  obeying the following:
  - $d(x \rightarrow, y \rightarrow) \geq 0$  with  $x \rightarrow, y \rightarrow \in X$
  - $d(x \rightarrow, y \rightarrow) = d(y \rightarrow, x \rightarrow)$  with  $x \rightarrow, y \rightarrow \in X$
  - $d(x \rightarrow, y \rightarrow) \leq d(z \rightarrow, x \rightarrow) + d(z \rightarrow, y \rightarrow)$  with  $x \rightarrow, y \rightarrow \in X$

$d$  is a metric of  $X$ . A space  $X$  with metric  $d$  is called a metric space.  
Additionally  $d$  defined the distance of  $x \rightarrow, y \rightarrow \in X$
- Euclidean metric in  $\mathbb{R}^n$ 
  - $d(x \rightarrow, y \rightarrow) := (\sum_{i=1}^n |x_i - y_i|^2)^{1/2}$  with  $x \rightarrow = (x_1, \dots, x_n) \in \mathbb{R}^n$ ,  $y \rightarrow = (y_1, \dots, y_n) \in \mathbb{R}^n$ 
    - Generalization of the Pythagorean Theorem
- Minkowski distance
  - $d(x \rightarrow, y \rightarrow) := (\sum_{i=1}^n |x_i - y_i|^p)^{1/p}$
- Chebyshev distance
  - $d(x \rightarrow, y \rightarrow) := \lim_{p \rightarrow \infty} (\sum_{i=1}^n |x_i - y_i|^p)^{1/p} = \max(|x_i - y_i|)$
- Manhattan distance
  - $d(x \rightarrow, y \rightarrow) := \sum_{i=1}^n |x_i - y_i|$

# Norm spaces

- Given a vector space  $V$  on  $F$  ( $F=\mathbb{R}$ ).and function  $\|\bullet\|:V \rightarrow \mathbb{R}:u \rightarrow \|u\|$  obeying the following:
  - $\|u\| \geq 0$  with  $u \in V$
  - $\|\lambda u\| = |\lambda| \|u\|$  with  $u, v \in V, \lambda \in \mathbb{R}$
  - $\|u+v\| \leq \|u\| + \|v\|$  with  $u, v \in V$  (triangular inequality)Then  $\|\bullet\|$  is a norm on  $V$ . Space  $V$  with  $\|\bullet\|$  is called normed space.
- Euclidean norm in  $\mathbb{R}^n$ 
  - $\|x\|_2 := (\sum_{i=1}^n |x_i|^2)^{1/2} \quad \mu \varepsilon x = (x_1, \dots, x_n) \in \mathbb{R}^n$
- Other norms
  - $\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$
  - $\|x\|_\infty := \max(|x_i|)$
  - $\|x\|_1 := \sum_{i=1}^n |x_i|$
- Minkowski inequality (triangular inequality for  $\|\bullet\|_p$ ):
  - $(\sum_{i=1}^n |x_i + y_i|^p)^{1/p} \leq (\sum_{i=1}^n |x_i|^p)^{1/p} + (\sum_{i=1}^n |y_i|^p)^{1/p}$
- Every normed space is a metric space (Can you prove it?)

# Inner Product

- A function  $\langle, \rangle$  obeying the following:
  - $\langle \vec{a} + \vec{b}, \vec{c} \rangle = \langle \vec{a}, \vec{c} \rangle + \langle \vec{b}, \vec{c} \rangle$ ,  $\langle \lambda \vec{a}, \vec{b} \rangle = \lambda \langle \vec{a}, \vec{b} \rangle$
  - $\langle \vec{a}, \vec{b} \rangle = \langle \vec{b}, \vec{a} \rangle$
  - $\langle \vec{a}, \vec{a} \rangle \geq 0$

A vector space with function  $\langle, \rangle$  is called a space with inner product

- Usually employed formula
  - $\langle \vec{x}, \vec{y} \rangle := \sum_{i=1}^n x_i y_i$   $\mu \varepsilon \vec{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ ,  $\vec{y} = (y_1, \dots, y_n) \in \mathbb{R}^n$
  - $\langle \vec{x}, \vec{x} \rangle := \|\vec{x}\|^2$
  - $\langle \vec{x}, \vec{y} \rangle := \|\vec{x}\| \|\vec{y}\| \cos(\vec{x} \wedge \vec{y})$
- Cauchy – Buniakowski – Schwarz inequality
  - $|\langle \vec{x}, \vec{y} \rangle| \leq \langle \vec{x}, \vec{x} \rangle^{1/2} \langle \vec{y}, \vec{y} \rangle^{1/2}$
  - Simply:
    - $|\langle \vec{x}, \vec{y} \rangle| \leq \|\vec{x}\| \|\vec{y}\|$

# Singular Value Decomposition (SVD)

Decomposition into eigen values and eigenvectors is applied to square matrices. Data tables are usually non square, in these case we apply *Singular Value Decomposition*.

•**singular value**: A non-negative real number  $\sigma$  is a **singular value** for  $M$  if there exist unit-length vectors  $\mathbf{u}$  in  $K^m$  and  $\mathbf{v}$  in  $K^n$  such that:

$$M\mathbf{v} = \sigma\mathbf{u} \text{ and } M^T\mathbf{u} = \sigma\mathbf{v}$$

$\mathbf{u}$  and  $\mathbf{v}$  are called **left-singular** and **right-singular vectors** for  $\sigma$ , respectively.

*singular value decomposition* :  $M = U\Sigma V^T$

- *diagonal* entries of  $\Sigma$  are the *singular values* of  $M$ .
- *columns* of  $U / V$  are *left/right-singular vectors* for the corresponding singular values.

.

# Singular Value Decomposition (SVD) - I

## Relation to eigenvectors/values

- Let **M**  $m \times n$  **table**, can be expressed  $U\Sigma V^T$
- **U**:  $m \times m$ , its columns are  $M^*M^T$  eigenvectors.
- **U, V** define orthogonal basis:  $UU^T = VV^T = 1$
- **$\Sigma$** :  $m \times n$  contains A's singular values (square roots of  $M^*M^T$  eigenvalues)
- **V** :  $n \times n$ , its columns are  $M^T*M$  eigenvectors

## **PROOF:**

$$M = U\Sigma V^T, M^T = V\Sigma^T U^T \rightarrow MM^T = U\Sigma(V^T V)\Sigma^T U^T \rightarrow MM^T = U\Sigma\Sigma^T U^T$$

$$\text{Similarly: } \rightarrow M^T M = U\Sigma(V^T V)\Sigma^T U^T \text{ therefore } M^T M = V\Sigma^T \Sigma U^T$$

*Hence: U: eigenvectors of  $MM^T$ , V: eigenvectors of  $MM^T$  and  $\Sigma$  sqrt of  $MM^T$  (or  $M^T M$ ) eigenvalues*

# Singular Value Decomposition (SVD) - II

## Matrix approximation

- The best rank  $r$  approximation  $M'$  of a matrix  $M$ . (minimizing the [Frobenius norm](#))

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

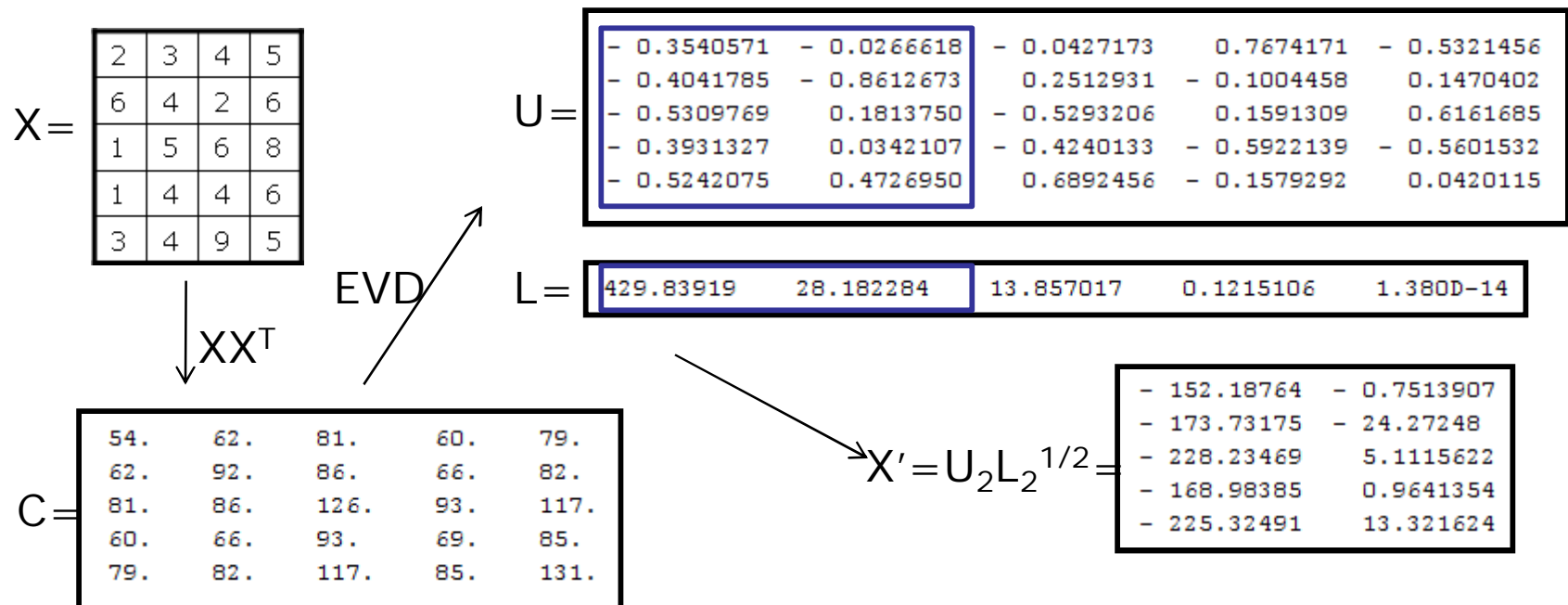
- where  $A^H$  [transpose](#) of  $A$ ,  $\sigma_i$  are the [singular values](#) of  $A$ , and the [trace function](#) is used.
- The Frobenius norm is sub-multiplicative and is very useful for [numerical linear algebra](#). This norm is often easier to compute than induced norms.
- $M' = U\Sigma'V^*$  ( $\Sigma'$  keeps the  $r$  largest singular values from  $\Sigma$ )

# Multidimensional Scaling (MDS)

- Initial we depict vectors in random places
- Iteratively reposition them in order to minimize Stress.
  - $\text{Stress} = \frac{\sum (f(d_{ij}) - d_{ij}')^2}{\sum f(d_{ij})^2}$
  - 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

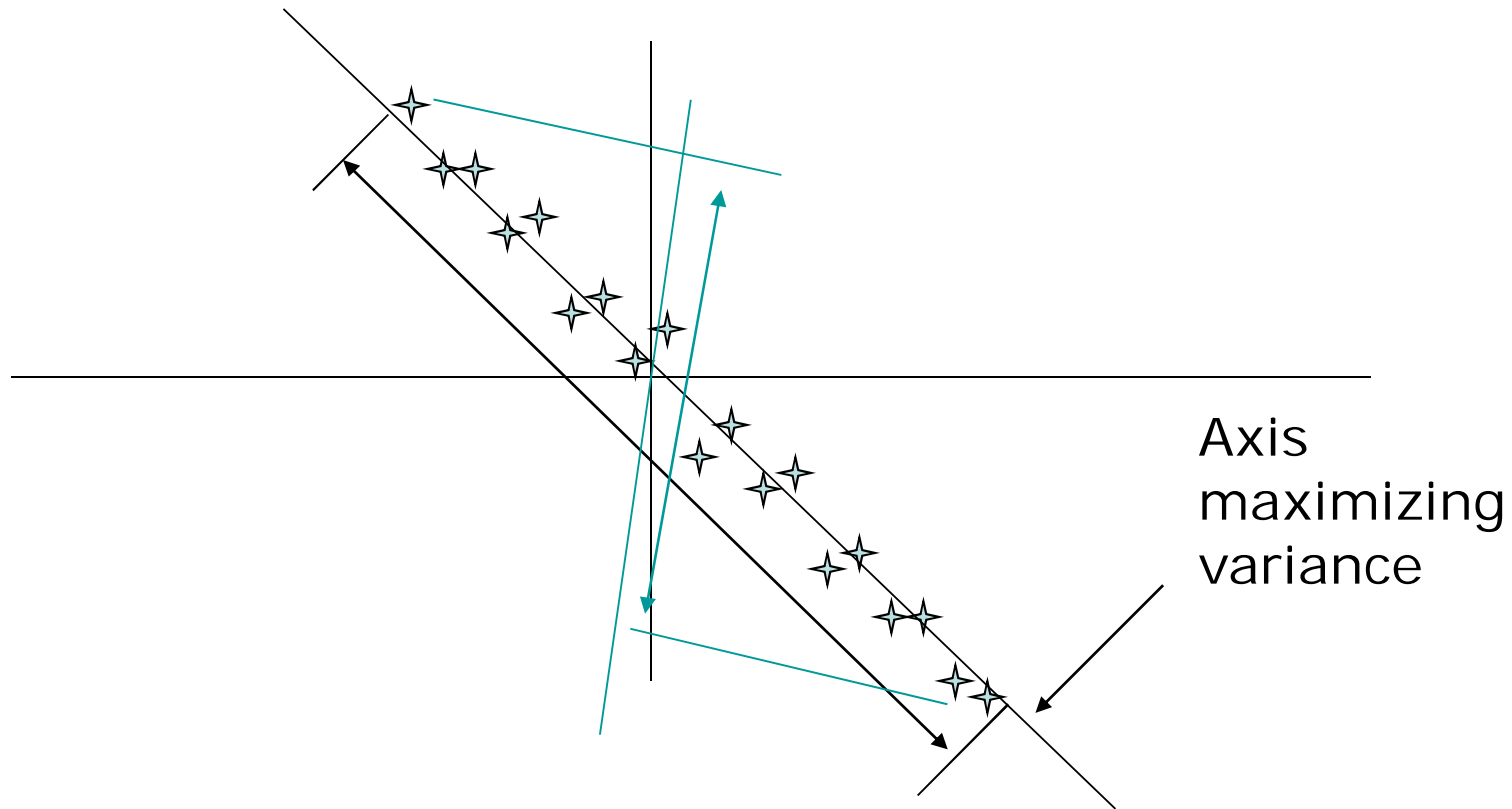


# Principal Components Analysis

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



# 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:  $x_1, \dots, x_n$
- The objective is to project the data to k dimensions via some linear decomposition:  
$$y_1 = a_1 * x_1 + \dots + a_n * x_n$$

.....

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

# Covariance Matrix

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

- covariance matrix  $\Sigma$  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:  $\text{cov}(X) = XX^T$

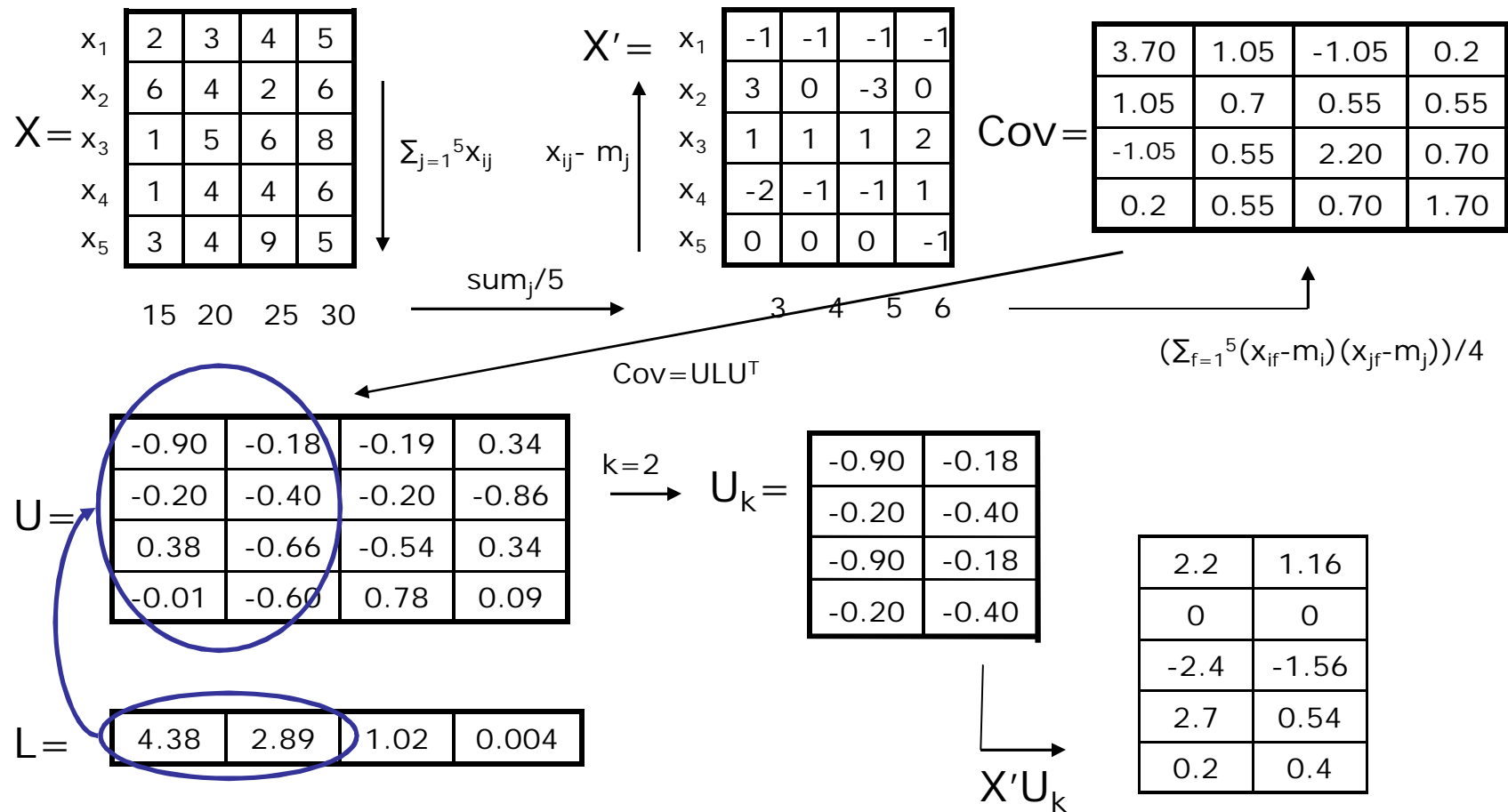
# Principal Components Analysis (PCA)

- 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  $X_{p \times n}$
  - We subtract mean values from columns.  $X' = (X - M)$
  - Calculate covariance matrix  $W = X'^T X'$

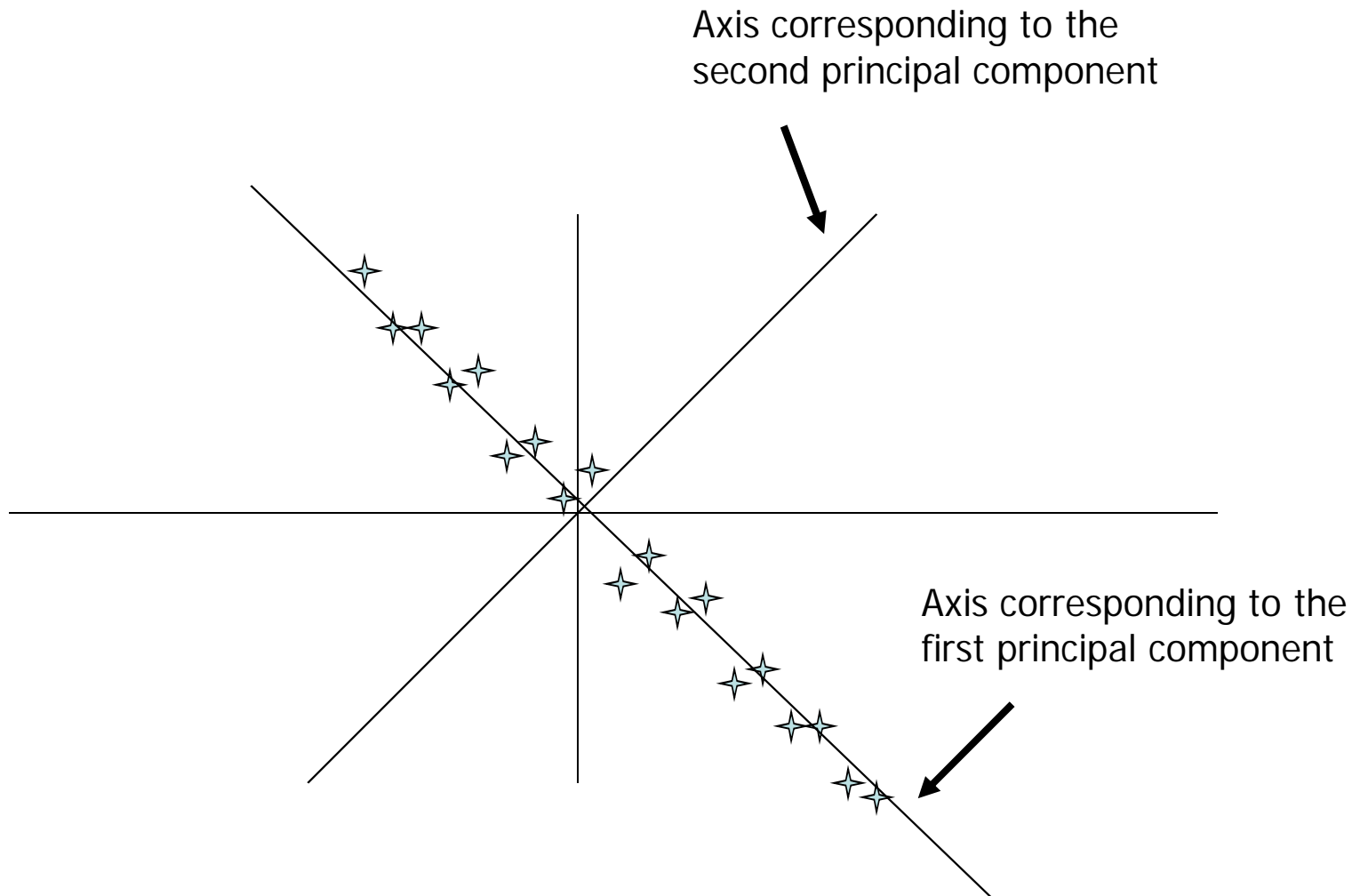
# Principal Components Analysis (PCA) – (2)

- Calculation of covariance matrix ( $W$ )
  - A matrix  $n \times n$ , in each cell of which  $(i,j)$  we have the covariance of  $X_i, X_j$ .
- Calculate eigenvalues and eigenvectors of  $\mathbf{W} (\mathbf{X}, \mathbf{D}) = \mathbf{U} \mathbf{A} \mathbf{U}^T$
- 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 Applications

- Preprocessing step preceding the application of data mining algorithms (such as clustering).
- Data Visualization.
- Noise reduction.

# PCA, synopsis

- 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.

# 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  $k \times k$  containing the top-k singular values of the diagonal in matrix L,
- $U_k$ , the  $m \times k$  matrix containing the first k columns in U (left singular vectors)
- $V_k^T$ , the  $k \times n$  matrix containing the first k lines of  $V^T$  (right singular vectors)

Typical values for  $k \sim 200-300$  (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  $A_k = U_k L_k V_k^T$
- 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.





# 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

$$A = ULV^T$$

$V =$

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

$$q^T = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U_k = \begin{bmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{bmatrix}$$

$$L_k^{-1} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.39 \end{bmatrix}$$

$$q_n = q^T U_k L_k^{-1} = \begin{bmatrix} 0.138 & -0.0273 \end{bmatrix}$$

# Query transformation

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

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

3.34	0
0	2.54

=

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

$q_n L_k =$

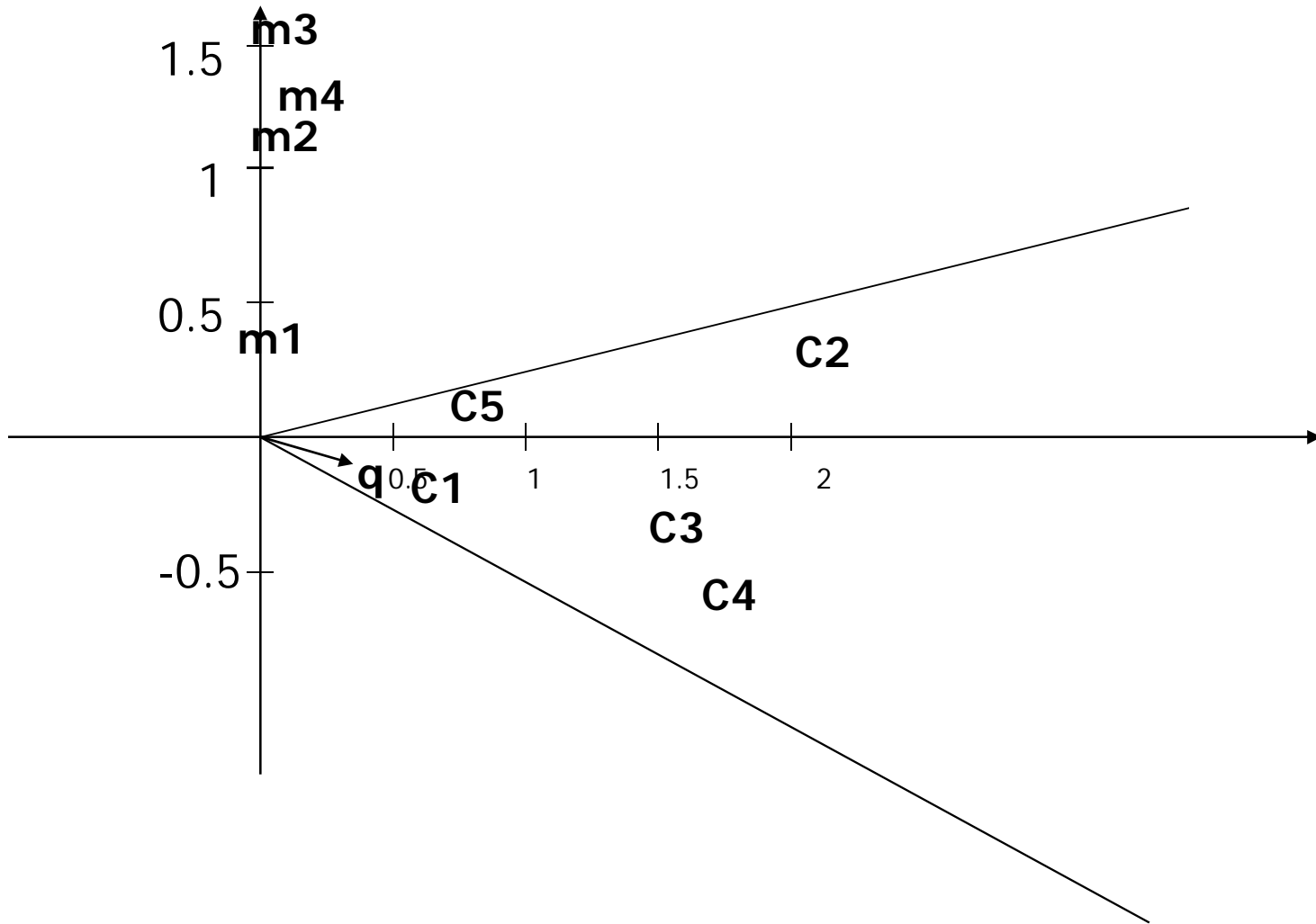
0.138	-0.0273
-------	---------

3.34	0
0	2.54

=

0.46	-0.069
------	--------

# Query transformation



# Query transformation

- Comparison of the transformed query to the new document vectors based on cosine similarity, where the similarity is computed as:  $\text{Cos}(x,y) = \frac{\langle x,y \rangle}{\|x\| \cdot \|y\|}$

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

$$\langle x,y \rangle = x_1 * y_1 + \dots + x_n * y_n$$

$$\|x\| = \text{sqrt}(\langle x,x \rangle)$$

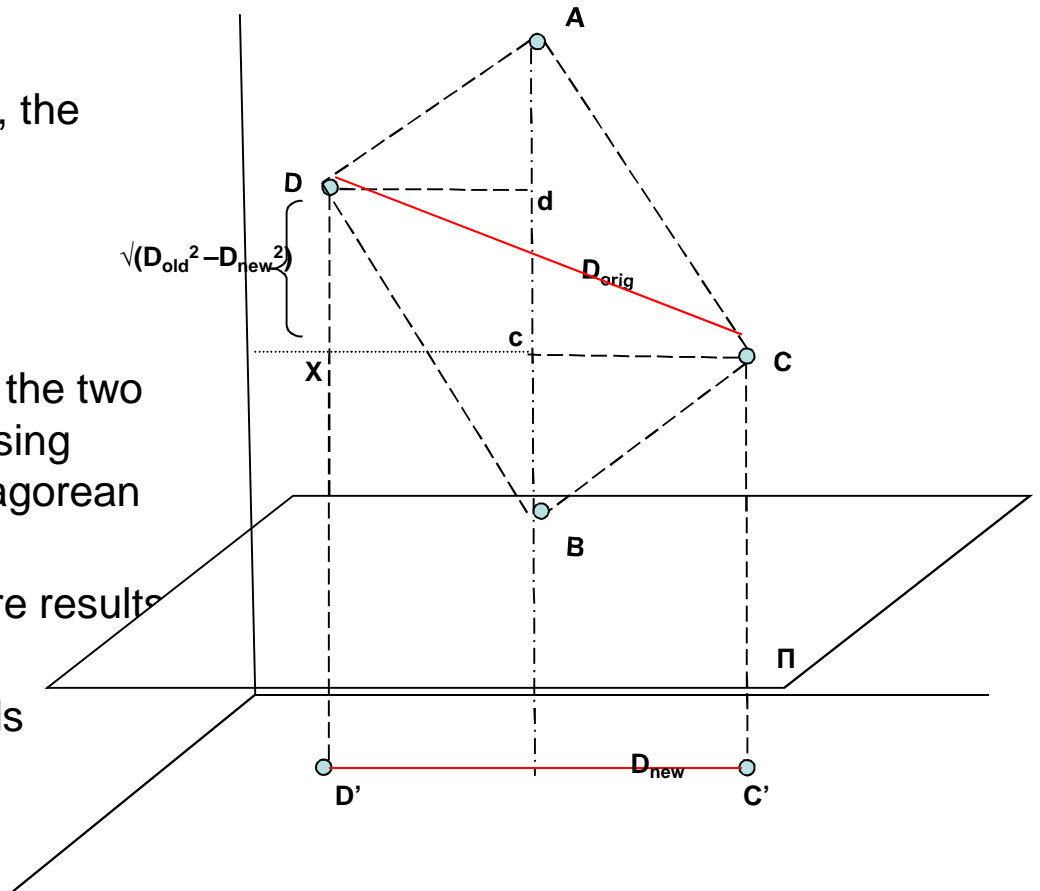
# Query transformation

- 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

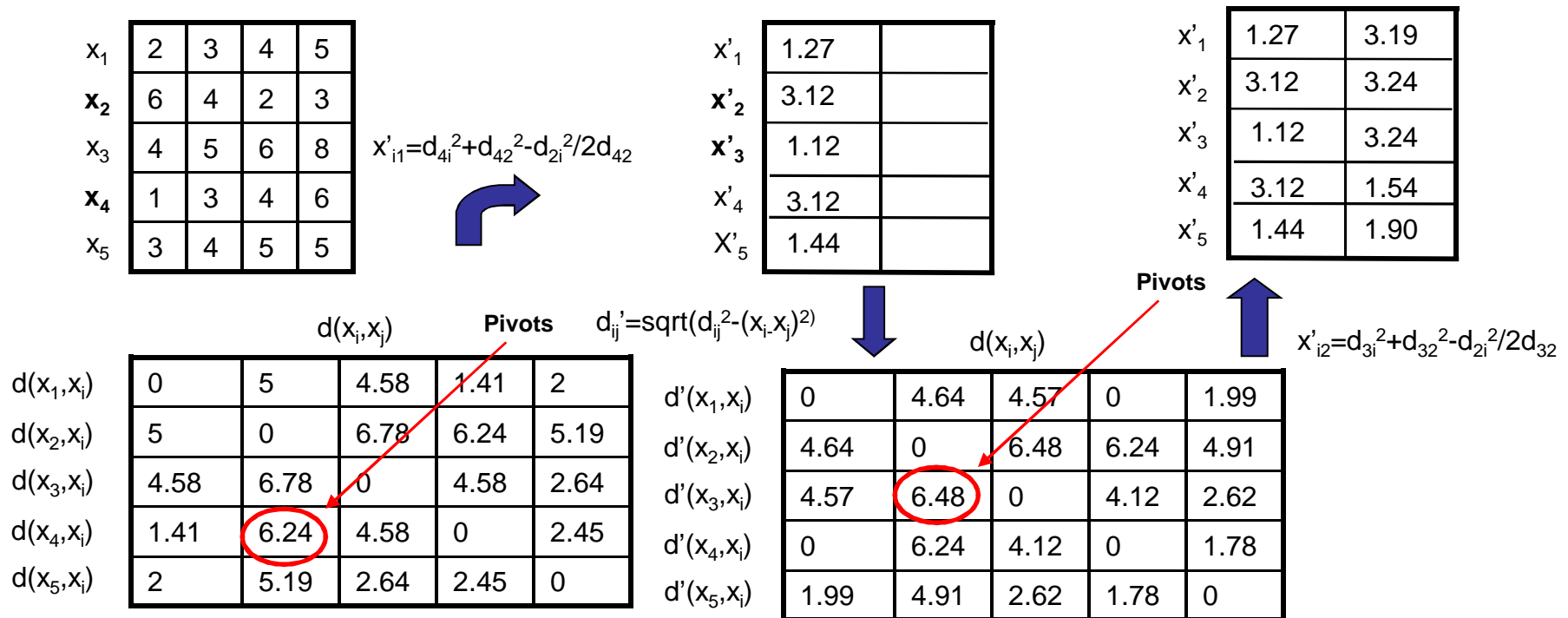
# FastMap

- FastMap (*Faloutsos et al. 1995*)
  - Input: Only distances between data, the projection dimensionality
  - Output: The projected dataset
  - Projects all data to a hyperplane perpendicular to the line defined by the two most distant points of the dataset using simply the cosine law and the Pythagorean theorem
  - Iterative application of this procedure results in the projection of data.
  - One of the fastest available methods
  - Algorithmic complexity:  $O(Nk)$



# FastMap

- Be careful with numeric zero:
  - When checking for  $X-Y=0$  then implement it as  $|X-Y|<\epsilon$ ,  $\epsilon=10^{-7}$
- Be careful while choosing pivots
  - Pivots are chosen with Choose-Distant-Objects



# Random Projections

- One of the simplest methods in available bibliography
- New data is acquired by multiplication:
  - $B_{p \times k} = (1/\sqrt{k}) X_{p \times n} W_{n \times k}$
- $W_{n \times k}$  is defined by one of the following distributions:
  - $w_{ij} = -1$  or  $1$  with probability  $1/2$
  - $w_{ij} = -\sqrt{3}$  or  $0$  or  $\sqrt{3}$  with probability  $1/6, 2/3, 1/6$
- Distance variation is bounded due to the Johnson-Lindenstrauss theorem
  - Given a real number  $\epsilon$ ,  $0 < \epsilon < 1$  and 2 positive integers  $k, n$  for which  $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \ln(n)$  then for every set of  $n$  points in  $R^d$  there exists  $f: R^d \rightarrow R^k$  such that for every  $u, v$  we have:  $(1-\epsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2$
- Complexity:  $O(pnk)$

# Random Projections

- Define array  $W_{n \times k}$ 
  - Possibility with distribution  $1/2 (1, -1)$   
 $W[i][j] = (-1)^{\text{randomInteger}\%2}$
  - Possibility with distribution  $1/6 (\sqrt{3}, 0, -\sqrt{3})$   
 If  $(\text{randomInteger}\%3=0)$   
 $W[i][j] = \sqrt{3} * (-1)^{\text{randomInteger}\%2}$   
 else  
 $W[i][j]=0$

$$(-1)^{213\%2}$$

