Linear Algebra Methods for Data Mining

Saara Hyvönen, Saara.Hyvonen@cs.helsinki.fi

Spring 2007

Overview of some topics covered and some topics not covered on this course

Linear algebra tool kit

- QR iteration
- eigenvalues, eigenvalue decomposition, generalized eigenvalue problem
- singular value decomposition SVD
- NMF
- power method (for finding eigenvalues and -vectors)

Data mining tasks encountered

- regression
- classification
- clustering
- finding latent variables
- visualizing and exploration
- ranking

QR was used for...

- orthogonalizing a set of (basis) vectors $\mathbf{X} = \mathbf{QR}$.
- solving the least-squares problem:

$$\|\mathbf{r}\|^2 = \|\mathbf{b} - \mathbf{Ax}\|^2 = \|\mathbf{Q}^T\mathbf{b} - \begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix}\mathbf{x}\|^2 = \|\mathbf{b}_1 - \mathbf{Rx}\|^2 + \|\mathbf{b}_2\|^2.$$

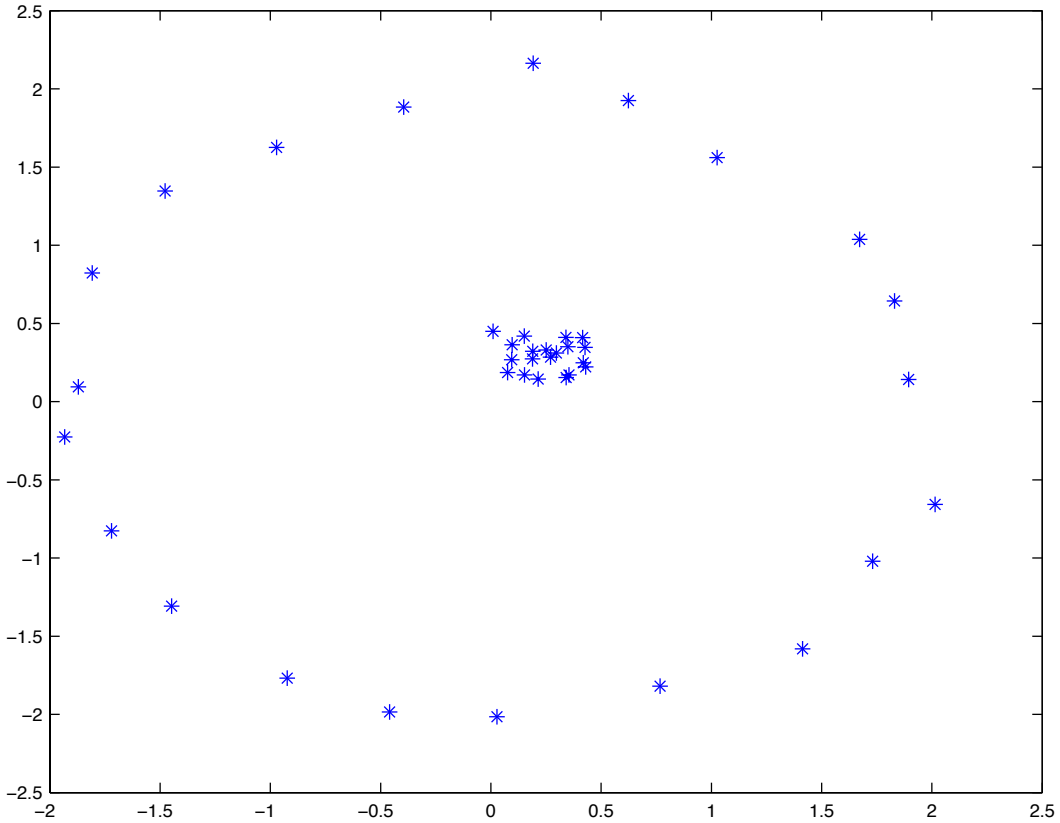
- least squares problems were encountered e.g. when we wish to express a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ in terms of a set of basis vectors $\mathbf{X} \in \mathbb{R}^{m\times k}$, $k < m$.

Eigenvalues/vectors were encountered in...

- PageRank: eigenvector corresponding to largest eigenvalue of the Google matrix.

- Linear discriminant analysis:
  linear discriminants $=$ eigenvectors corresponding to the largest eigenvalues of the generalized eigenvalue problem $\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w}$.

- Spectral clustering: based on running k-means clustering on the matrix obtained from the eigenvectors corresponding to the largest eigenvalues of the graph laplacian matrix $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.


2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5

Spectral clustering

Use methods from spectral graph partitioning to do clustering.

Needed: pairwise distances between data points.

These can be thought of as weights of links in a graph: clustering problem becomes a graph partitioning problem.

Unlike k-means, clusters need not be convex.

Algorithm

We have n data points $(\mathbf{x}_1, \ ..., \ \mathbf{x}_n)$.
We wish to partition them into k disjoint clusters $C_1, ..., C_k$.

1. Form affinity matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ defined by

$$\mathbf{A}_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2) & \text{if} \quad i \neq j \\ = 0 & \text{if} \quad i = j. \end{cases}$$

2. Define $\mathbf{D}$ to be the diagonal matrix whose i^{th} diagonal element is the sum of $\mathbf{A}$'s i^{th} row, and construct the matrix

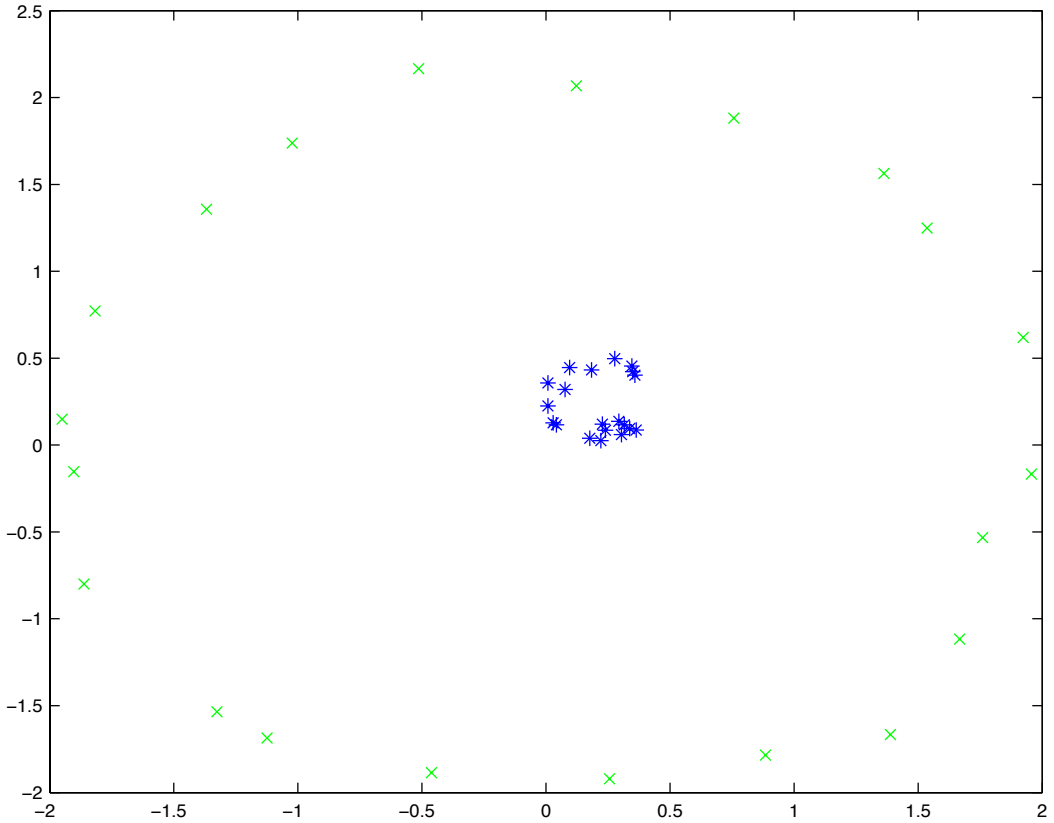
$$\mathbf{L} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}.$$

3. Find the eigenvectors $\mathbf{v}_j$ of $\mathbf{L}$ corresponding to the k largest eigenvalues, and form the matrix $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ ...\ \mathbf{v}_k] \in \mathbb{R}^{n \times k}$.

4. Form the matrix $\mathbf{Y}$ from $\mathbf{V}$ by renormalizing each of $\mathbf{V}$'s *rows* to have unit length.

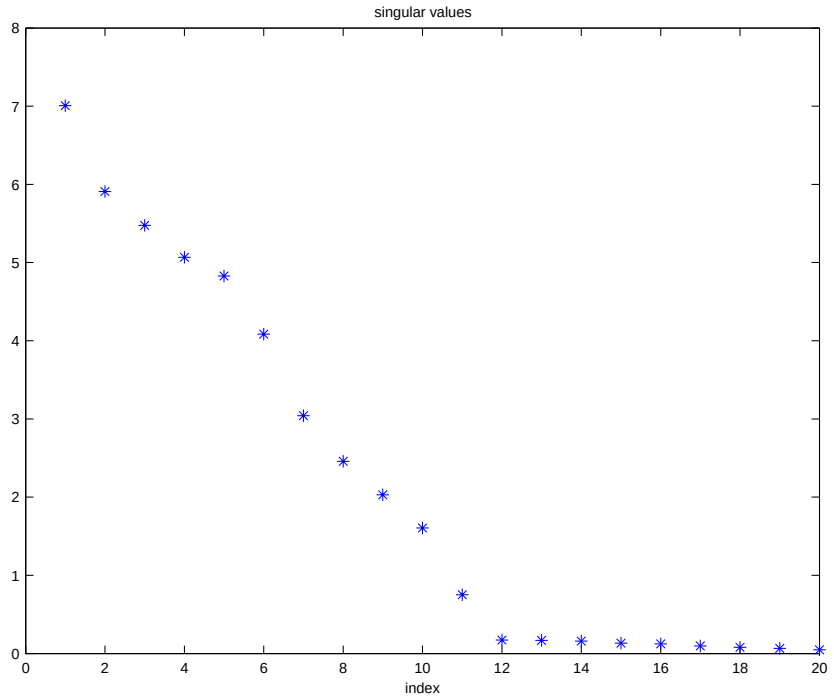
5. Treating each row of $\mathbf{Y}$ as a point in $\mathbb{R}^k$, cluster them into k clusters via k-means (or any other clustering algorithm).

6. If the row i of the matrix $\mathbf{Y}$ was assigned to cluster j, assign the data point $\mathbf{x}_i$ to the cluster j.


2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2

SVD was useful for...

- noise reduction

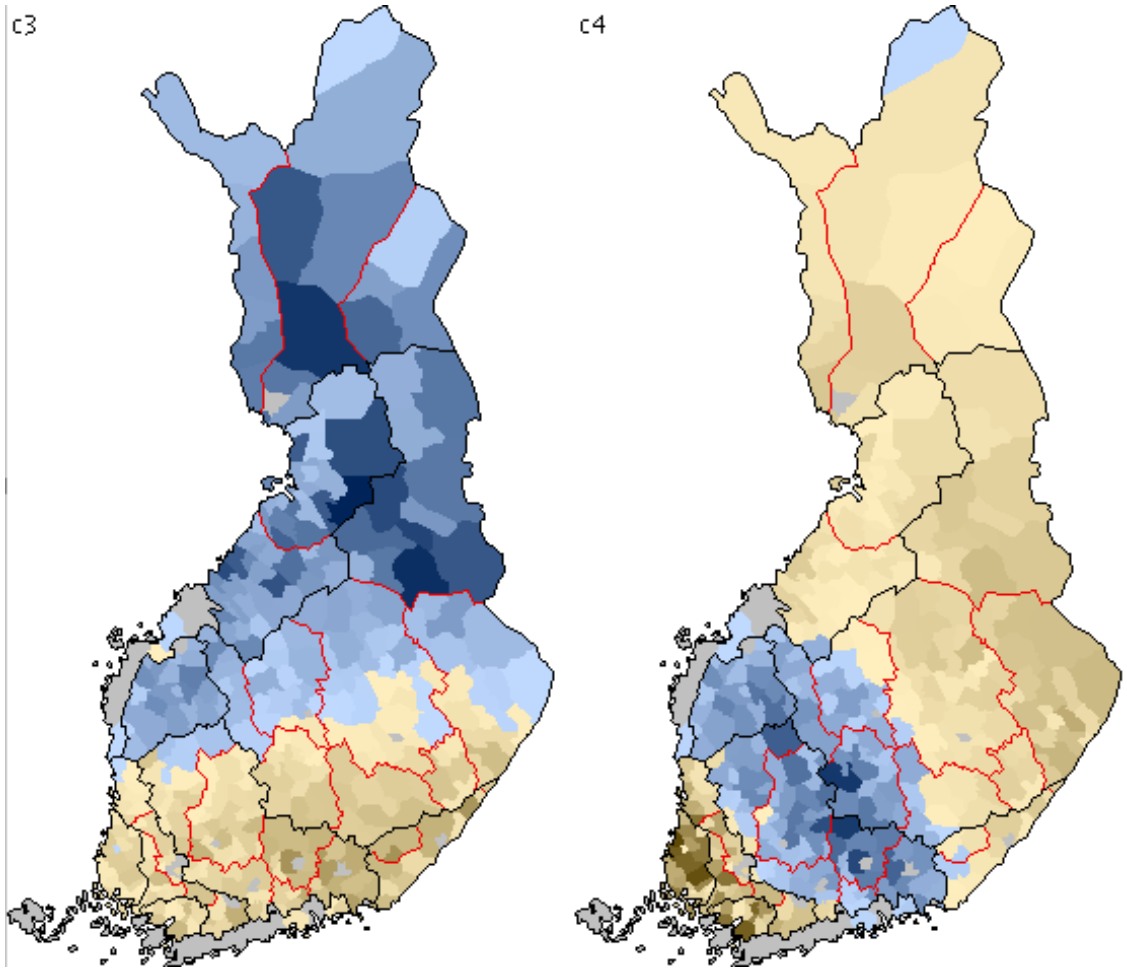


- data compression

If $\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$, where $\mathbf{\Sigma}_k$ contains the k first singular values of $\mathbf{A}$, and the columns of $\mathbf{U}_k$ and $\mathbf{V}_k$ are the corresponding (left and right) singular vectors, then

$$\min_{\mathsf{rank}(\mathbf{B}) \leq k} \|\mathbf{A} - \mathbf{B}\|_2 = \|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}.$$

- visualizing data: PCA



- information retrieval, LSI

$\mathbf{A}$ term-to-document matrix, $\mathbf{q}$ query. Instead of doing query matching $\mathbf{q}^T\mathbf{A} > tol$ in the full space, do SVD on $\mathbf{A}$ and use only the k first singular values/vectors.

Result: compression plus (often) better performance in terms of precision vs recall.

- HITS

The HITS algorithm distinguishes between *authorities*, which contain high-quality information, and *hubs* which are comprehensive lists of links to authorities.

Form the adjacency matrix of the directed web graph.

Hub scores and authority scores are the left and right singular vectors of the adjacency matrix.

Power method

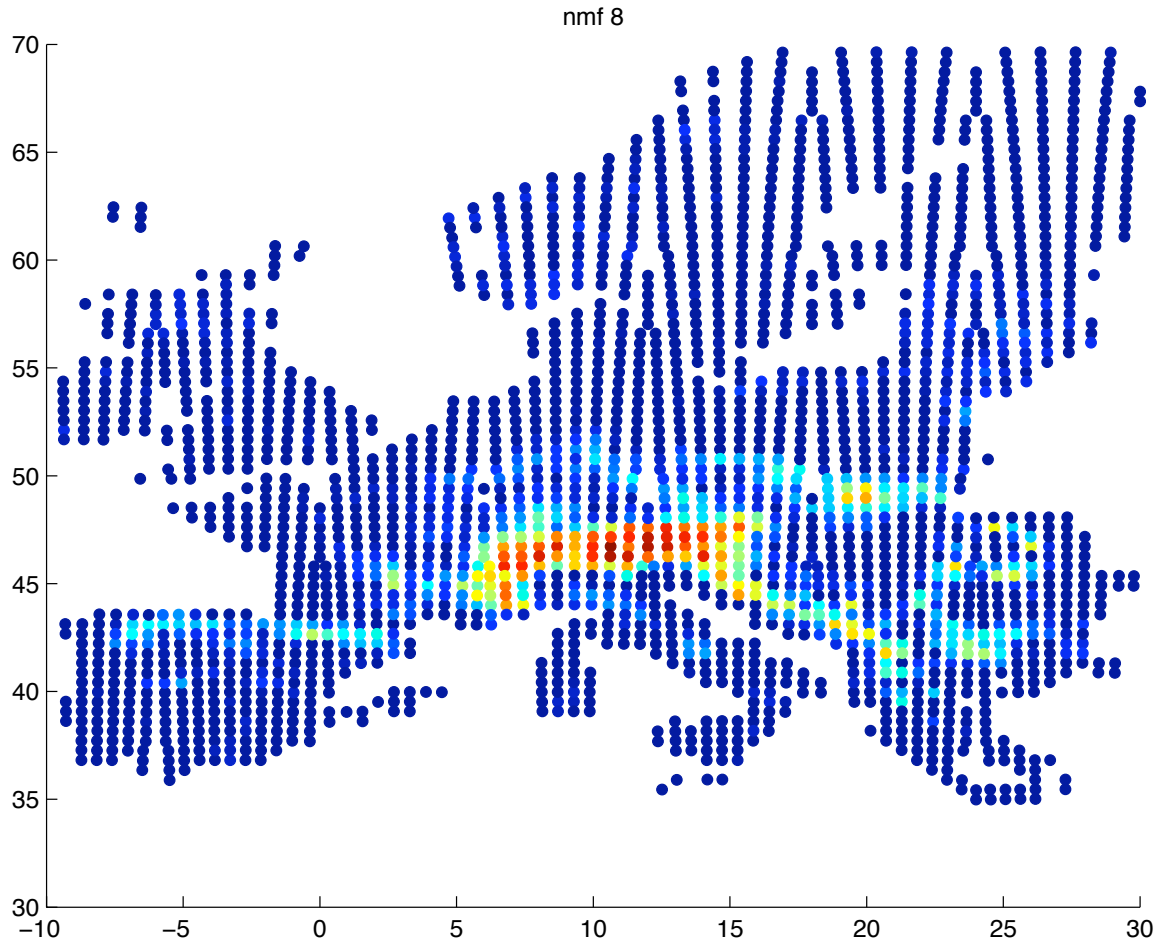
- is used to find the largest eigenvalue in magnitude and the corresponding eigenvector.
- PageRank
- subsequent eigenvalues/vectors could be found by using *deflation*. In the symmetric case:

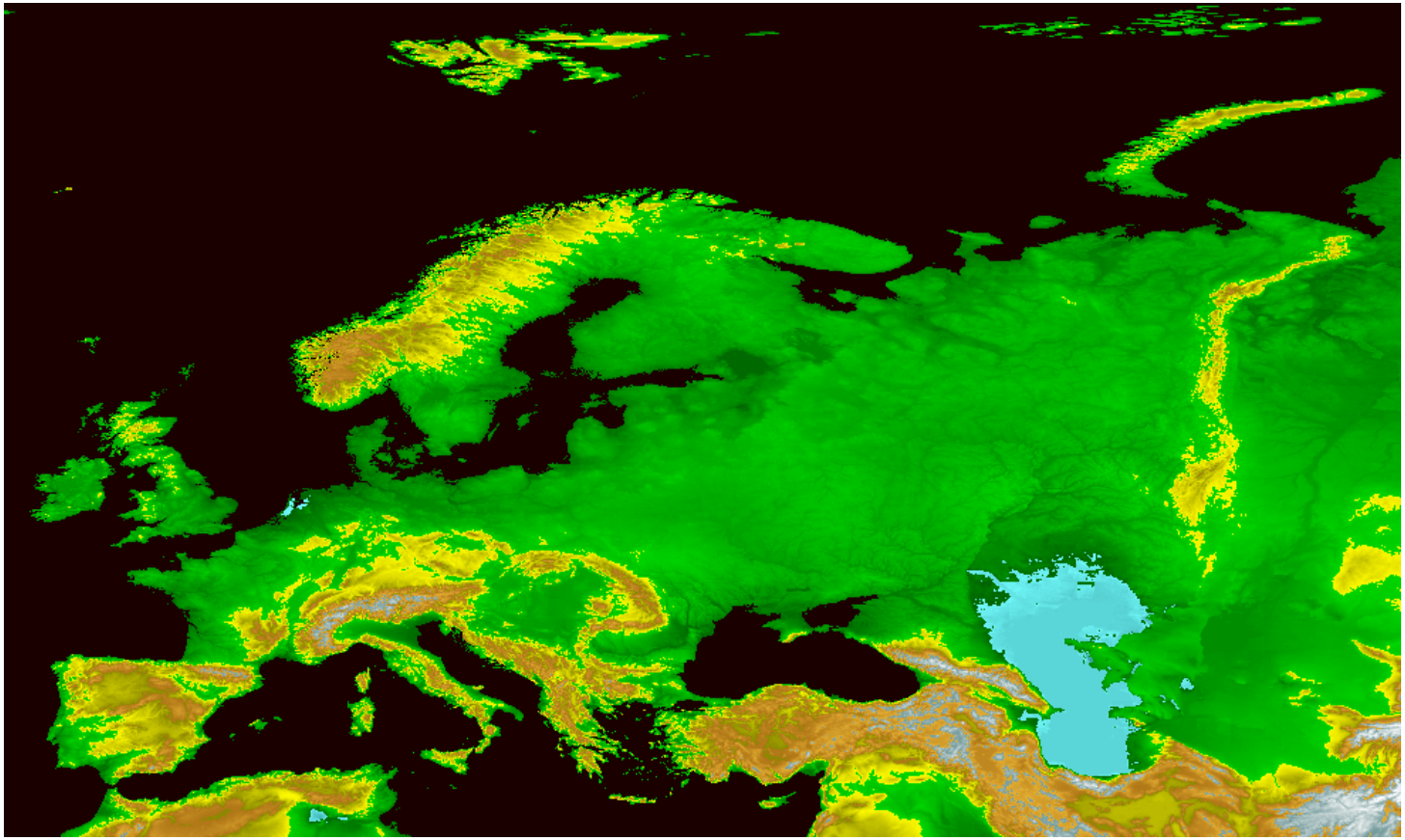
$$\mathbf{A} = \sum_{j=1}^{n} \lambda_j \mathbf{u}_j \mathbf{u}_j^T, \quad \hat{\mathbf{A}} = \mathbf{A} - \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T.$$

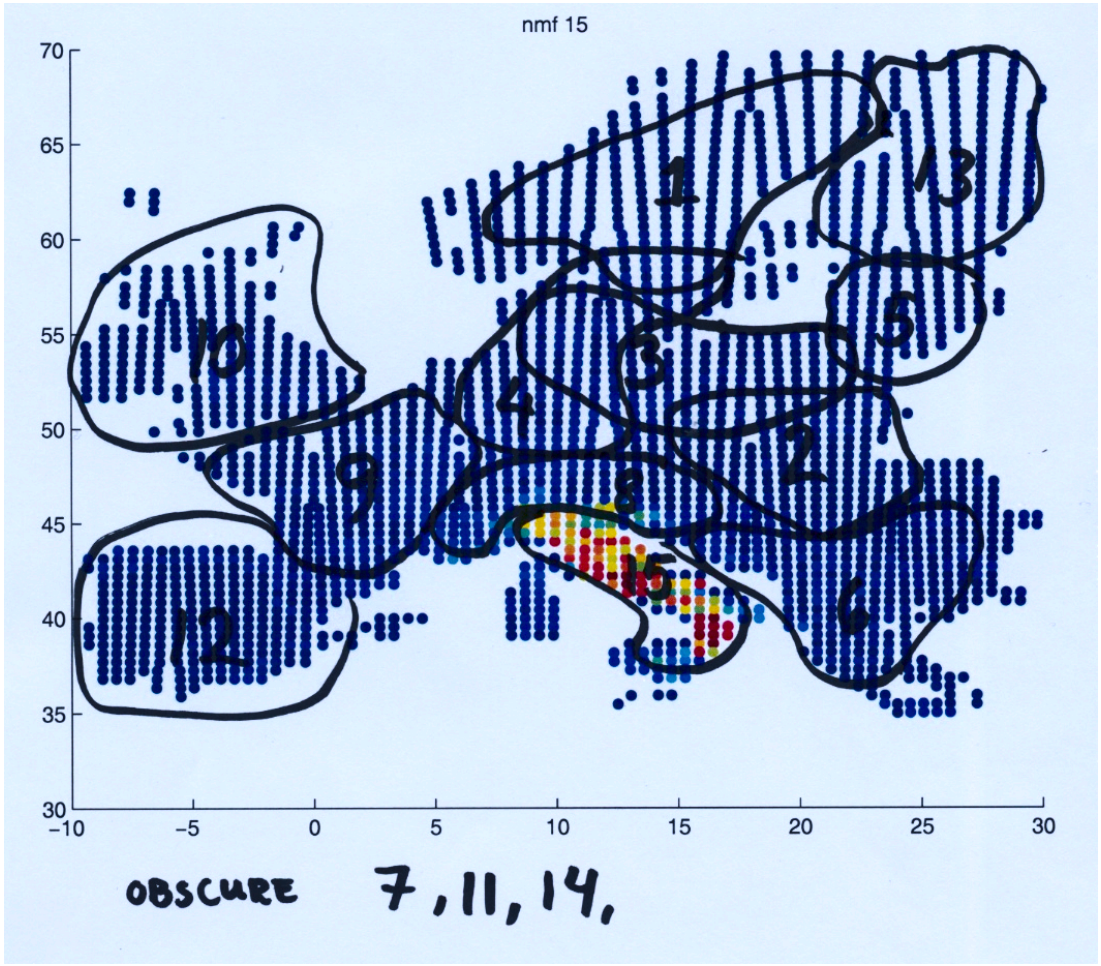
Nonnegative matrix factorization

Given a nonnegative matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we wish to express the matrix as a product of two nonnegative matrices $\mathbf{W} \in \mathbb{R}^{m \times k}$ and $\mathbf{H} \in \mathbb{R}^{k \times n}$:

$$\mathbf{A} \approx \mathbf{WH}$$


nmf 8
70
65
60
55
50
45
40
35
30
−10
−5
0
5
10
15
20
25
30




nmf 15
70
65
60
55
50
45
40
35
30
−10
−5
0
5
10
15
20
25
30
1
13
10
5
3
4
2
9
8
15
12
6
OBSCURE
7, 11, 14,

Roaming beyond the scope of this course

- There are plenty of thing related to linear algebra and data mining that we did not cover on this course, e.g.
- tensor SVD, generalized SVD
- kernel methods
- independent component analysis ICA
- multidimensional scaling
- canonical correlations

- generalized linear models
- factor analysis, mPCA,...
- spectral ordering
- ...

Tensors

- vector: one-dimensional array of data
- matrix: two-dimensional array of data
- tensor: n-dimensional array of data, e.g. n=3: $\mathbf{A} \in \mathbb{R}^{l \times m \times n}$
- it is possible to define Higher Order SVD for such a 3-mode array or tensor.
- psychometrics, chemometrics

Face recognition using Tensor SVD

- collection of images of n_p persons
- each image is an $m_1 \times m_2$ array: stack columns to get vector of length $n_i = m_1 m_2$.
- each person has been photographed with n_e different expressions/illuminations.
- so we have a tensor $\mathbf{A} \in \mathbb{R}^{n_i \times n_e \times n_p}$
- HOSVD can be used for face recognition, or e.g. reducing the effect of illumination.

Data: Digitized images of 10 people, 11 expressions.

Task: Find from the data base the closest match to the given figures (top row).

Below: closest match using HOSVD. In each case, the right person was identified.

Independent Component Analysis

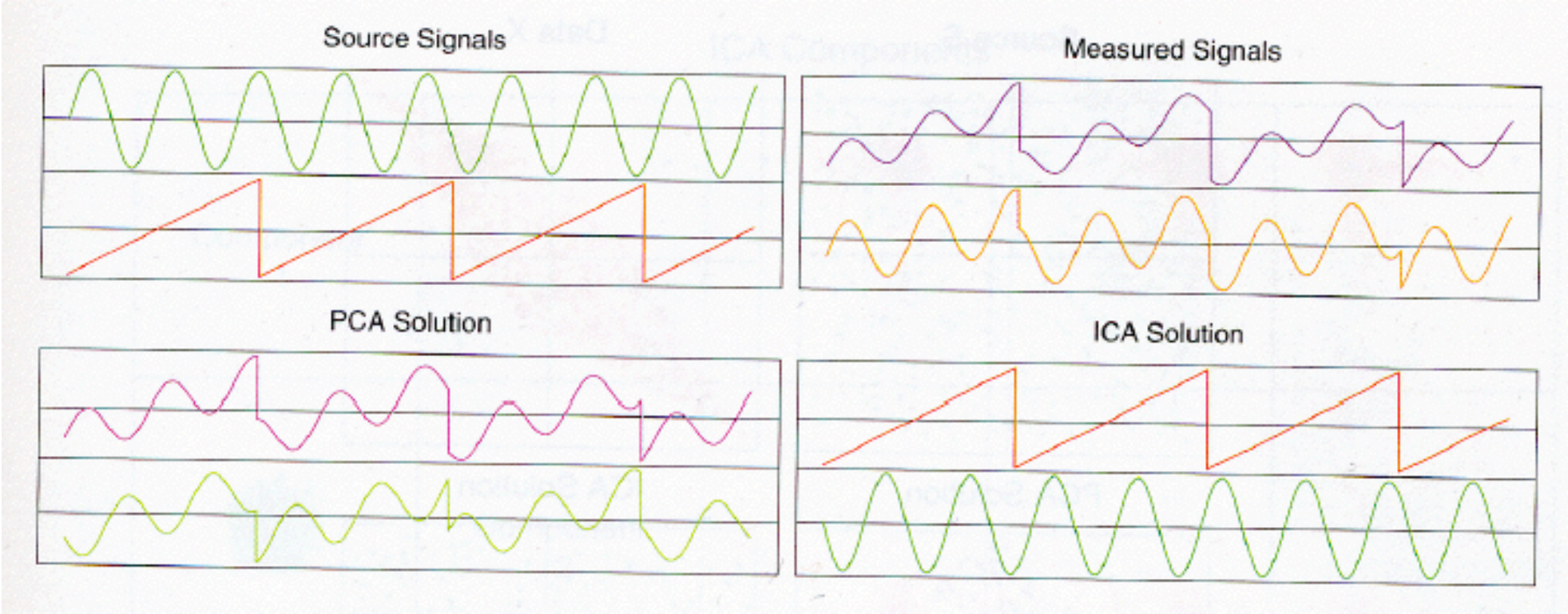
Consider the cocktail-party problem: in a room, you have two people speaking simultaneously, and two microphones recording the mixture of these speech signals.

Each recording is a weighted sum of the speech signals $s_1(t)$ and $s_2(t)$:

$$x_1(t) = a_{11}s_1 + a_{12}s_2$$
$$x_2(t) = a_{21}s_1 + a_{22}s_2$$

where a_{ij} are some parameters depending on the distances of the microphones from the speakers.

How to recover the original signals s_1 and s_2 from the recorded signals x_1 and x_2?



From Hastie, Tibshirani, Friedman [5].

PCA versus ICA

- PCA gives uncorrelated components. In the cocktail-party problem this is not the right answer.
- ICA gives *statistically independent* components.
- Variables y_1 and y_2 are independent, if information on the value of y_1 does not give any information on the value of y_2, and vica versa.
- Note: data must be nongaussian!

Example: Image separation. Mixtures of images:

ICA produced the following images:

See also

www.cis.hut.fi/projects/ica/cocktail/cocktail_en.cgi

Kernel methods

Idea: take a mapping

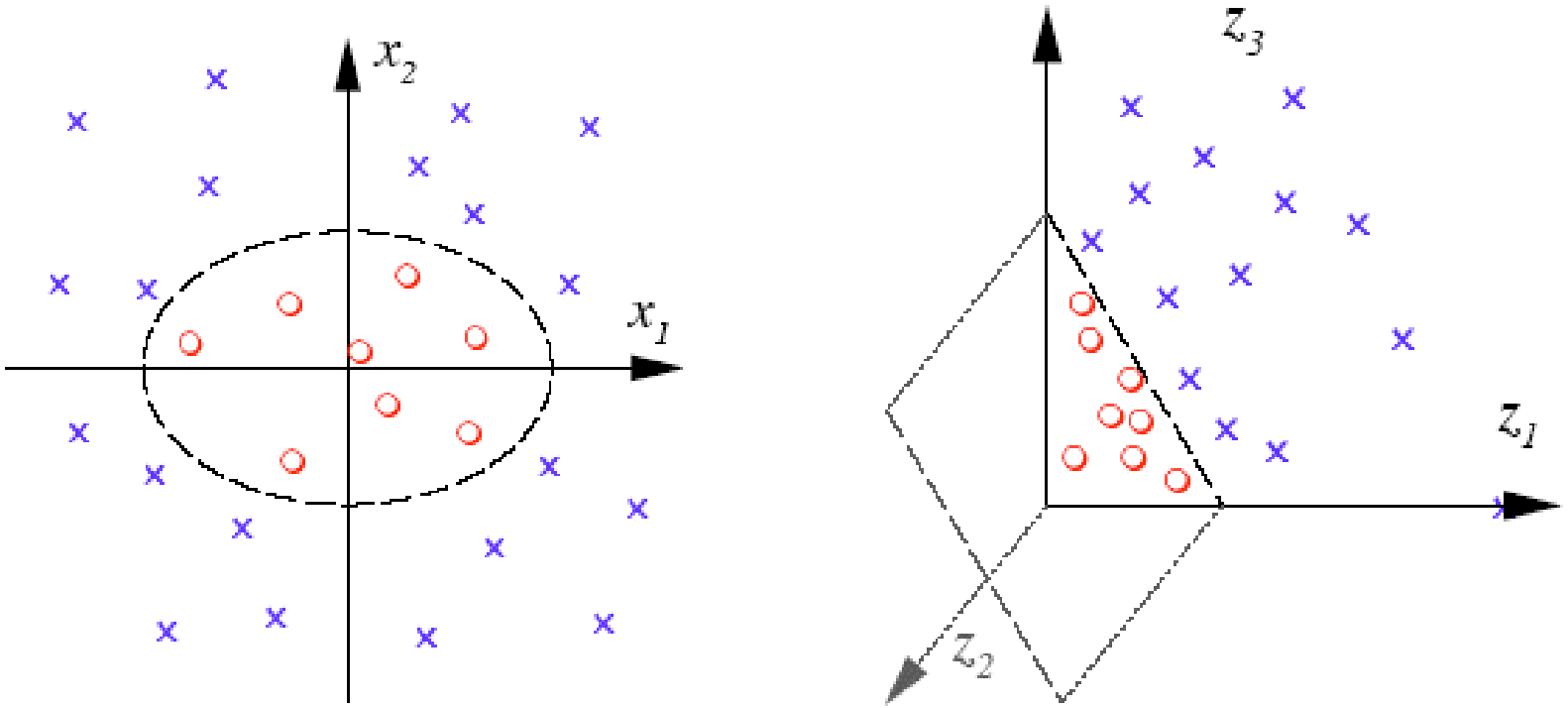
$$\phi : \mathcal{X} \to \mathcal{F},$$

where $\mathcal{F}$ is an inner product space, and map data $\mathbf{x}$ to the (higher dimensional) feature space:

$$\mathbf{x} \to \phi(\mathbf{x}).$$

Then work in the feature space $\mathcal{F}$.

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$



B. Schölkopf, NIPS, 3 December 2001

In this case:

$$\phi : (x_1, x_2) \rightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2).$$

So the inner product in the feature space is

$$\begin{aligned}\langle \phi(x), \phi(x') \rangle &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)({x'_1}^2, \sqrt{2}x'_1x'_2, {x'_2}^2)^T \\ &= \langle x, x' \rangle^2 =: k(x, x')\end{aligned}$$

So the inner product can be computed in $\mathbb{R}^2$!

Here k is the kernel function.

This is the very idea in kernel methods: you can operate in high dimensional feature spaces while doing all your (inner product) computations in a lower dimensional space. All you need is a suitable kernel.

A kernel is a function k such that for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$,

$$k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle,$$

where ϕ is a mapping from $\mathcal{X}$ to and (inner product) feature space $\mathcal{F}$.

There are numerous ways to define kernels.

We can use any algorithm that only depends on dot products: after the kernel trick, we are operating in the feature space.

In practice the dimension of the feature space can be huge.

If our data consists of images of size 16×16, and we use as a feature map polynomials of degree $d = 5$, then our feature space is of dimension 10^{10}!

Regardless of the dimension of the feature space, we can compute the inner products in the lower dimensional space: computation is not a problem.

Overfitting? Not a problem (in theory): for reasons, see the references.

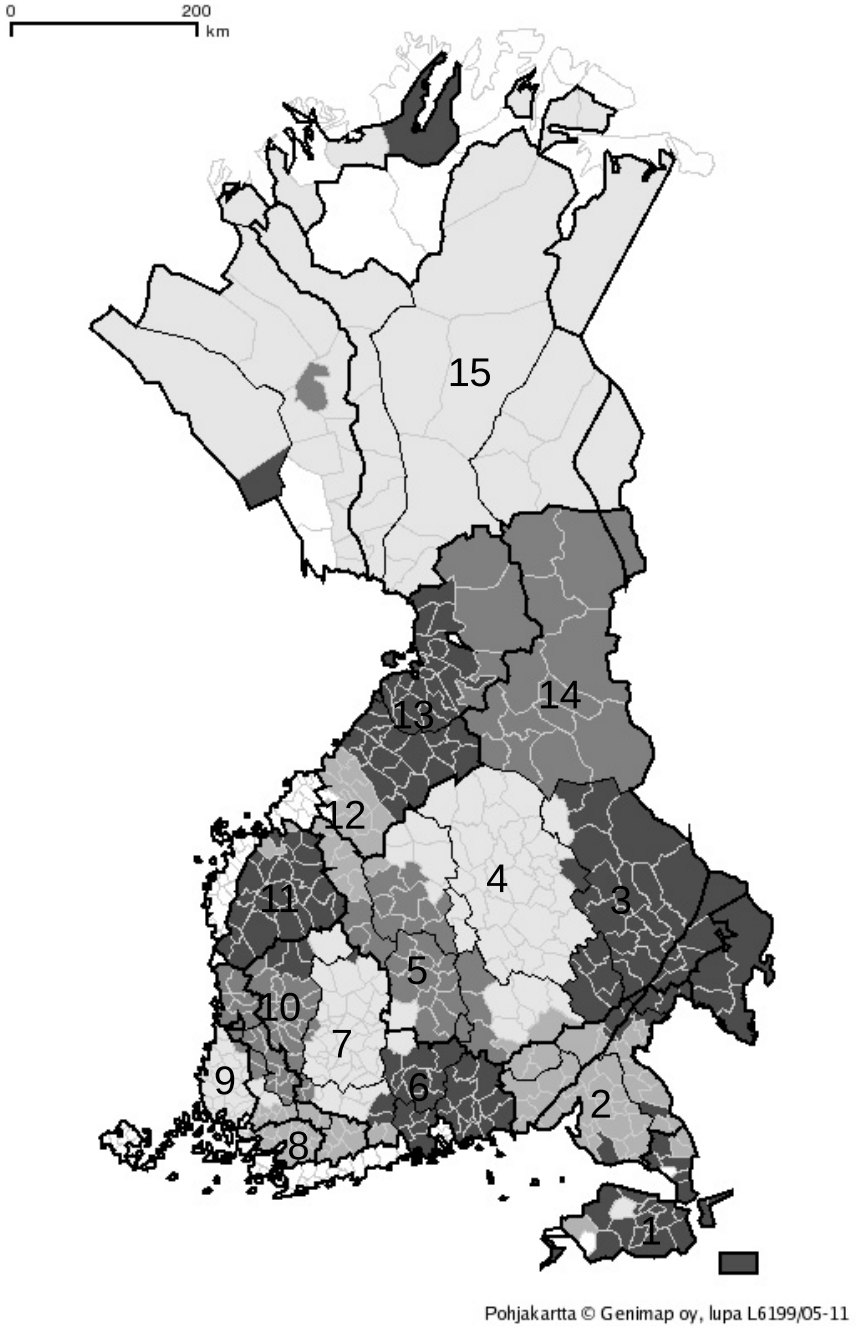
Kernel methods can be used for

- Pattern recognition
- classification (SVM= support vector machines)
- outlier detection,
- canoncial correlations,
- ...

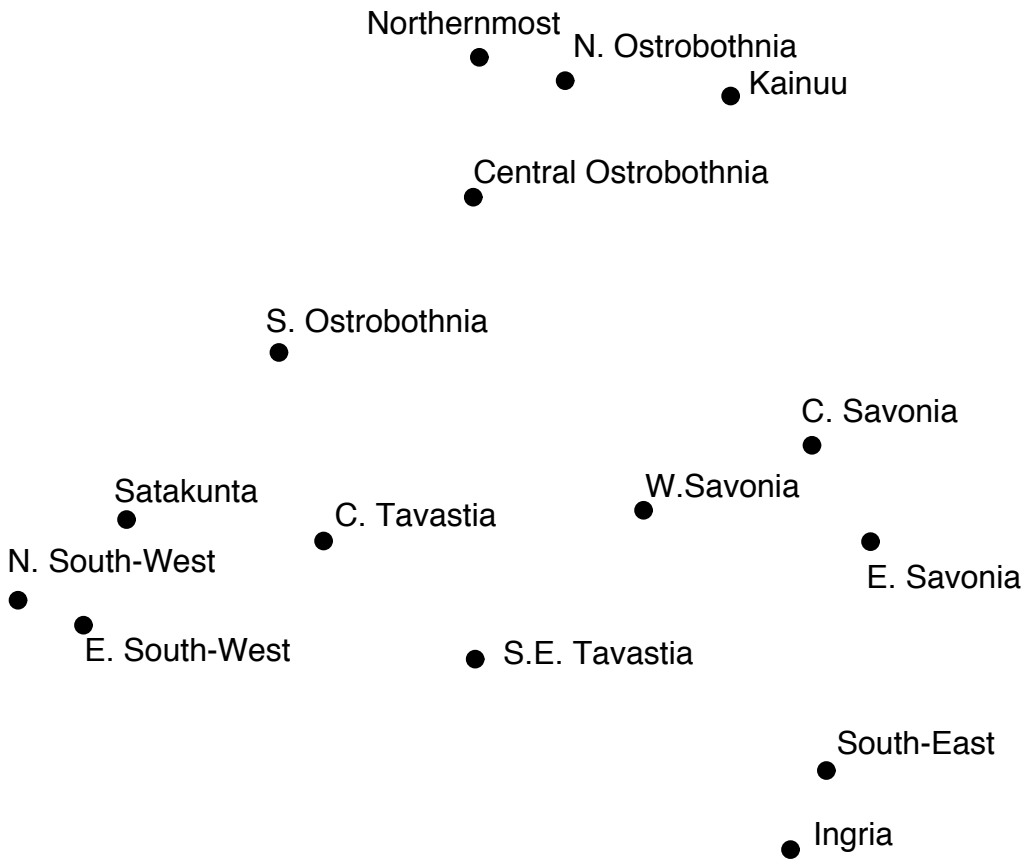
MDS

- uses pairwise distances between points

- finds a low dimensional representation of the data in such a way, that distances between points are preserved as well as possible



1. Ingria 2. South-East
3. E. Savonia
4. C. Savonia
5. W. Savonia
6. S.E. Tavastia
7. C. Tavastia
8. E. South-West
9. N. South-West
10. Satakunta
11. S. Ostrobothnia
12. C. Ostrobothnia
13. N. Ostrobothnia
14. Kainuu 15. Northernmost


Northernmost
N. Ostrobothnia
Kainuu
Central Ostrobothnia
S. Ostrobothnia
C. Savonia
Satakunta
C. Tavastia
W.Savonia
N. South-West
E. Savonia
E. South-West
S.E. Tavastia
South-East
Ingria

Final words

You can get far with a basic linear algebra toolkit.

But there remains a world of methods to explore!

References

[1] Lars Eldén: Matrix Methods in Data Mining and Pattern Recognition, SIAM 2007.

[2] A. Hyvärinen and E. Oja: Independent Component Analysis: Algorithms and Applications, Neural Networks 13 (4-5), 2000.

[3] J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004

[4] D. Lee and H. S. Seung, Learning the parts of objects with nonnegative matrix factorization, Nature 401, 788 (1999).

[5] T. Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer Verlag, 2001.

[6 D. Hand, H. Mannila, P. Smyth: Principles of Data Mining, MIT Press, 2001.