Introduction to Machine learning

Unsupervised learning

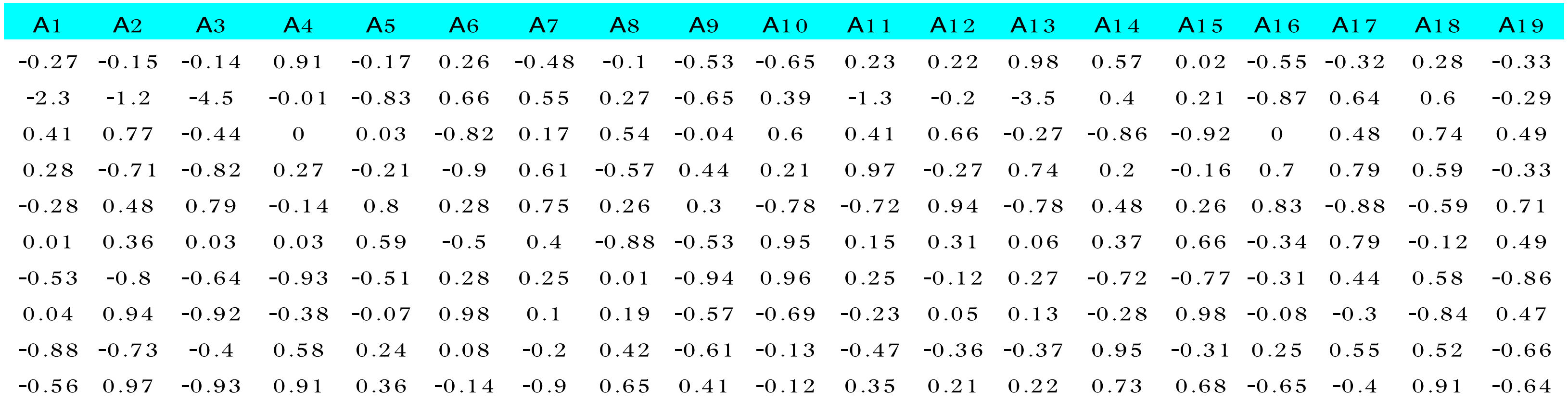
Pierre Geurts

Department of Electrical Engineering and Computer Science
University of Liège

November 27, 2018

Unsupervised learning

- Unsupervised learning tries to find any regularities in the data without guidance about inputs and outputs



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
-0.27	-0.15	-0.14	0.91	-0.17	0.26	-0.48	-0.1	-0.53	-0.65	0.23	0.22	0.98	0.57	0.02	-0.55	-0.32	0.28	-0.33
-2.3	-1.2	-4.5	-0.01	-0.83	0.66	0.55	0.27	-0.65	0.39	-1.3	-0.2	-3.5	0.4	0.21	-0.87	0.64	0.6	-0.29
0.41	0.77	-0.44	0	0.03	-0.82	0.17	0.54	-0.04	0.6	0.41	0.66	-0.27	-0.86	-0.92	0	0.48	0.74	0.49
0.28	-0.71	-0.82	0.27	-0.21	-0.9	0.61	-0.57	0.44	0.21	0.97	-0.27	0.74	0.2	-0.16	0.7	0.79	0.59	-0.33
-0.28	0.48	0.79	-0.14	0.8	0.28	0.75	0.26	0.3	-0.78	-0.72	0.94	-0.78	0.48	0.26	0.83	-0.88	-0.59	0.71
0.01	0.36	0.03	0.03	0.59	-0.5	0.4	-0.88	-0.53	0.95	0.15	0.31	0.06	0.37	0.66	-0.34	0.79	-0.12	0.49
-0.53	-0.8	-0.64	-0.93	-0.51	0.28	0.25	0.01	-0.94	0.96	0.25	-0.12	0.27	-0.72	-0.77	-0.31	0.44	0.58	-0.86
0.04	0.94	-0.92	-0.38	-0.07	0.98	0.1	0.19	-0.57	-0.69	-0.23	0.05	0.13	-0.28	0.98	-0.08	-0.3	-0.84	0.47
-0.88	-0.73	-0.4	0.58	0.24	0.08	-0.2	0.42	-0.61	-0.13	-0.47	-0.36	-0.37	0.95	-0.31	0.25	0.55	0.52	-0.66
-0.56	0.97	-0.93	0.91	0.36	-0.14	-0.9	0.65	0.41	-0.12	0.35	0.21	0.22	0.73	0.68	-0.65	-0.4	0.91	-0.64

- Are there interesting groups of variables or samples? outliers? What are the dependencies between variables?

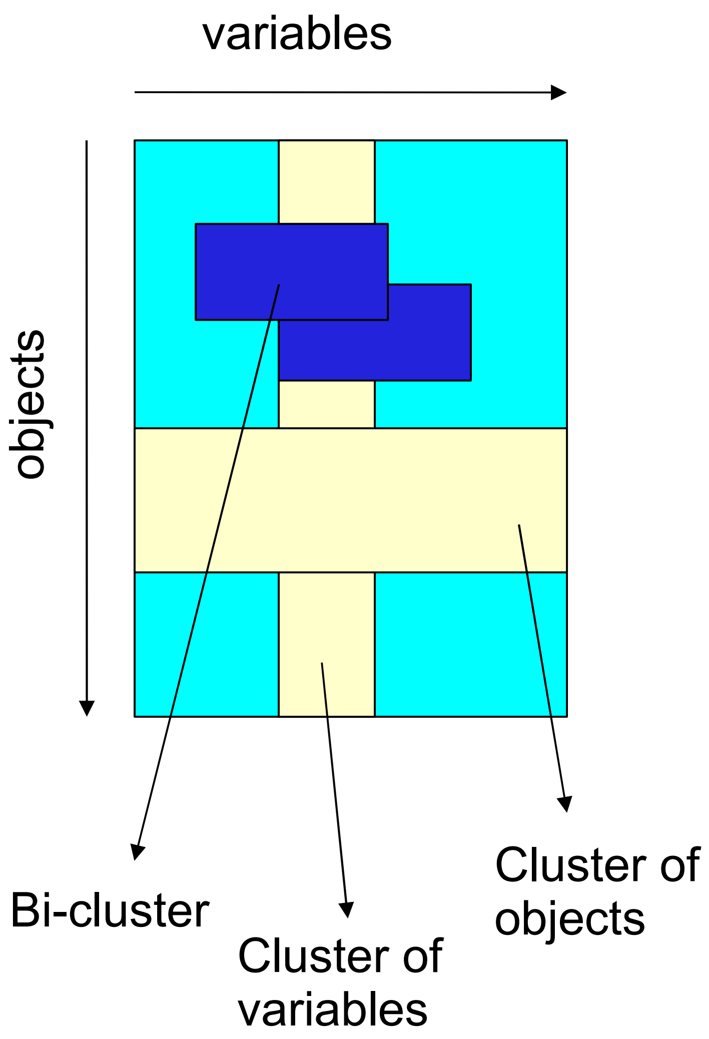
Unsupervised learning methods

- Many families of problems exist, among which:
  - Clustering: try to find natural groups of samples/variables
    - eg: k-means, hierarchical clustering
  - Dimensionality reduction: project the data from a high-dimensional space down to a small number of dimensions
    - eg: principal/independent component analysis, MDS
  - Density estimation: determine the distribution of data within the input space
    - eg: bayesian networks, mixture models.

Clustering

- Goal: grouping a collection of objects into subsets or “clusters”, such that those within each cluster are more closely related to one another than objects assigned to different clusters

Clustering



- Clustering rows

  grouping similar objects

- Clustering columns

  grouping similar variables across samples

- Bi-Clustering/Two-way clustering

  grouping objects that are similar across a subset of variables

Applications of clustering

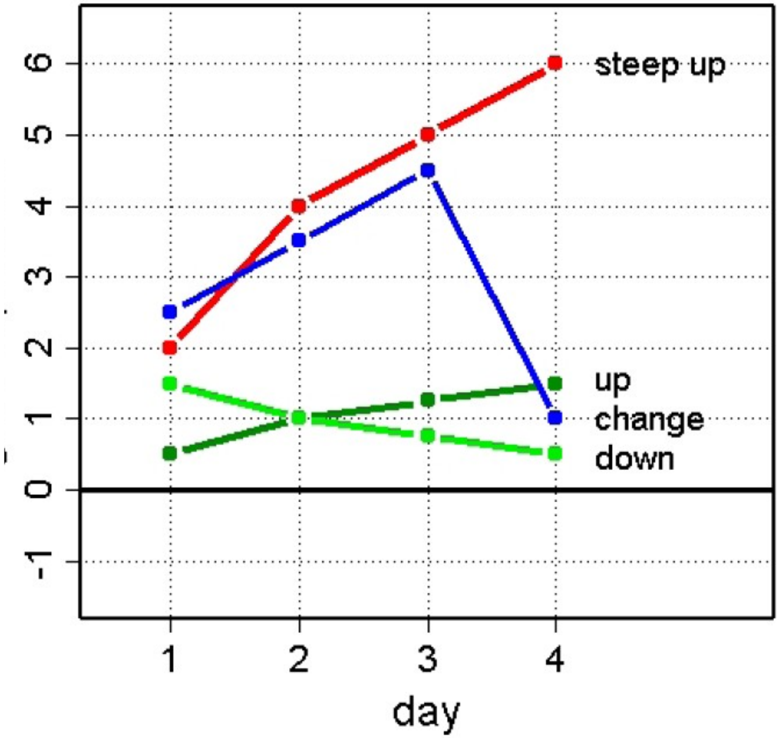
- Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
- Biology: classification of plants and animals given their features;
- Insurance: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
- City-planning: identifying groups of houses according to their house type, value and geographical location;
- Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones;
- WWW: document classification; clustering weblog data to discover groups of similar access patterns.

Clustering

- Two essential components of cluster analysis:
  - **Distance measure:** A notion of distance or similarity of two objects: When are two objects close to each other?
  - **Cluster algorithm:** A procedure to minimize distances of objects within groups and/or maximize distances between groups

Examples of distance measures

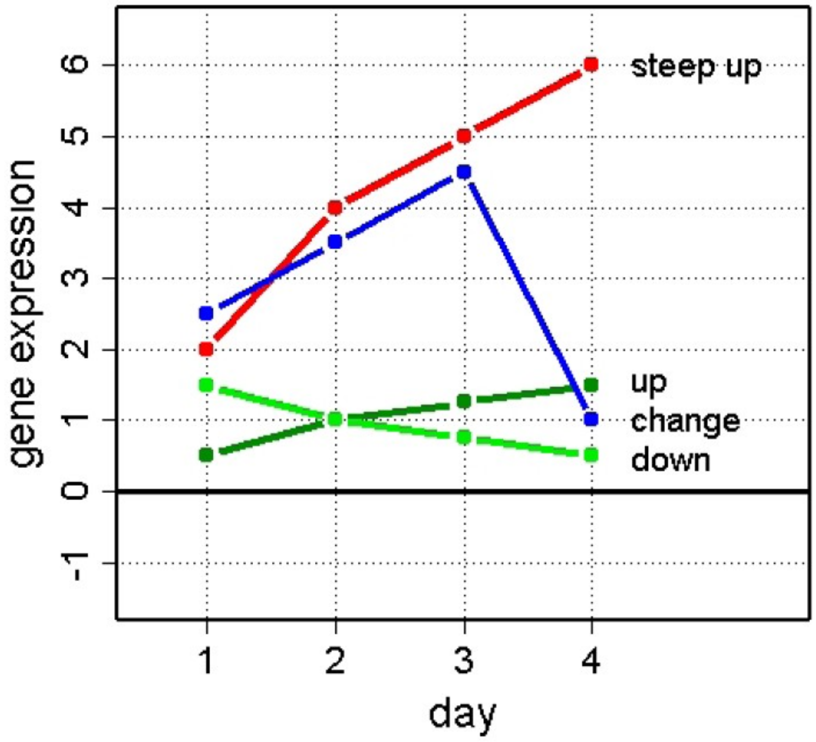
- **Euclidean** distance measures average difference across coordinates
- **Manhattan** distance measures average difference across coordinates, in a robust way
- **Correlation** distance measures difference with respect to trends



Time series example

- Measurement of gene expression on 4 (consecutive) days
- Every gene is coded by a vector of length 4

- Step up: x1=(2,4,5,6)
- up: x2=(2/4,4/4,5/4,6/4)
- down: x3=(6/4,4/4,3/4,2/4)
- change: x4=(2.5,3.5,4.5,1)



Euclidean distance

- The distance between two vectors is the square root of the sum of the squared difference over all coordinates

$$d_E(x1, x2)=\sqrt{(2-2/4)^2+(4-4/4)^2+(5-5/4)^2+(6-6/4)^2}=3\sqrt{3/4}=2.598$$

- Step up: x1=(2,4,5,6)
- up: x2=(2/4,4/4,5/4,6/4)
- down: x3=(6/4,4/4,3/4,2/4)
- change: x4=(2.5,3.5,4.5,1)

0	2.6	2.75	2.25
2.6	0	1.23	2.14
2.75	1.23	0	2.15
2.25	2.14	2.15	0

Matrix of pairwise distances

Manhattan distance

- The distance between two vectors is the sum of the absolute (unsquared) differences over all coordinates

$$d_M(x1, x2) = |2-2/4| + |4-4/4| + |5-5/4| + |6-6/4| = 5\ 1/4 = 12.75$$

- Step up: x1=(2,4,5,6)
- up: x2=(2/4,4/4,5/4,6/4)
- down: x3=(6/4,4/4,3/4,2/4)
- change: x4=(2.5,3.5,4.5,1)

0	12.75	13.25	6.5
12.75	0	2.5	8.25
13.25	2.5	0	7.75
6.5	8.25	7.75	0

Matrix of pairwise distances

Correlation distance

- Distance between two vectors is 1-ρ, where ρ is the Pearson correlation of the two vectors

$$d_c(x1,x2)=\frac{(2-\frac{17}{4})(\frac{2}{4}-\frac{17}{16})+(4-\frac{17}{4})(\frac{4}{4}-\frac{17}{16})+(5-\frac{17}{4})(\frac{5}{4}-\frac{17}{16})+(6-\frac{17}{4})(\frac{6}{4}-\frac{17}{16})}{\sqrt{(2-\frac{17}{4})^2+(4-\frac{17}{4})^2+(5-\frac{17}{4})^2+(6-\frac{17}{4})^2}\sqrt{(\frac{2}{4}-\frac{17}{16})^2+(\frac{4}{4}-\frac{17}{16})^2+(\frac{5}{4}-\frac{17}{16})^2+(\frac{6}{4}-\frac{17}{16})^2}}$$

- Step up: x1=(2,4,5,6)
- up: x2=(2/4,4/4,5/4,6/4)
- down: x3=(6/4,4/4,3/4,2/4)
- change: x4=(2.5,3.5,4.5,1)

0	0	2	1.18
0	0	2	1.18
2	2	0	0.82
1.18	1.18	0.82	0

Matrix of pairwise distances

Comparison of the distances

Euclidean

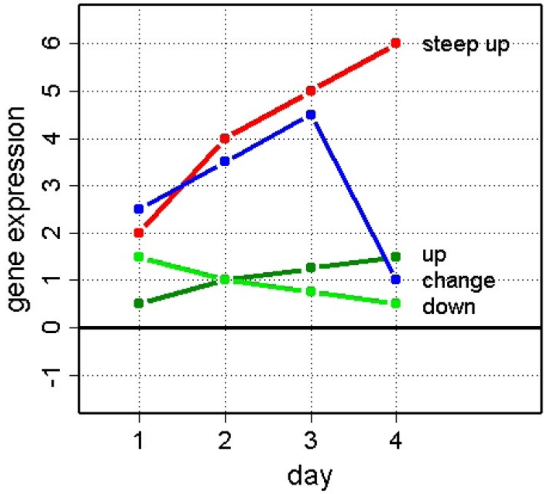
0	2.60	2.75	2.25
2.60	0	1.23	2.14
2.75	1.23	0	2.15
2.25	2.14	2.15	0

Manhattan

0	12.75	13.25	6.50
12.75	0	2.50	8.25
13.25	2.50	0	7.75
6.50	8.25	7.75	0

Correlation

0	0	2	1.18
0	0	2	1.18
2	2	0	0.82
1.18	1.18	0.82	0



	steep up	up	down	change
steep up	0 0 0	9 9 0	10 10 10	8 4 5
up	9 9 0	0 0 0	4 1 10	7 6 5
down	10 10 10	4 1 10	0 0 0	7 5 4
change	8 4 5	7 6 5	7 5 4	0 0 0

All distances are normalized to the interval [0,10] and then rounded

Clustering algorithms

- Popular algorithms for clustering
  - hierarchical clustering
  - K-means
  - SOMs (Self-Organizing Maps)
  - autoclass, mixture models...
- Hierarchical clustering allows the choice of the dissimilarity matrix.
- k-Means and SOMs take original data directly as input. Attributes are assumed to live in Euclidean space.

Hierarchical clustering

Agglomerative clustering:

1. Each object is assigned to its own cluster

2. Iteratively:

    - the two most similar clusters are joined and replaced by a new one
    - the distance matrix is updated with this new cluster replacing the two joined clusters

(divisive clustering would start from a big cluster)

Distance between two clusters

- Single linkage uses the smallest distance

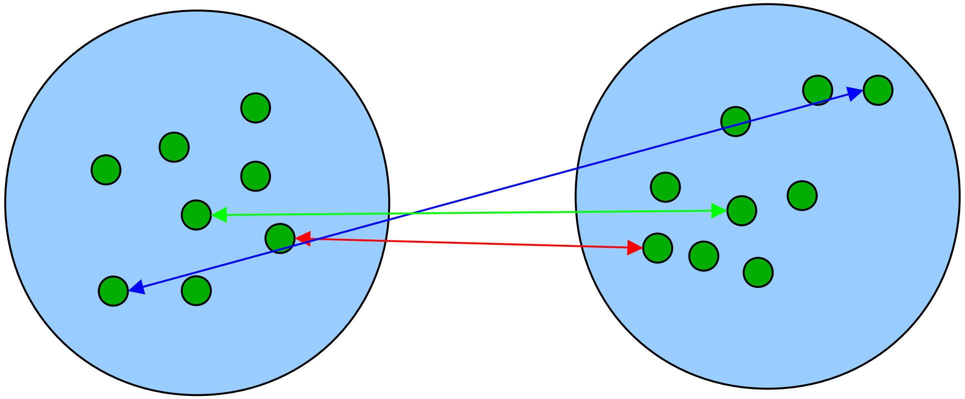
$$d_S(G,H) = \min_{i \in G,\, j \in H} d_{ij}$$

- Complete linkage uses the largest distance

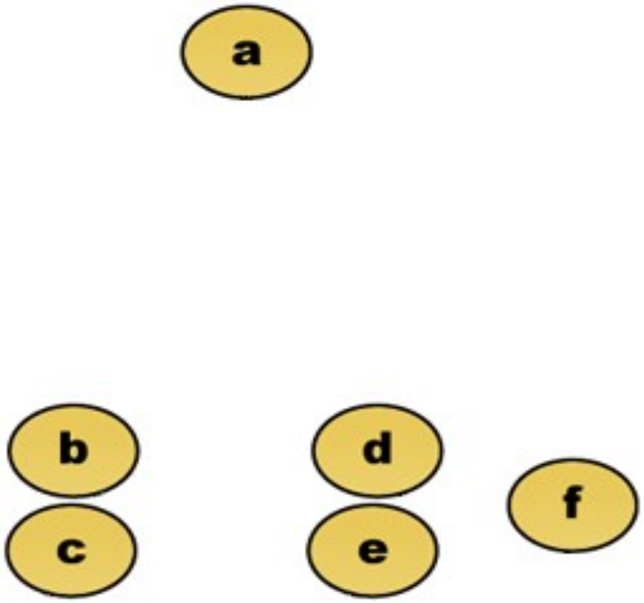
$$d_C(G,H) = \max_{i \in G,\, j \in H} d_{ij}$$

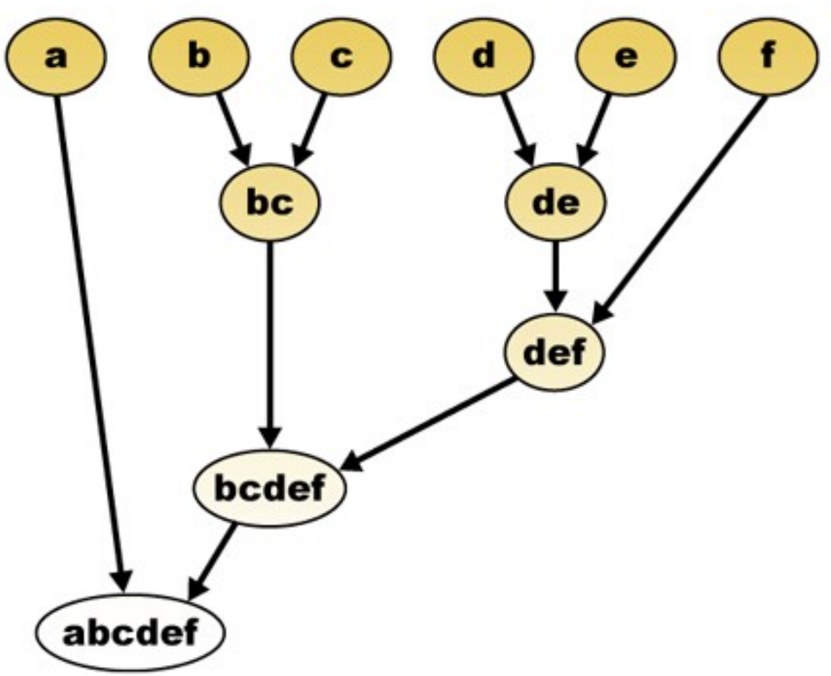
- Average linkage uses the average distance

$$d_A(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{ij}$$



Hierarchical clustering

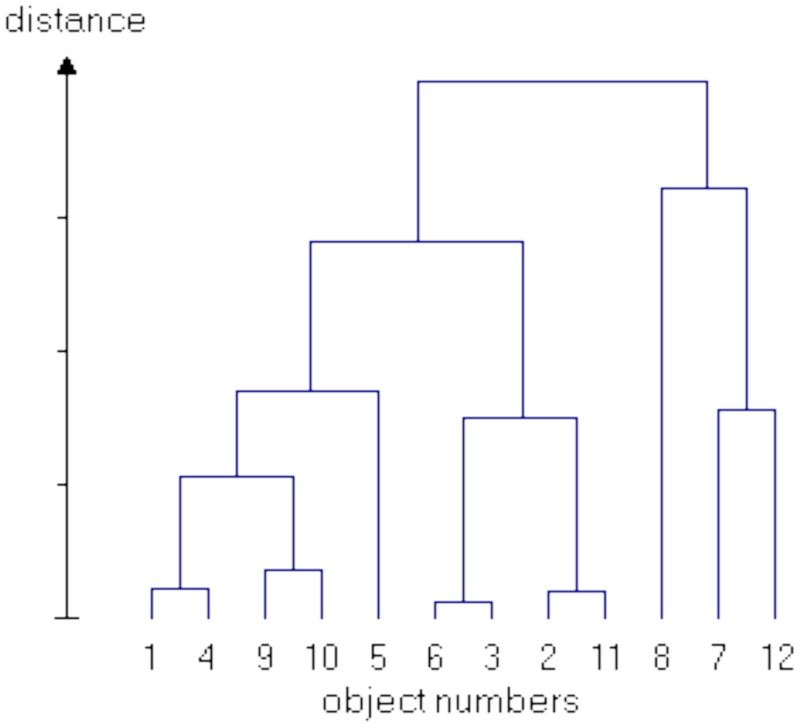




(wikipedia)

Dendrogram

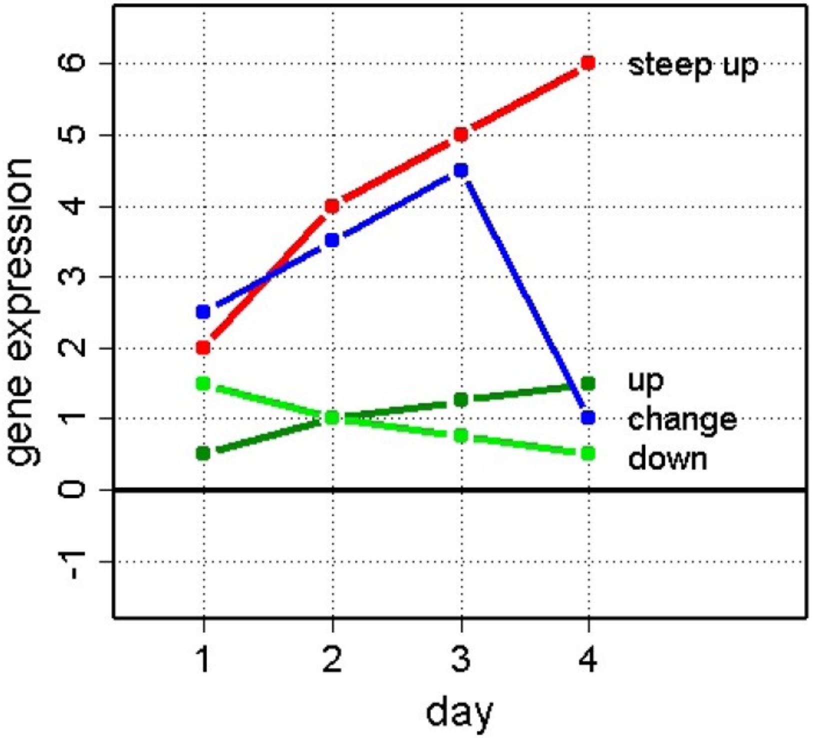
- Hierarchical clustering are visualized through dendrograms
  - Clusters that are joined are combined by a line
  - Height of line is distance between clusters
  - Can be used to determine visually the number of clusters

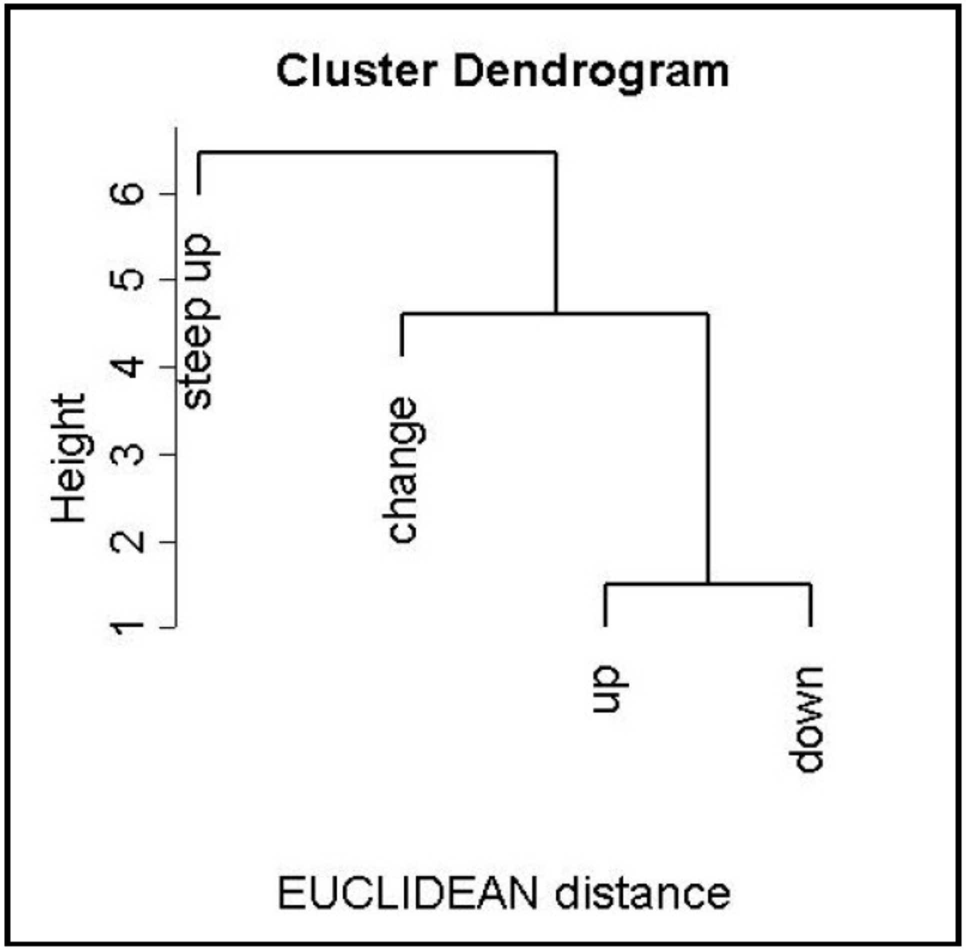


Time series example

Euclidian distance

Similar values are clustered together

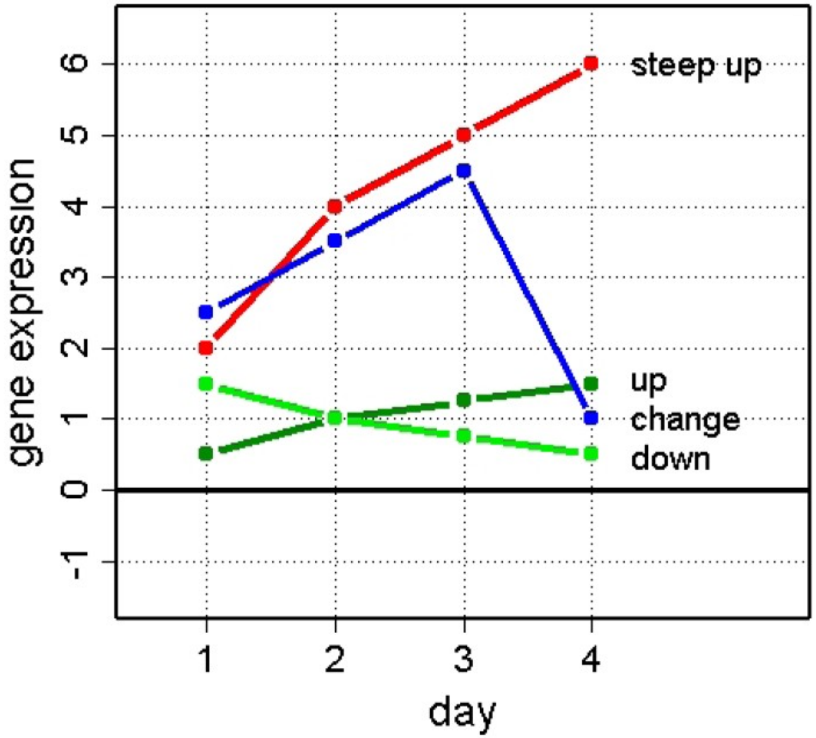


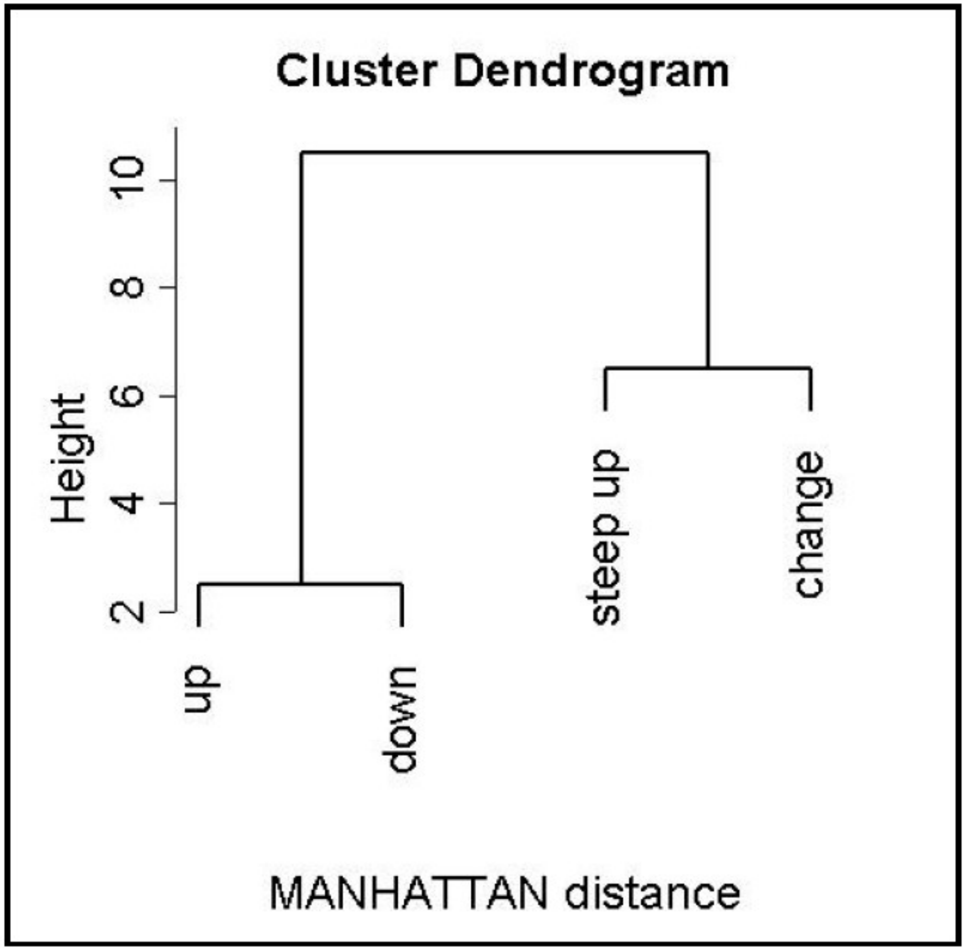


Time series example

Manhattan distance

Similar values are clustered together (robust)

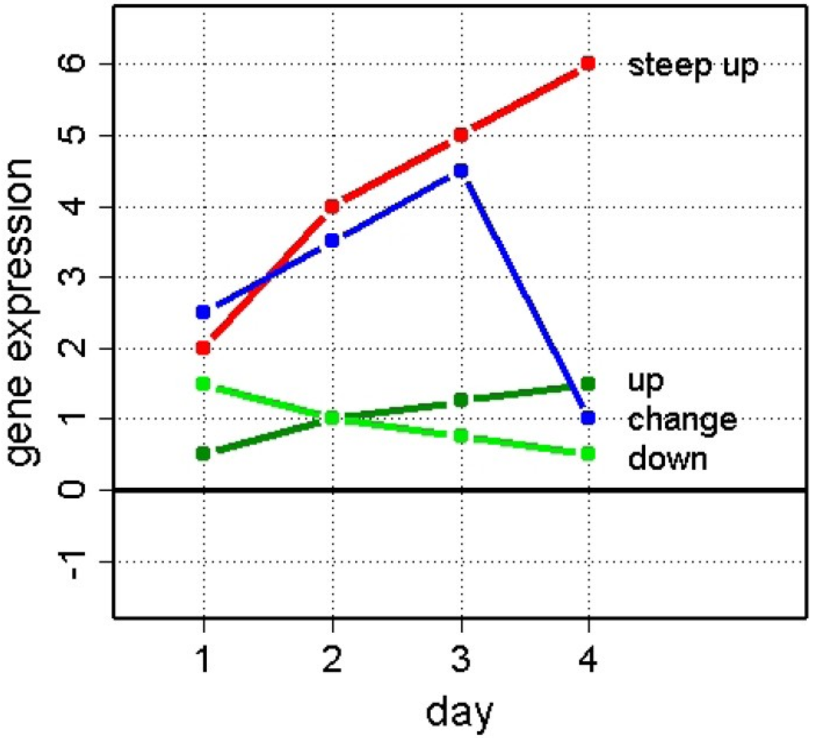


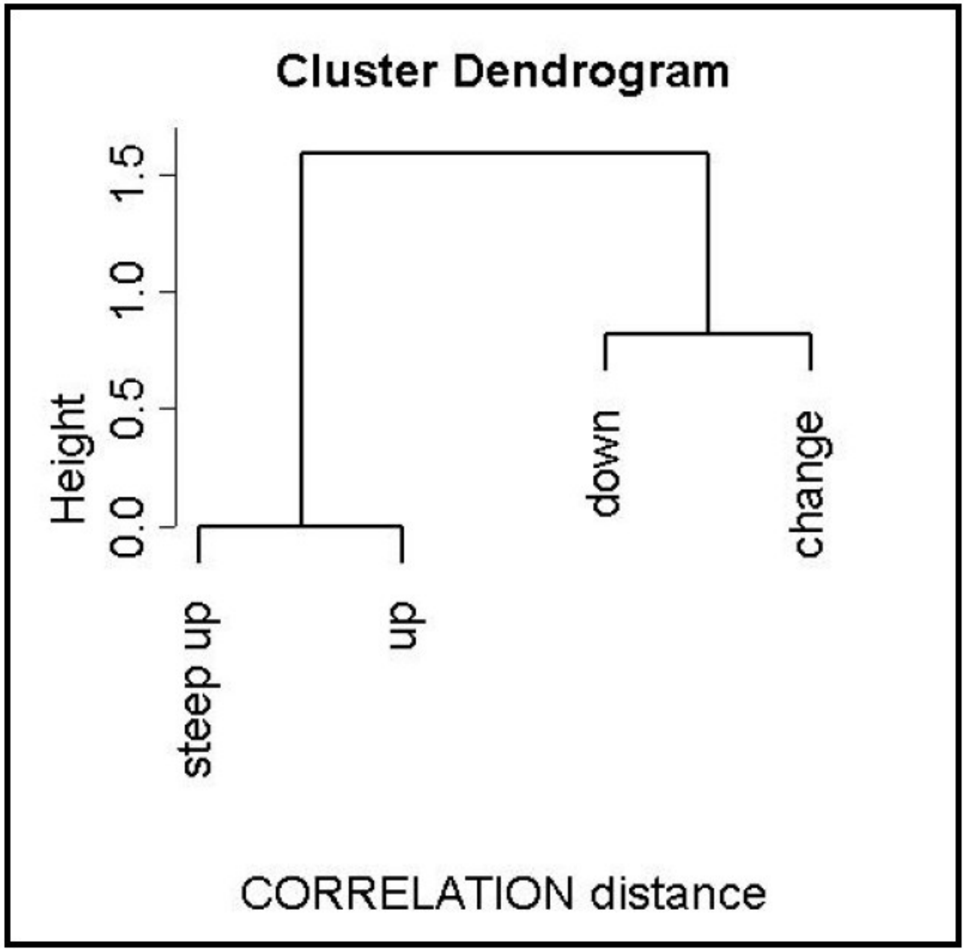


Time series example

Correlation distance

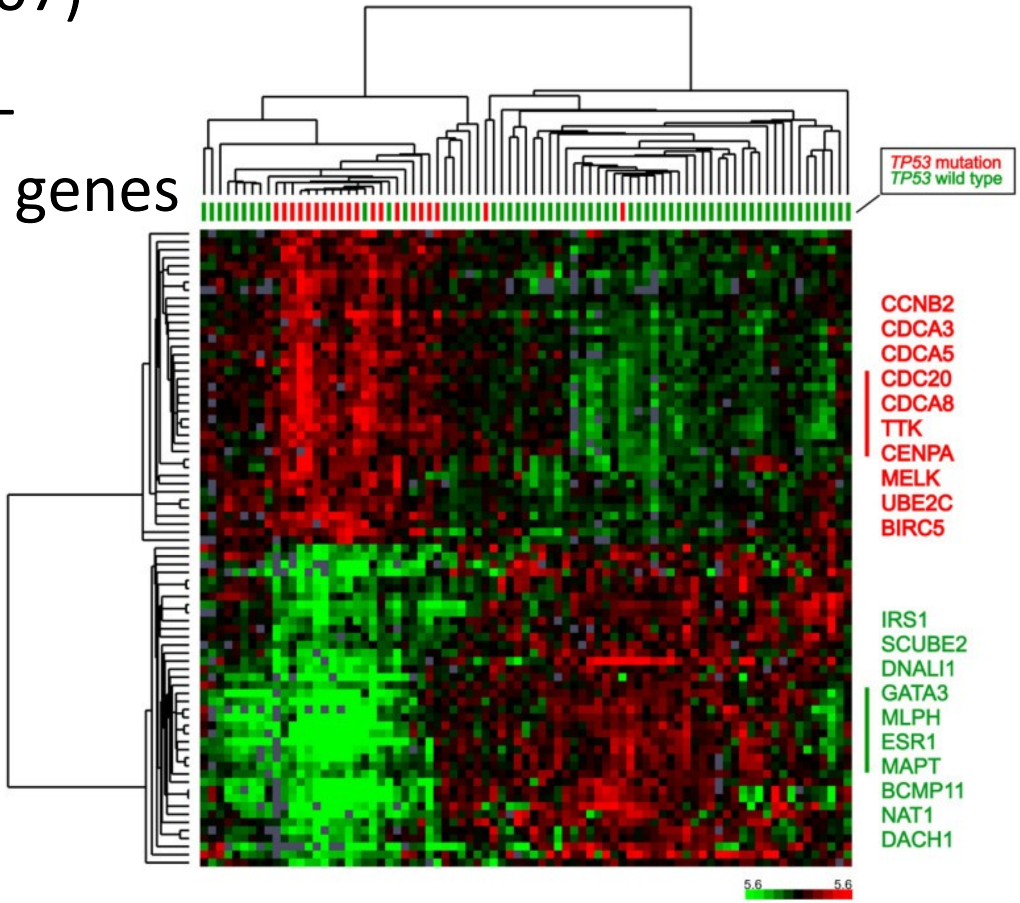
Correlated values are clustered together





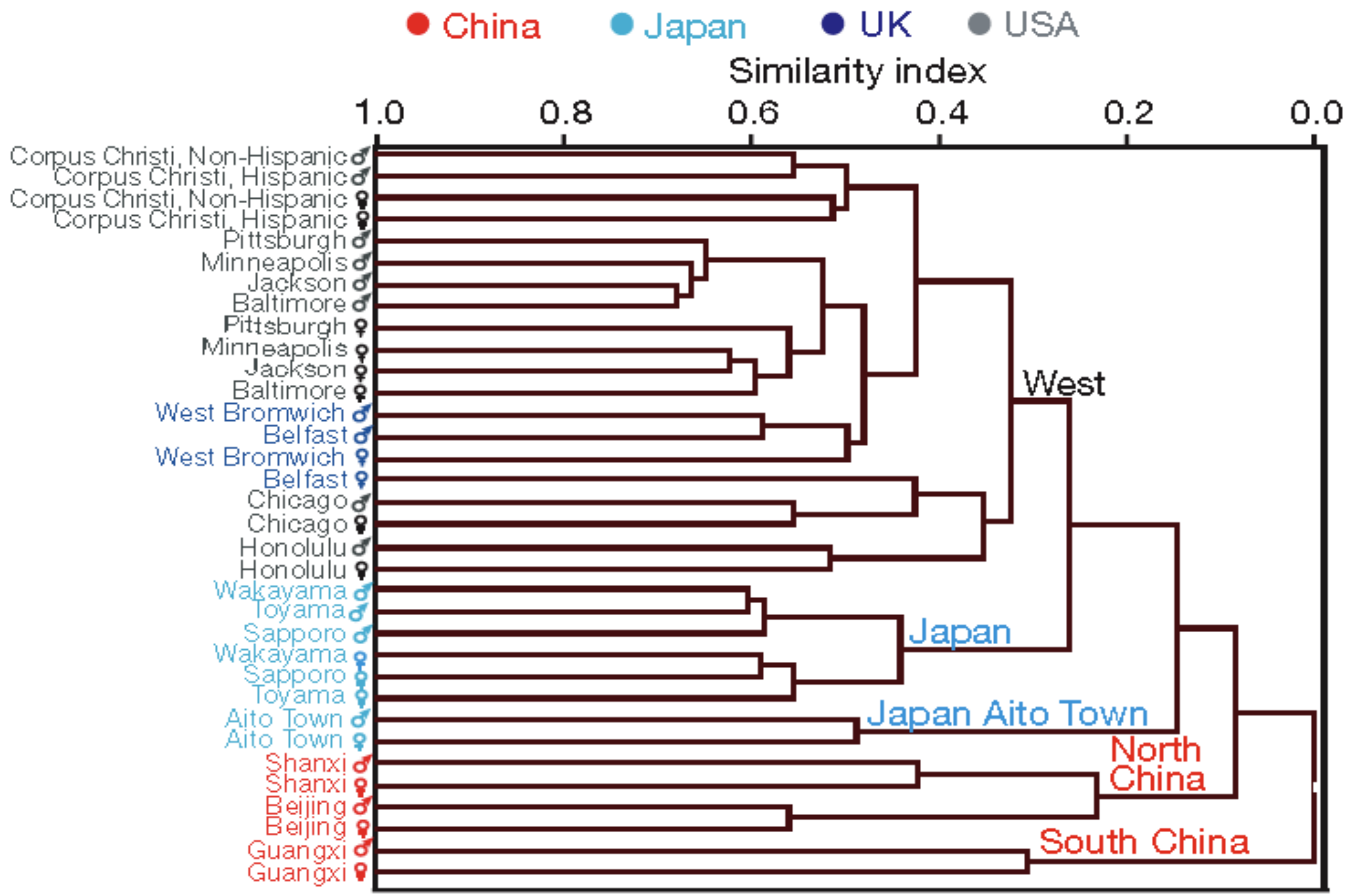
Illustrations (1)

- Breast cancer data (Langerød et al., Breast cancer, 2007)
- 80 tumor samples (wild-type,TP53 mutated), 80 genes



Illustration

Holmes et al., *Nature*, *Vol. 453, No. 15, May 2008*

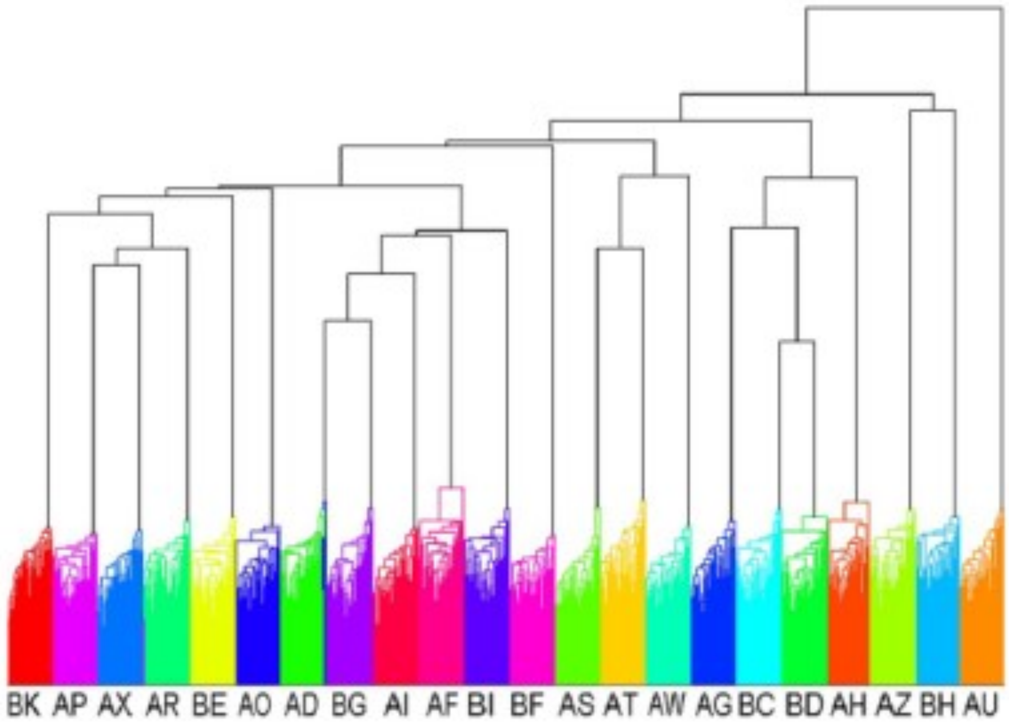


(average linkage, euclidean distance)

Illustrations (2)

Assfalg et al., *PNAS*, Jan 2008

- Evidence of different metabolic phenotypes in humans
- Urine samples of 22 volunteers over 3 months, NMR spectra analysed by HCA



Hierarchical clustering

- Strengths
  - No need to assume any particular number of clusters
  - Can use any distance matrix
  - Find sometimes a meaningful taxonomy
- Limitations
  - Find a taxonomy even if it does not exist
  - Once a decision is made to combine two clusters it cannot be undone
  - Not well theoretically motivated

Combinatorial clustering algorithm

- Given a number of clusters *K*<*N* and an encoder *C* that assigns the *i*th observation to cluster *C*(*i*)

- Clustering=finding the function C^* that *minimizes* some “loss” function that measures the degree to which the clustering goal is *not* met

- Example of loss function: within cluster scatter

$$W(C) = \frac{1}{2}\sum_{k=1}^{K} \frac{1}{N_k} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})$$

with $N_k = \sum_{i=1}^{N} I(C(i) = k)$

- Number of possible assignments is too high for enumeration

$$S(N,K) = \frac{1}{K!}\sum_{k=1}^{K}(-1)^{K-k}\binom{K}{k} k^N.$$

$S(10,4) = 34105, S(19,4) = 10^{10}.$

k-Means clustering

- Partitioning algorithm with a prefixed number k of clusters
- Use Euclidean distance between objects

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = ||x_i - x_{i'}||^2$$

- Try to minimize the sum of intra-cluster variances

$$\begin{aligned} W(C) &= \frac{1}{2} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{C(i)=k} \sum_{C(i')=k} ||x_i - x_{i'}||^2 \\ &= \sum_{k=1}^{K} \sum_{C(i)=k} ||x_i - \bar{x}_k||^2 \end{aligned}$$

where $\bar{x}_k = (\bar{x}_{1k}, \ldots, \bar{x}_{pk})$ is the center of cluster k and N_k is the number of points in cluster k:

$$N_k = \sum_{i=1}^{N} I(C(i) = k)$$

K-Means clustering

- Equivalent to solve:

$$\min_{C,\{m_k\}_1^K} \sum_{k=1}^{K} \sum_{C(i)=k} ||x_i - m_k||^2.$$

- Randomly assign each point to a cluster

- Iterate through:

- Given the current cluster assignment, compute the cluster means $\{m_1, \ldots, m_K\}$

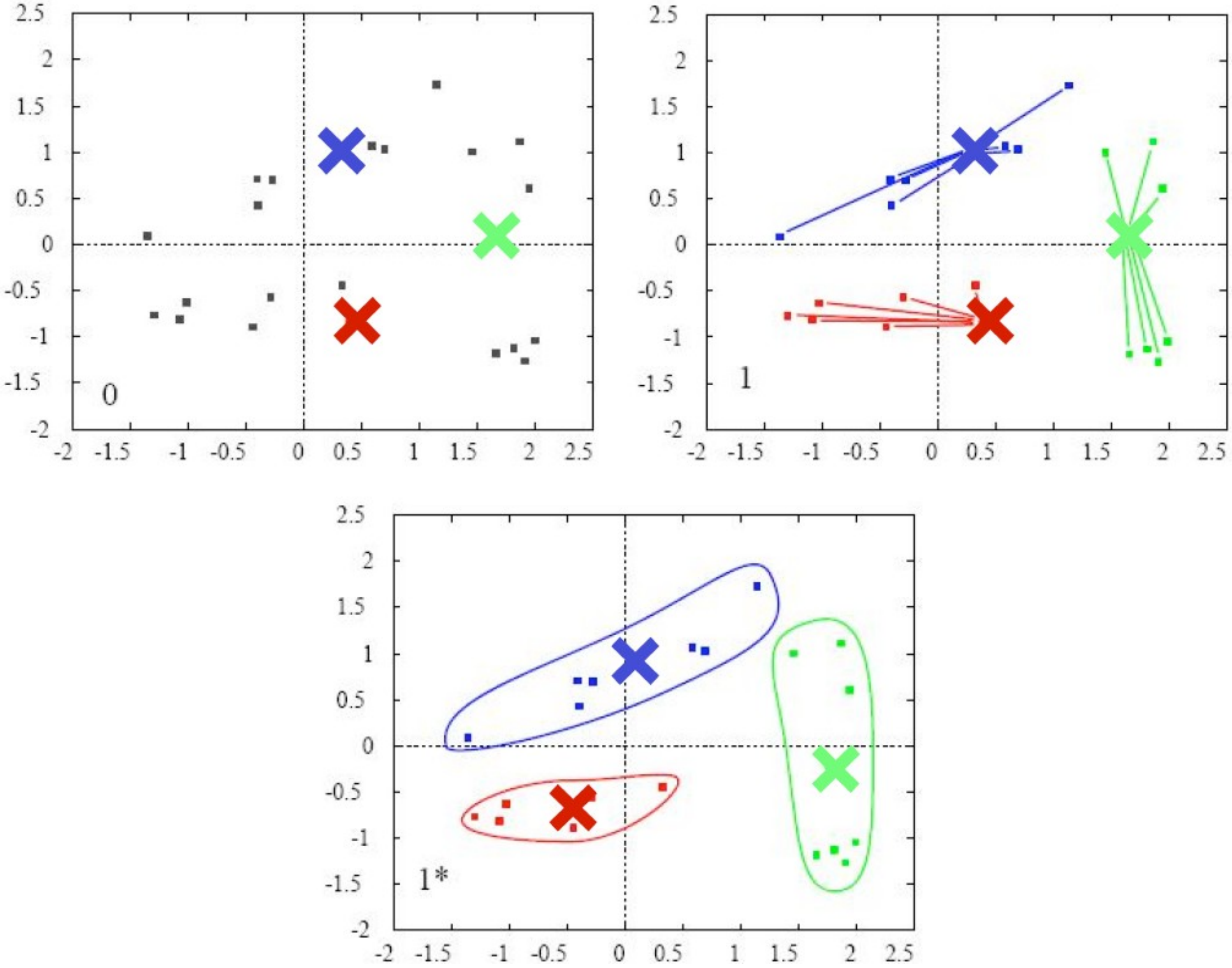
- Given the current cluster means, assign each observation to the closest cluster mean

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} ||x_i - m_k||^2.$$

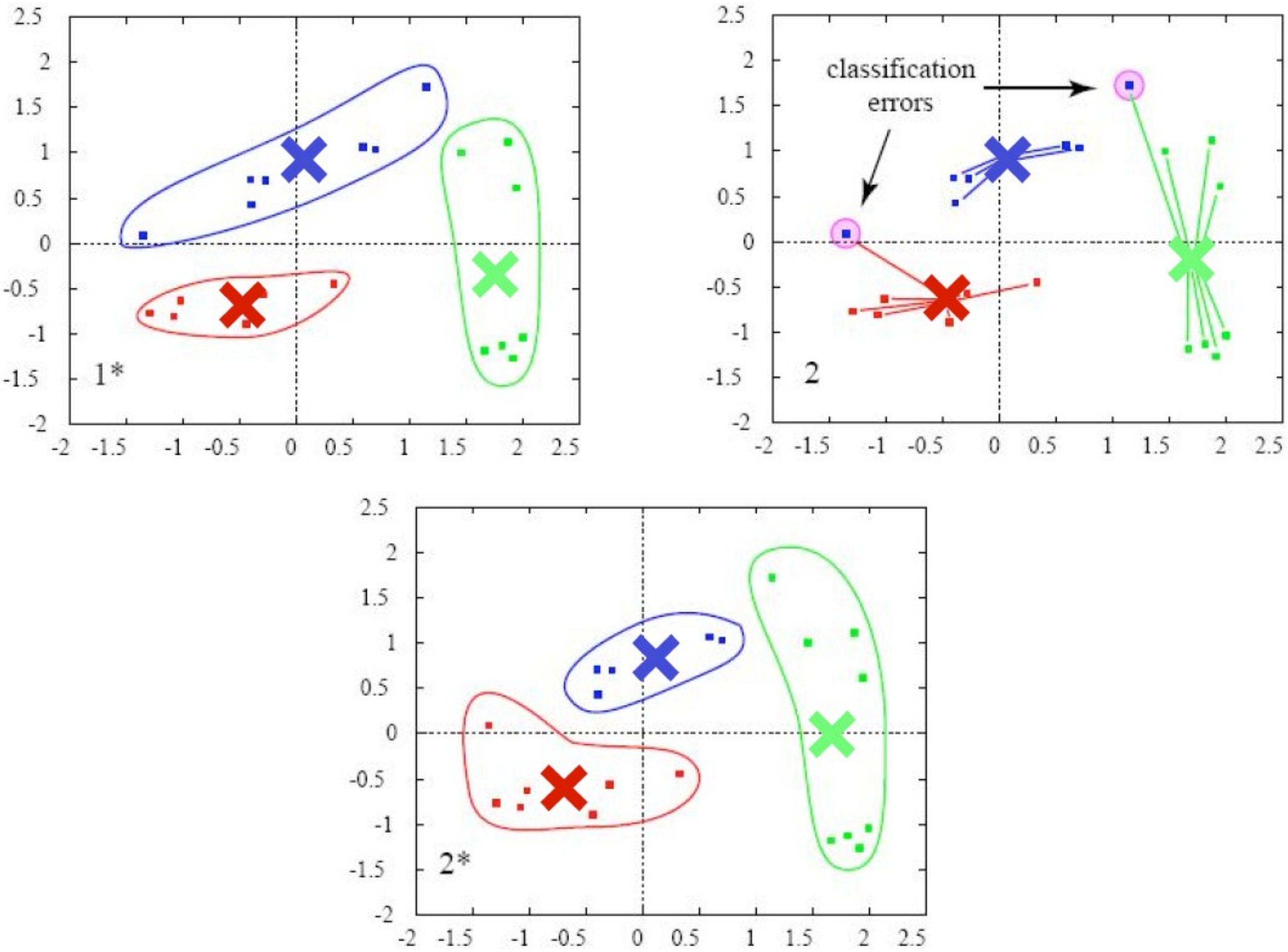
- Stop when the assignments do not change

k-Means clustering

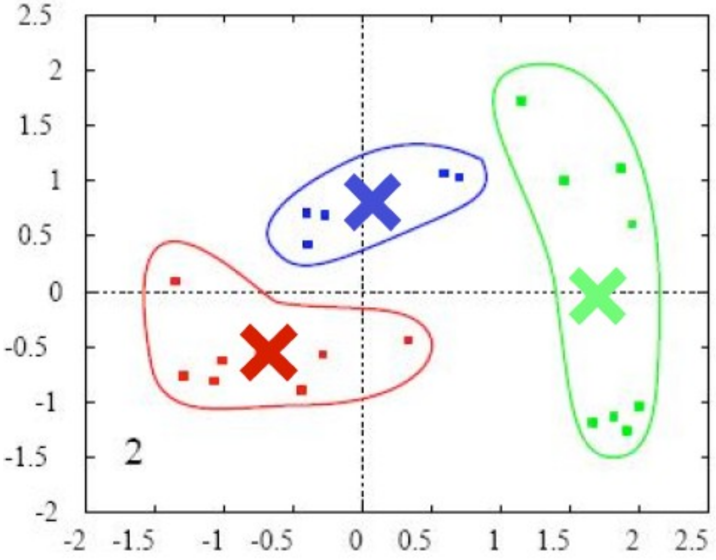
start

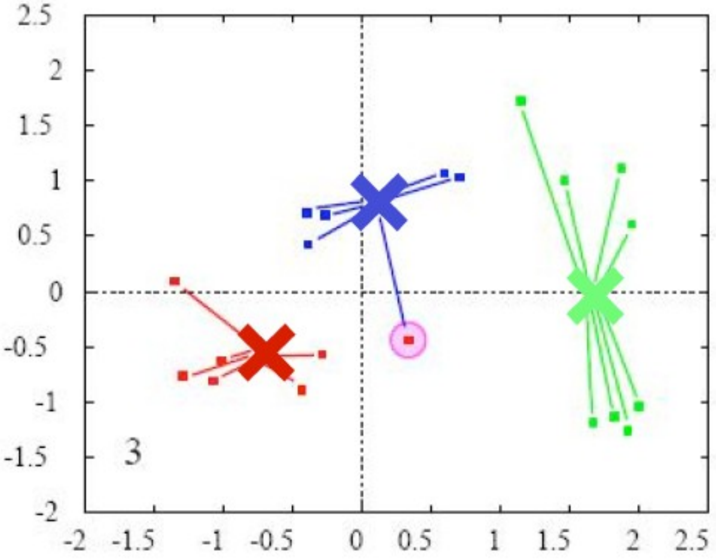


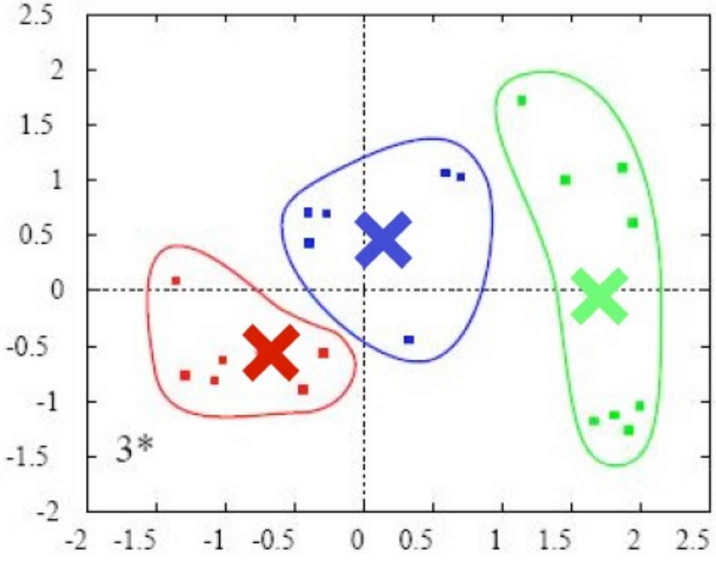
k-Means clustering



k-Means clustering



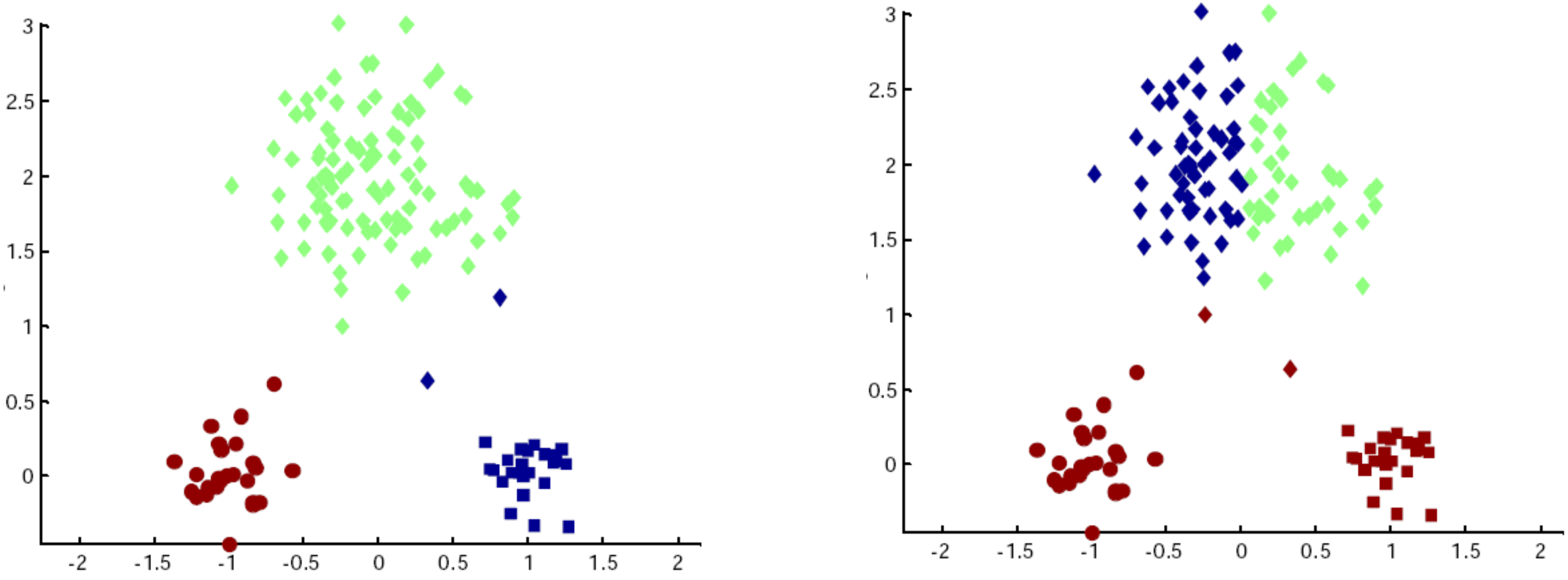




end

K-Means convergence

- Each step reduces within cluster scatter => convergence is ensured but towards a local optimum only
- You could obtain any of these from a random start of k-means



- Solution: restart the algorithm several times

Application: vector quantization

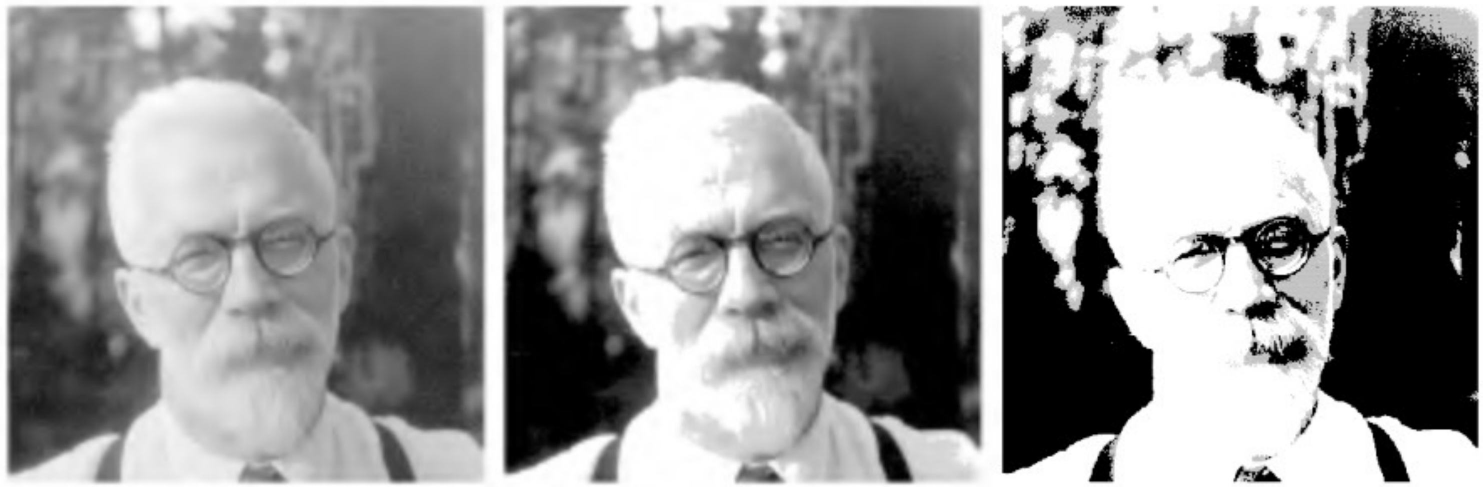
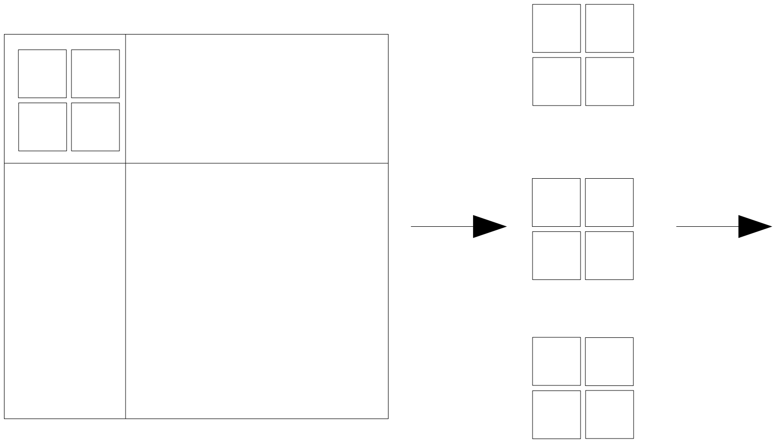


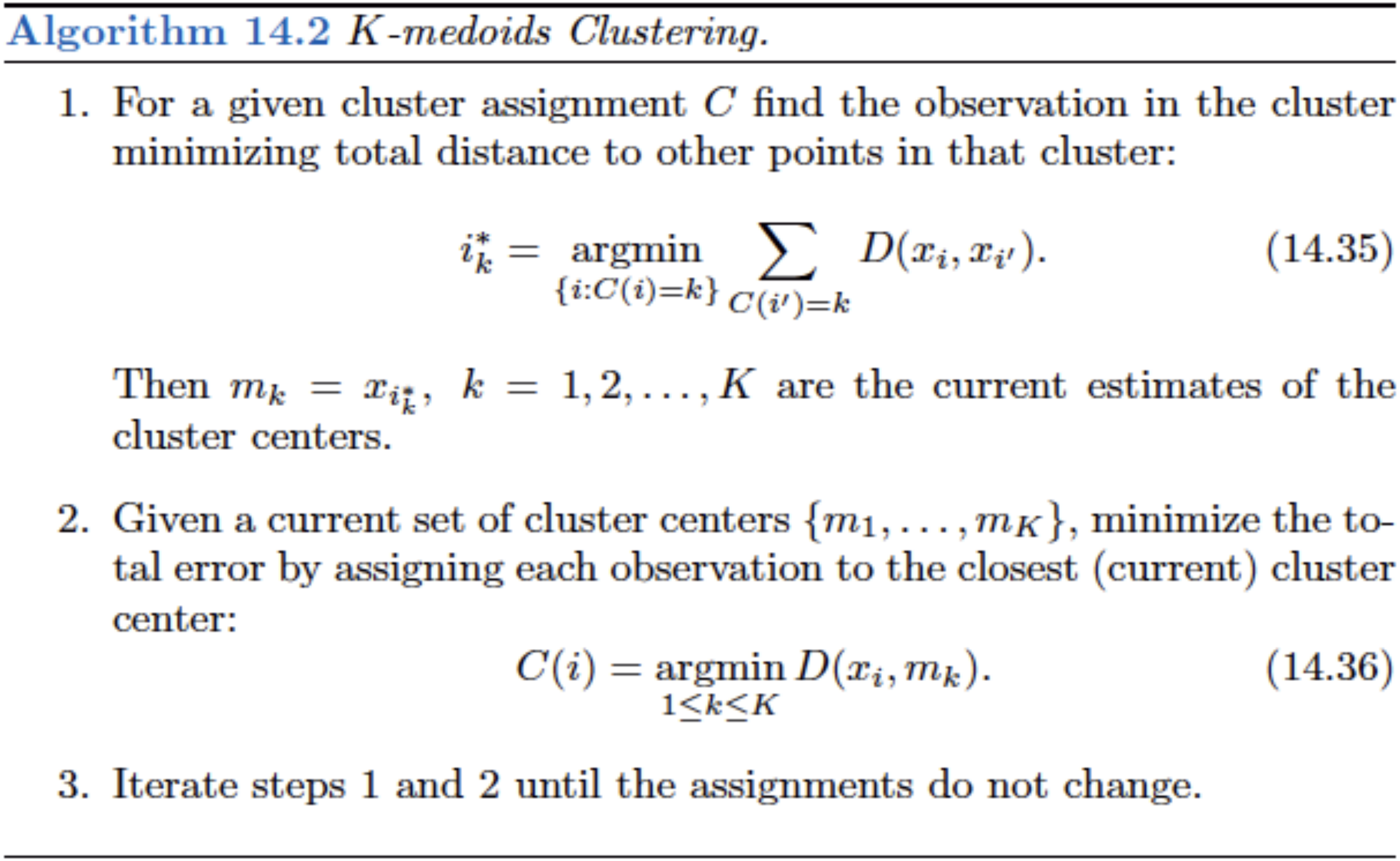
FIGURE 14.9. *Sir Ronald A. Fisher (1890-1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a* 1024×1024 *grayscale image at* 8 *bits per pixel. The center image is the result of* 2 × 2 *block VQ, using* 200 *code vectors, with a compression rate of* 1.9 *bits/pixel. The right image uses only four code vectors, with a compression rate of* 0.50 *bits/pixel*



k-Means

K-medoids

Extension of k-Means to handle any distance measure



Algorithm 14.2 *K-medoids Clustering.*

1. For a given cluster assignment C find the observation in the cluster minimizing total distance to other points in that cluster:

$$i_k^* = \underset{\{i:C(i)=k\}}{\operatorname{argmin}} \sum_{C(i')=k} D(x_i, x_{i'}). \tag{14.35}$$

Then $m_k = x_{i_k^*}, \; k = 1, 2, \ldots, K$ are the current estimates of the cluster centers.

2. Given a current set of cluster centers $\{m_1, \ldots, m_K\}$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} D(x_i, m_k). \tag{14.36}$$

3. Iterate steps 1 and 2 until the assignments do not change.

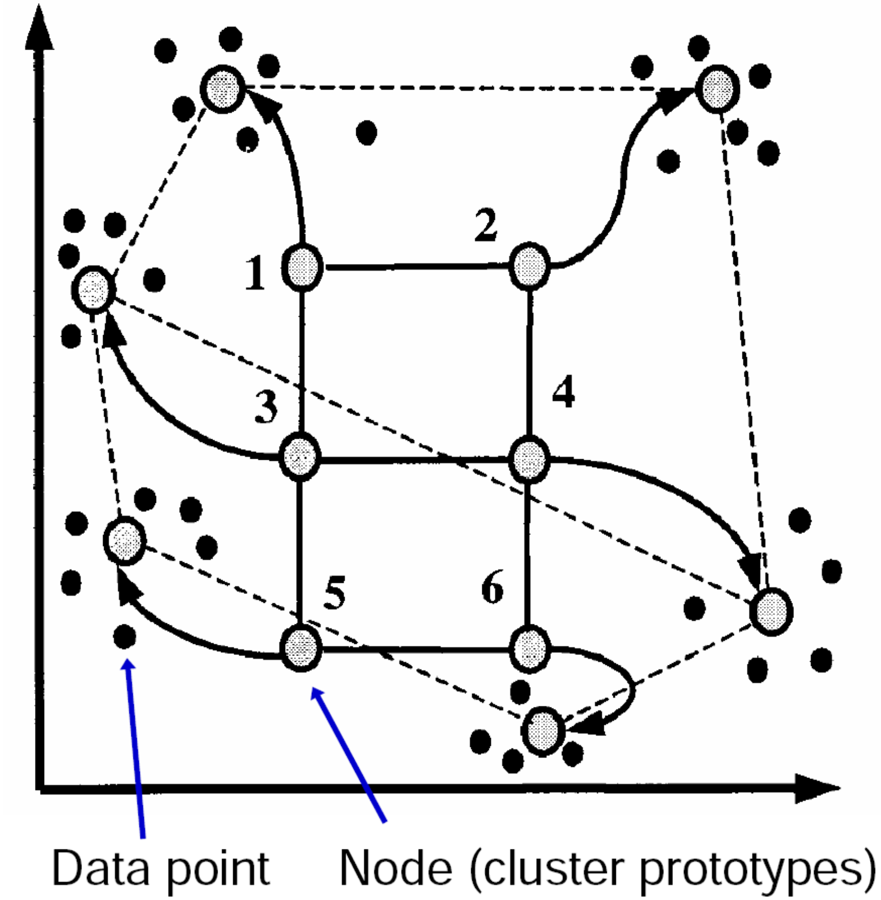
Much slower than K-means

k-Means clustering

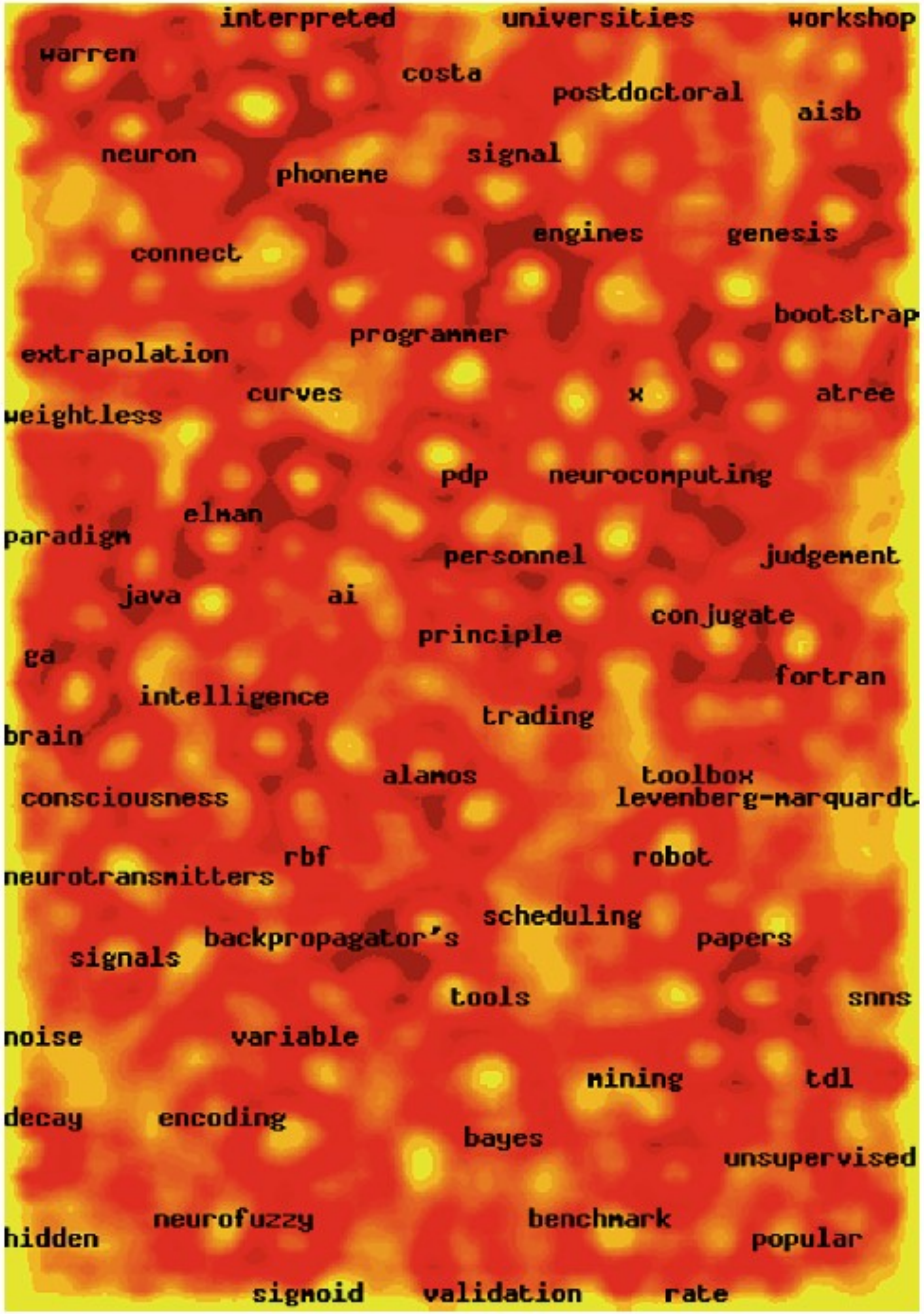
- Strengths
  - Simple, understandable
  - Can cluster any new point (unlike hierarchical clustering)
  - Well motivated theoretically
- Limitations
  - Must fix the number of clusters beforehand
  - Sensitive to the initial choice of cluster centers
  - Sensitive to outliers

Self-organizing maps

- SOM's are similar to k-means but with additional constraints
- Mapping from data space onto one or two-dimensional array of k total nodes
- Iterations steps:
  - Pick data point P at random.
  - Move all nodes in direction of P: the closer (further) a node is in network topology, the most (less).
  - Decrease amount of movement with iteration steps.



Document organization



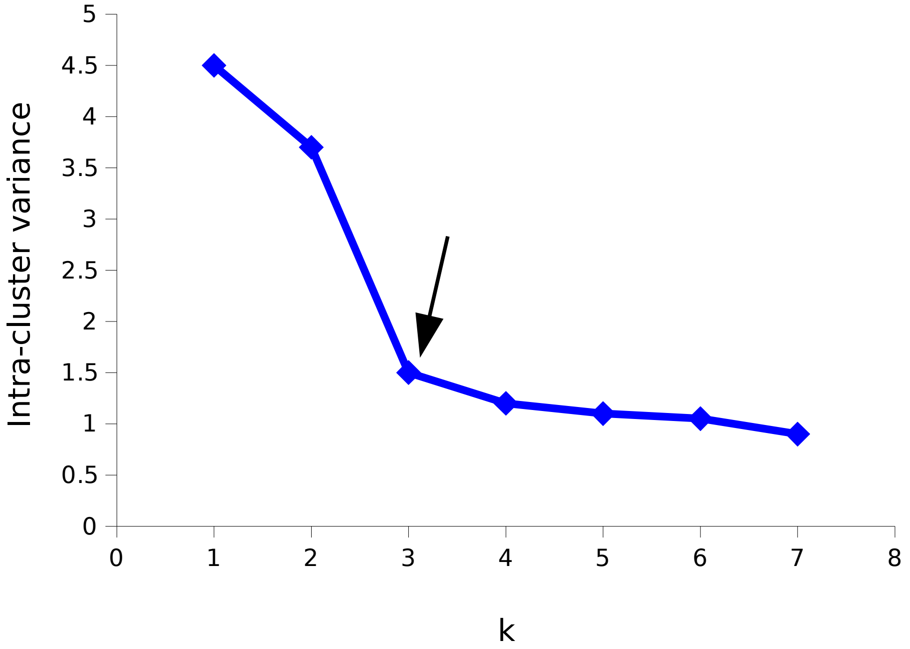
http://websom.hut.fi

How many clusters?

- Where to stop hierarchical clustering? How to choose k for k-means and SOMs?
- Very difficult and open question.
- Similar to overfitting in SL...
  - Too many clusters: overfit the data. You find non existing clusters in the data (noise)
  - Too few clusters: underfit the data. We miss some truly existing clusters.
- ...but without the possibility to cross-validate

How many clusters?

- Locating the “knee” in the intra-cluster variance curve



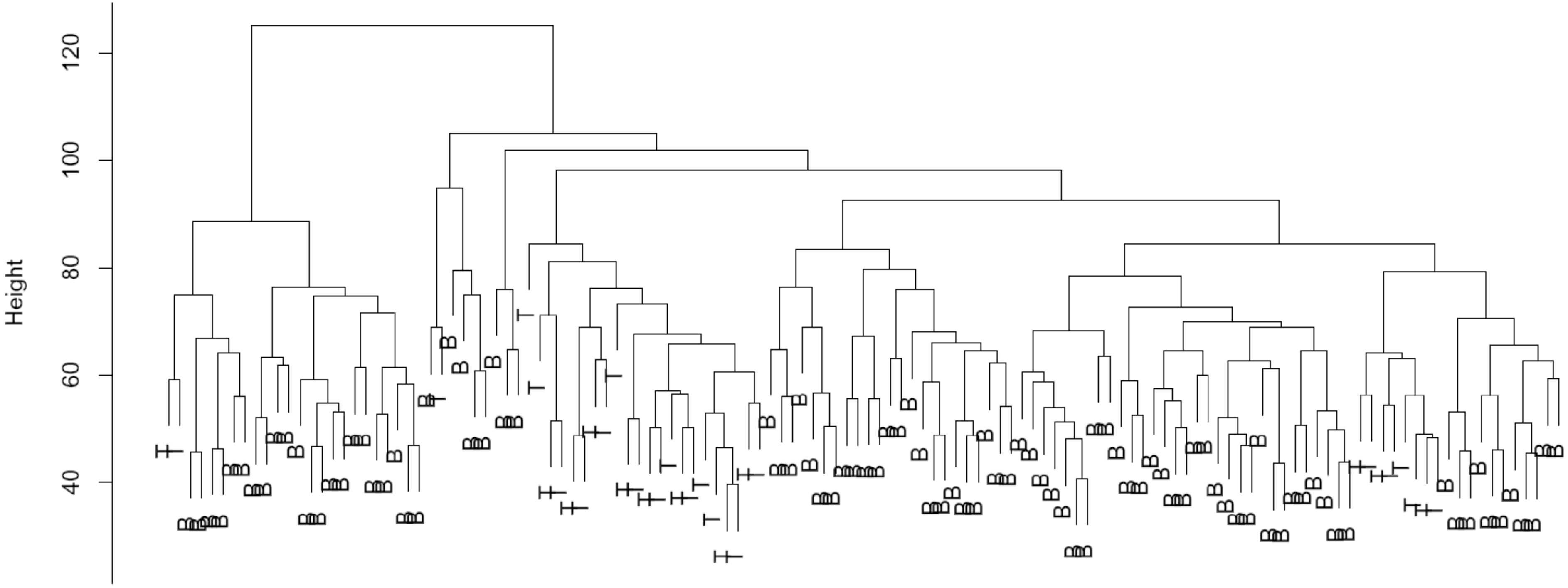
How many clusters?

- Other criteria:
  - Internal indices:
    - Statistics based on within- and between-clusters distances
    - Select k that minimizes/maximizes such internal index
  - Gap statistic:
    - Resampling method that compares some internal index with what would be obtained from random data
    - Search for the value of k that maximizes the difference (Tibshirani et al., 2001)
  - Stability: select k that leads to the more stable clusters (computed by a bootstrap analysis) (Ben-Hur et al., PSB 2002)

Feature selection for clustering

- Feature selection can also improve clustering by decreasing noise (and computing times)
- Example on Leukemia patients (Chiaretti et al., 2004)

Without gene selection

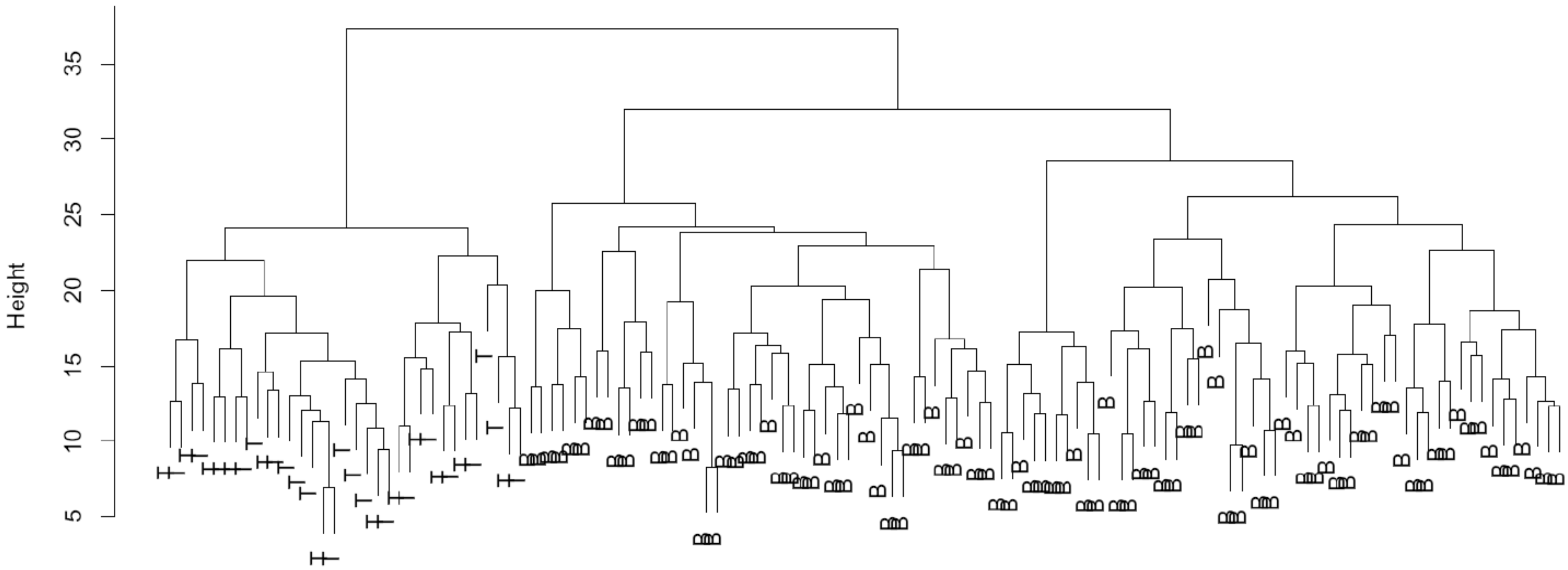


(from J.Rahnenführer, 2007)

Feature selection for clustering

- Feature selection can also improve clustering by decreasing noise (and computing times)
- Example on Leukemia patients (Chiaretti et al., 2004)

With 100 top variance genes

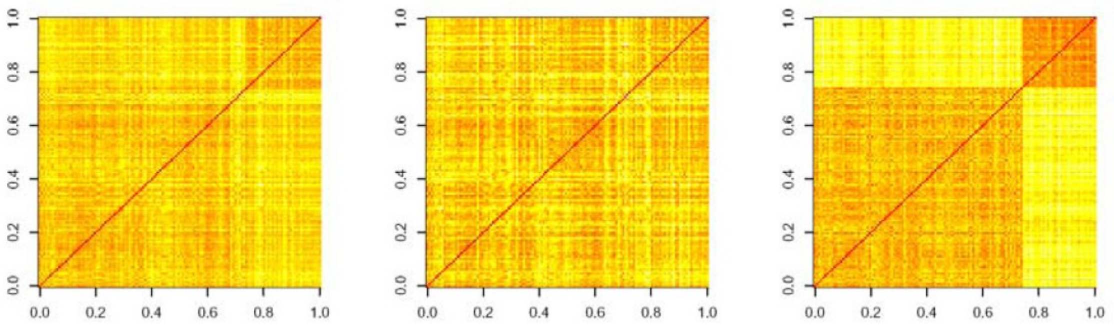


(from J.Rahnenführer, 2007)

Feature selection for clustering

- Feature selection can also improve clustering by decreasing noise (and computing times)
- Example on Leukemia patients (Chiaretti et al., 2004)

Distance matrices for clustering Leukemia patients (Chiaretti et al., 2004)

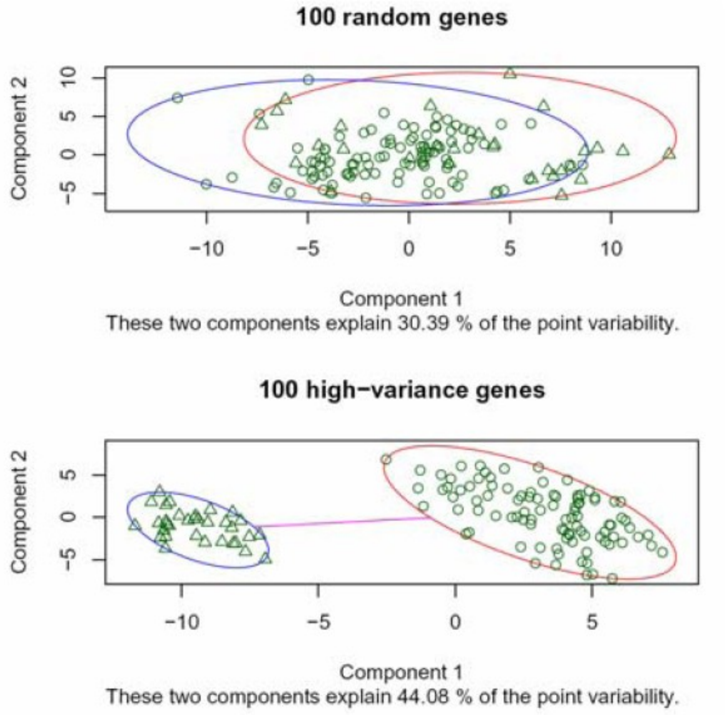


All genes

100 random genes

100 high-variance genes

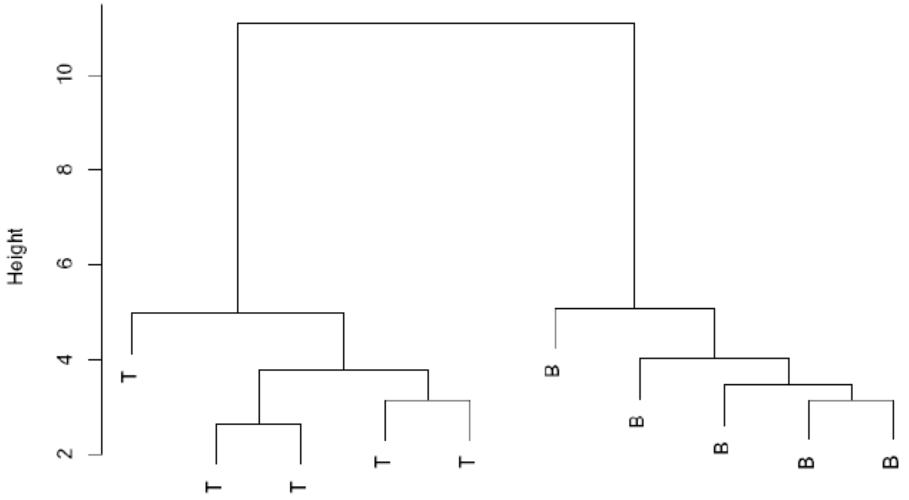
Plot of sample types in first two principal components

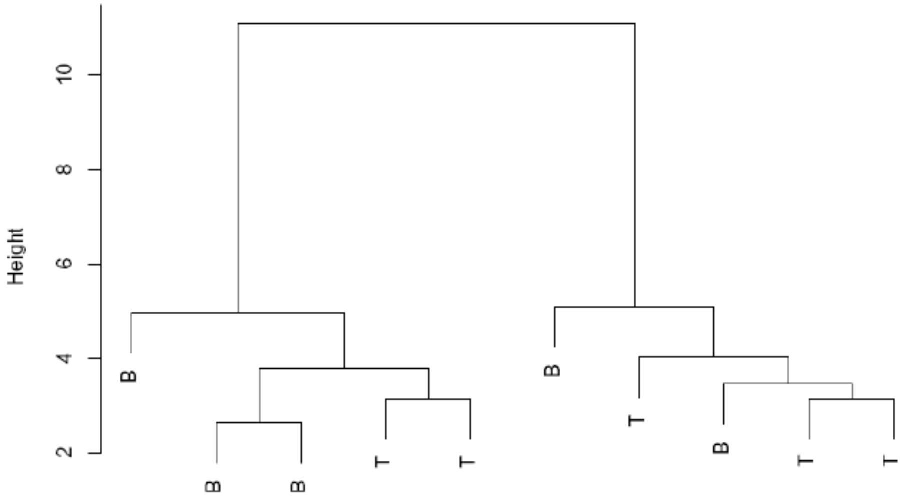


(from J.Rahnenführer, 2007)

Selection bias in clustering

- Clustering after supervised feature selection should be avoided
- You will always retrieve the classification since this is the criterion you used to select the variables





Left dendrogram obtained by

1. Random assignment of sample labels
2. Selection of best discriminating genes
3. Clustering with selected genes

Right plot shows original labels

Dimensionality reduction

- Main goal: reduce the dimensionality of the data set to a smaller space (2D, 3D)

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0.86	-0.48	-0.18	0.37	0.98	-0.97	0.84	-0.06	-0.35	0.56
-2.3	-1.2	-4.5	-0.13	0.02	0.09	-0.71	0.88	0.78	0.7
0.26	-0.41	0.02	0.33	0.39	-0.46	0.92	0.15	-0.06	-0.26
0.21	-0.13	-0.33	-0.5	0.82	-0.19	0.08	0.48	0.64	-0.38
0.25	0.11	-0.94	-0.04	0.45	-0.15	-0.85	0.45	0.42	0.29
0.34	0.25	0.83	-0.24	-0.46	0.94	0.12	-0.02	-0.49	0.71

→

X'_1	X'_2
0.11	-0.21
-2.3	-1.2
0.76	-0.46
-0.03	-0.45
0.23	-0.02
-0.65	-0.91

- Feature selection: find a subset of the original variables ($X'_i=X_j$ for some j)

- Feature extraction: transform the original space into a space of fewer dimensions ($X'_i=f(X_1,\ldots,X_1)$)

- Linear methods: $f(X_1,\ldots,X_p)=w_0+w_1X_1+\ldots+w_pX_1$

Objectives of dimensionality reduction

- Reduce dimensionality (pre-processing for other methods)
- Choose the most useful (informative) variables
- Compress the data
- Visualize multidimensional data
    - to identify groups of objects
    - to identify outliers

Principal Component Analysis

- A linear feature extraction technique

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0.86	-0.48	-0.18	0.37	0.98	-0.97	0.84	-0.06	-0.35	0.56
-2.3	-1.2	-4.5	-0.13	0.02	0.09	-0.71	0.88	0.78	0.7
0.26	-0.41	0.02	0.33	0.39	-0.46	0.92	0.15	-0.06	-0.26
0.21	-0.13	-0.33	-0.5	0.82	-0.19	0.08	0.48	0.64	-0.38
0.25	0.11	-0.94	-0.04	0.45	-0.15	-0.85	0.45	0.42	0.29
0.34	0.25	0.83	-0.24	-0.46	0.94	0.12	-0.02	-0.49	0.71

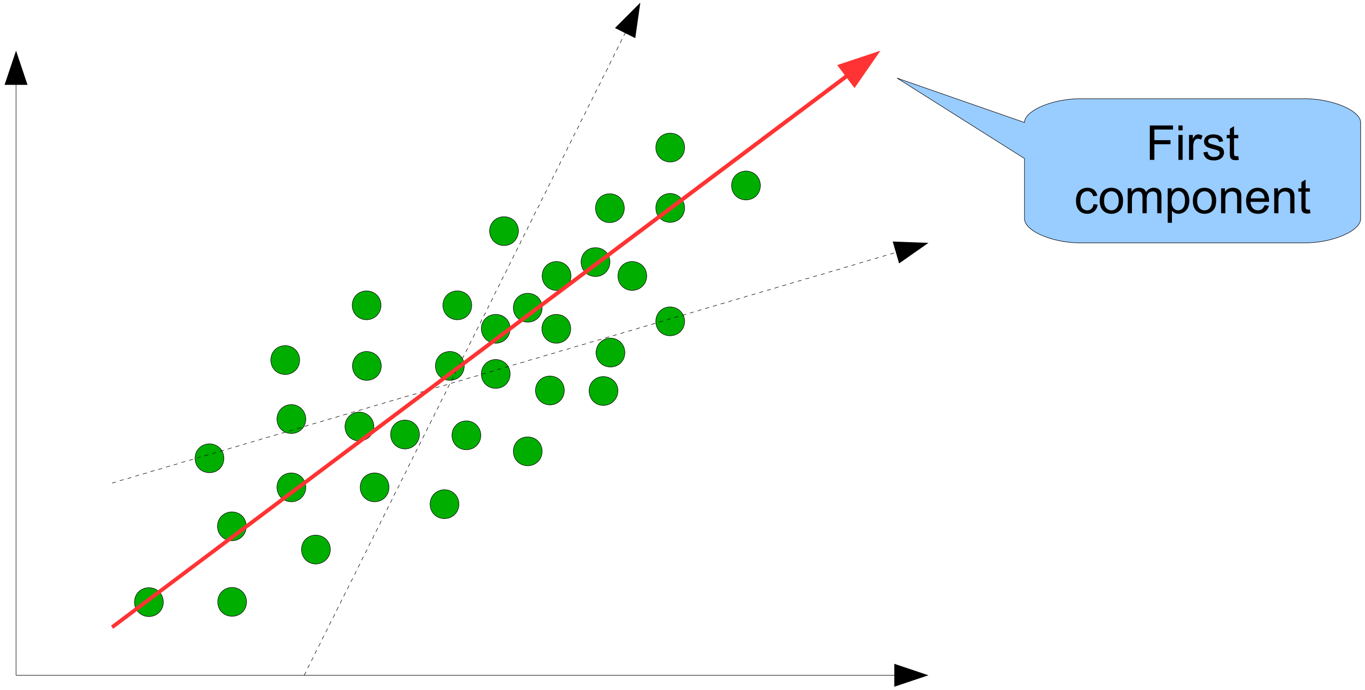
→

PC_1	PC_2
0.67	0.48
-2.3	-1.2
-0.03	0.67
-0.75	-0.79
-0.1	-0.04
0.47	0.84

- Transform some large number of variables into a smaller number of uncorrelated variables called principal components (PCs)

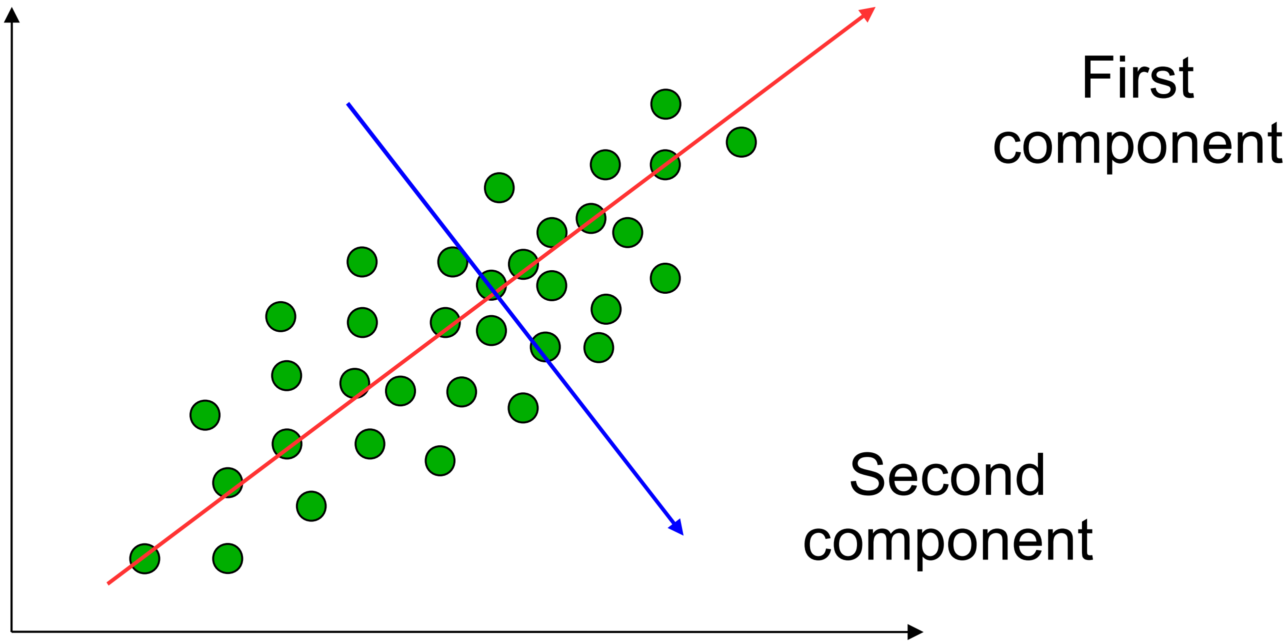
Basic idea

- Goal: map data points into a few dimensions while trying to preserve the variance of the data as much as possible



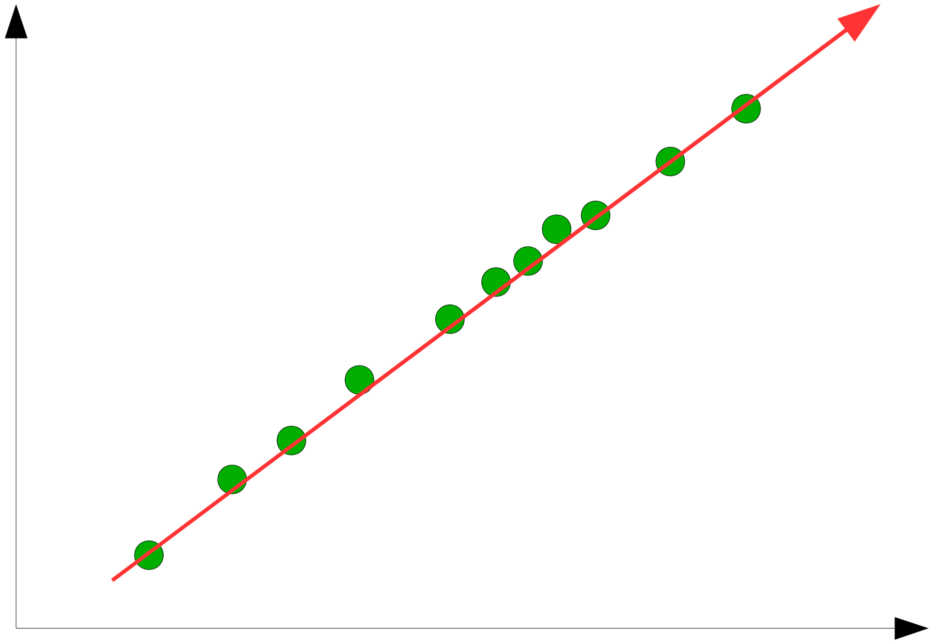
Basic idea

- Goal: map data points into a few dimensions while trying to preserve the variance of the data as much as possible



Principal Component Analysis

- Particularly efficient when there are a lot of correlation between variables (correlation=redundancy)



Mathematically

- Two formulations
  - Maximum variance: find the directions that maximize the variance of the projected data
  - Minimum-error formulation: minimizes the reconstruction error of the projected data

Mathematically

- Consider a set of observations $\{x_n\}, n = 1, \ldots, N$ with x_n a vector of dimension D.
- We want to find the unit direction u_1 that maximizes the variance of the projection:

$$\arg\max_{u_1} \frac{1}{N} \sum_{n=1}^{N} ||u_1^T x_n - u_1^T \bar{x}||^2 = u_1^T C u_1$$

with $$||u_1|| = u_1^T u_1 = 1$$

$$C = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T$$

Mathematically

- Introducing lagrange multiplier:

$$u_1^T C u_1 + \lambda_1 (1 - u_1^T u_1)$$

- Setting the derivative with respect to u_1 equal to zero:

$$C u_1 = \lambda_1 u_1$$

$\Rightarrow$ u_1 must be an eigenvector of C .

- The variance is given by:

$$u_1^T C u_1 = \lambda_1$$

$\Rightarrow$ u_1 is the eigenvector corresponding to the highest eigenvalue λ_1

Mathematically

- The (M+1)th component is obtained by maximizing:

$$u_{M+1}^T C u_{M+1}$$

With the constraints $u_{M+1}^T u_{M+1} = 1$

$$u_{M+1}^T u_i = 0 \quad \forall i = 1, \ldots, M+1$$

- Using lagragian multiplier:

$$u_{M+1}^T C u_{M+1} + \lambda_{M+1}(1 - u_{M+1}^T u_{M+1}) + \sum_{i=1}^{M} \eta_i u_{M+1}^T u_i$$

- At the optimum:

$$0 = 2Cu_{M+1} - 2\lambda_{M+1} u_{M+1} + \sum_{i=1}^{M} \eta_i u_i$$

- Multiplying by u_i^T at the left, one gets $\eta_i = 0$ and thus

$$Cu_{M+1} = \lambda_{M+1} u_{M+1}$$

$\Rightarrow u_{M+1}^T$ is the eigenvector with M+1 largest eigenvalue

Mathematically

- The *i*th principal component for objects x_j is computed by $x'_{ji} = u_i^T x_j$
- The reconstructed input is thus:

$$\hat{x}_j = \sum_{i=1}^{M} x'_{ji} u_i = \sum_{i=1}^{M} (u_i^T x_j) u_i$$

- PCA also minimizes the reconstruction error:

$$\arg \max_{u_1,\ldots,u_M} \frac{1}{N} \sum_{i=1}^{N} ||x_j - \hat{x}_j||^2$$

Algorithm 1

Recover basis: Calculate $XX^\top = \sum_{i=1}^{t} x_i x_i^\top$ and let U = eigenvectors of $XX^\top$ corresponding to the top d eigenvalues.

Encode training data: $Y = U^\top X$ where Y is a $d \times t$ matrix of encodings of the original data.

Reconstruct training data: $\hat{X} = UY = UU^\top X$.

Encode test example: $y = U^\top x$ where y is a d-dimensional encoding of x.

Reconstruct test example: $\hat{x} = Uy = UU^\top x$.

Table 1.1: *Direct PCA Algorithm*

Each component is a linear combination of the original variables

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
-0.39	-0.38	0.29	0.65	0.15	0.73	-0.57	0.91	-0.89	-0.17
-2.3	-1.2	-4.5	-0.15	0.86	-0.85	0.43	-0.19	-0.83	-0.4
0.9	0.4	-0.11	0.62	0.94	0.97	0.1	-0.41	0.01	0.1
-0.82	-0.31	0.14	0.22	-0.49	-0.76	0.27	0	-0.43	-0.81
0.71	0.39	-0.09	0.26	-0.46	-0.05	0.46	0.39	-0.01	0.64
-0.25	0.27	-0.81	-0.42	0.62	0.54	-0.67	-0.15	-0.46	0.69

PC1	PC2
0.62	-0.33
-2.3	-1.2
0.88	0.31
-0.18	-0.05
-0.39	-0.01
-0.61	0.53

Scores for each sample and PC

$$PC1=0.2*A1+3.4*A2-4.5*A3$$

$$PC2=0.4*A4+5.6*A5+2.3*A7$$

…

$$VAR(PC1)=4.5 \rightarrow 45\%$$

$$VAR(PC2)=3.3 \rightarrow 33\%$$

…

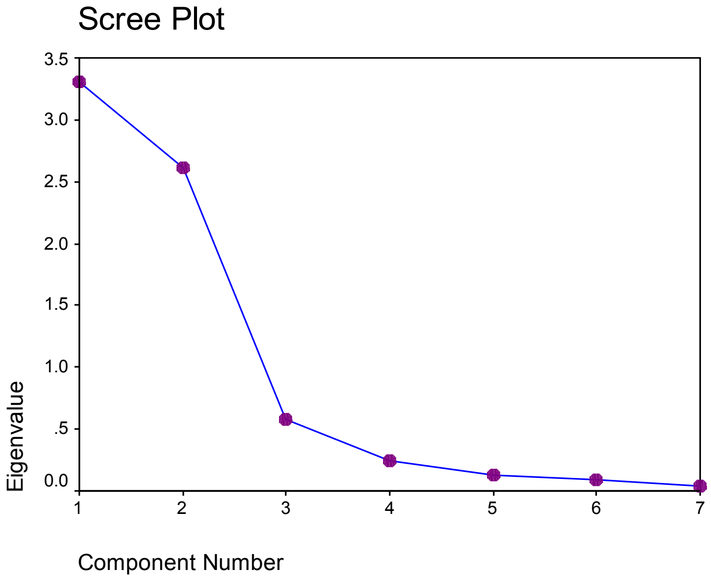
Loading of a variable

- Gives an idea of its importance in the component
- Can be use for feature selection

For each component, we have a measure of the percentage of the **variance** of the initial data that it contains

How many components?

- Scree plot: plots eigenvalues (variance) of each component in decreasing order



- Rules of thumb:
  - remove components with eigenvalues lower than 1
  - select k at the "knee" of the curve (where the scree (debris) starts to accumulate)

Illustration (1/3)

(Hastie et al., 2009)

X =

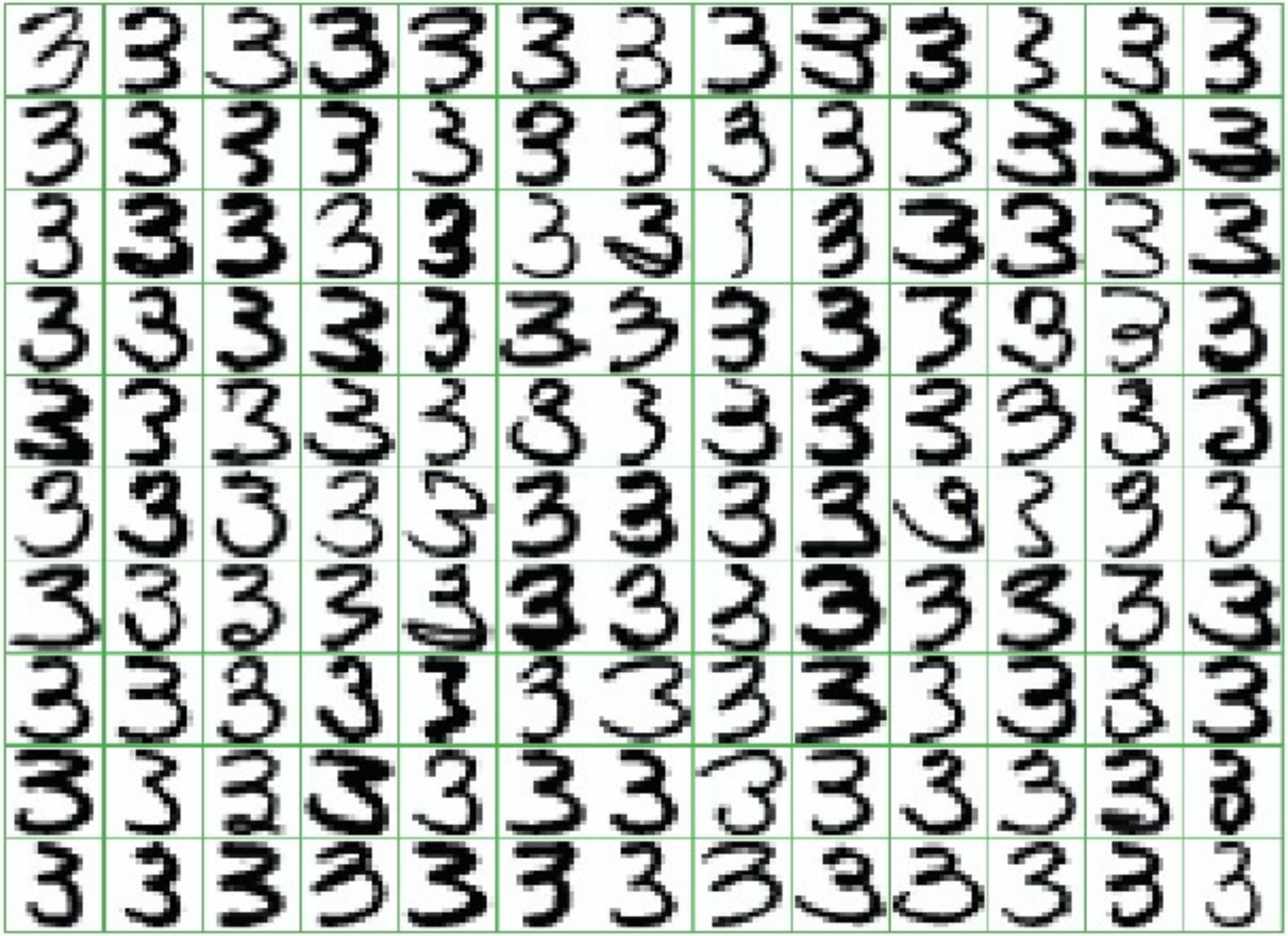


FIGURE 14.22. *A sample of* 130 *handwritten 3's shows a variety of writing styles.*

Illustration (1/3)

(Hastie et al., 2009)

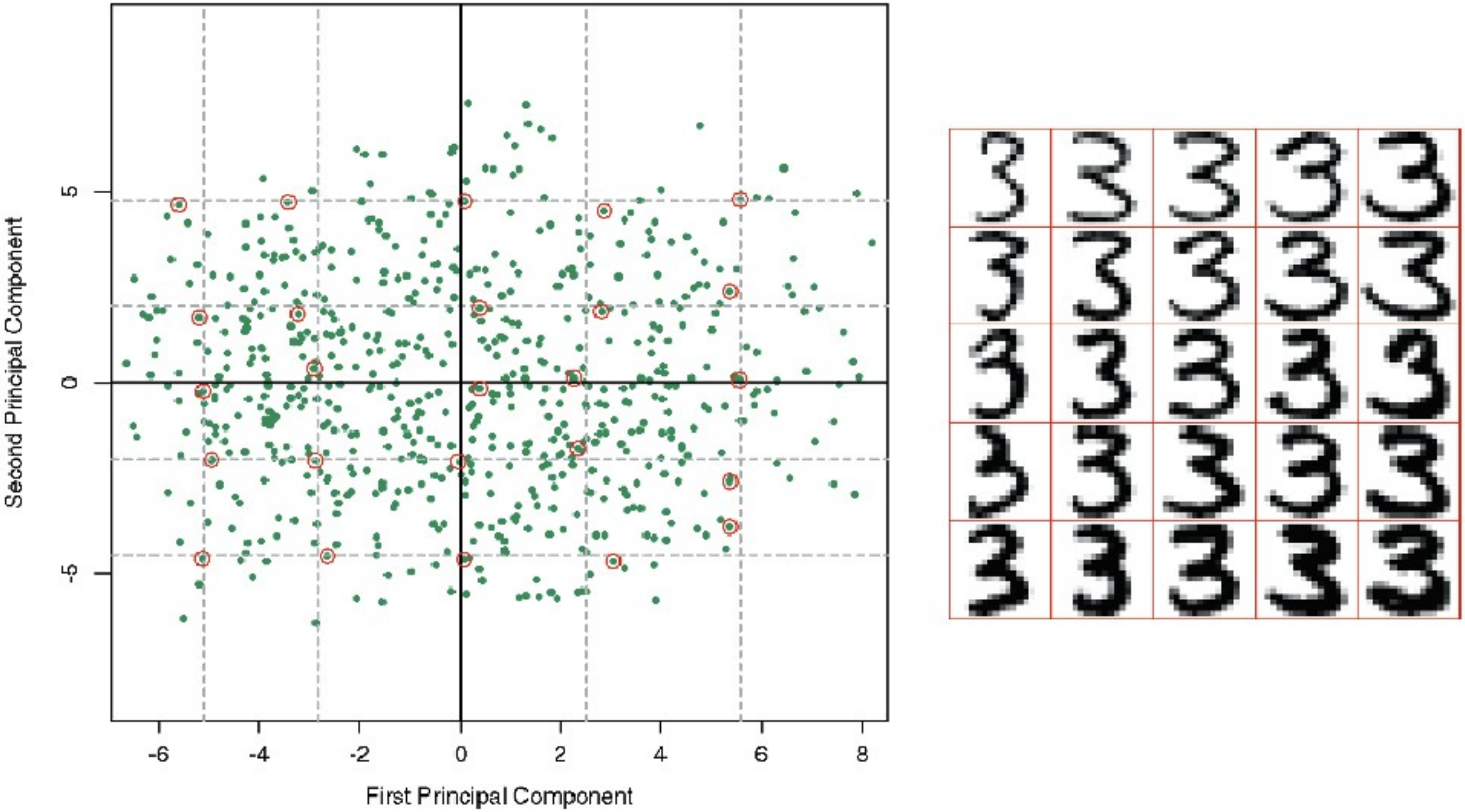


FIGURE 14.23. *(Left panel:) the first two principal components of the handwritten threes. The circled points are the closest projected images to the vertices of a grid, defined by the marginal quantiles of the principal components. (Right panel:) The images corresponding to the circled points. These show the nature of the first two principal components.*

Illustration (1/3)

(Hastie et al., 2009)

$$\hat{f}(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2$$

$=$ $+ \lambda_1 \cdot$ $+ \lambda_2 \cdot$.

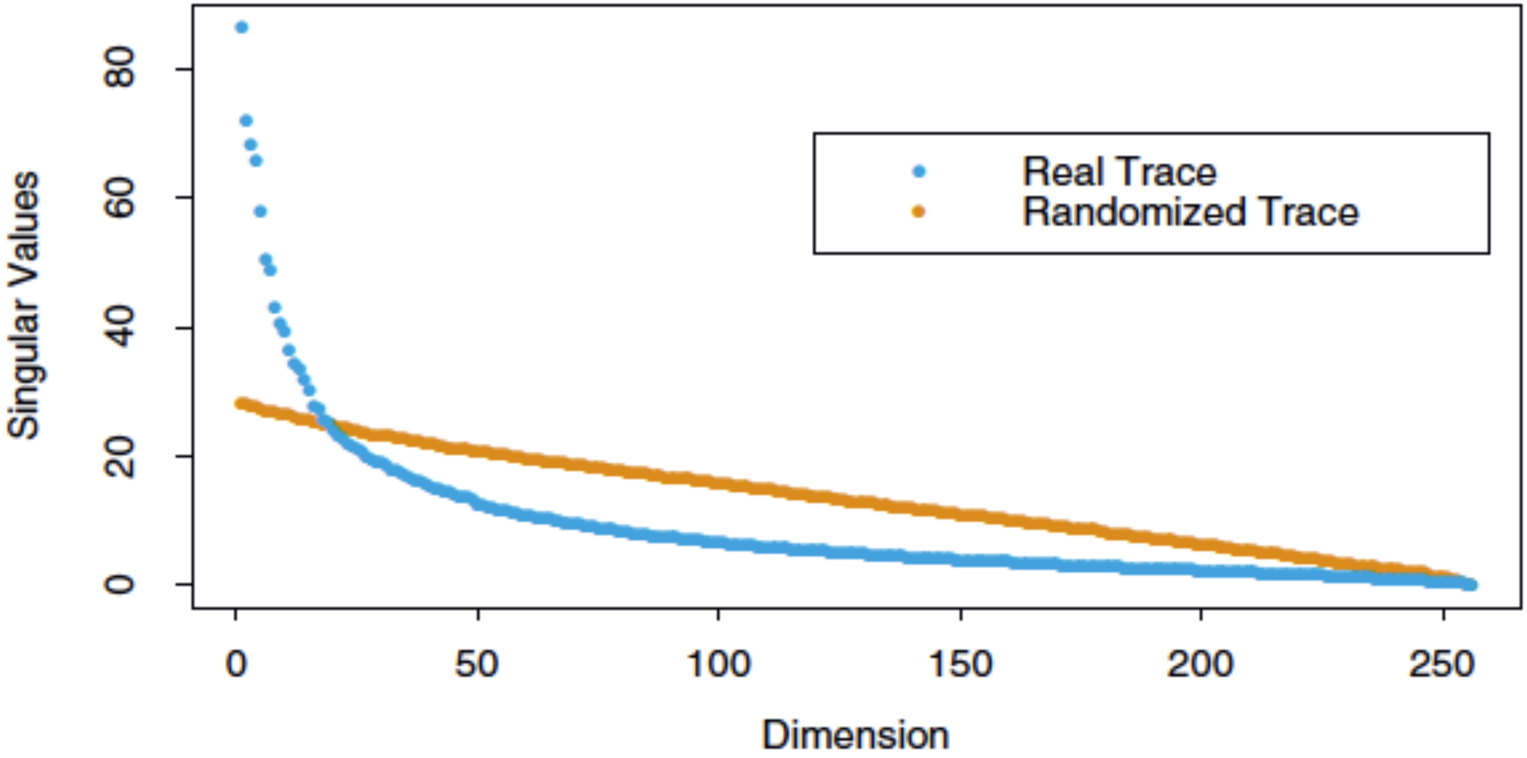


Illustration (2/3)

Holmes et al., *Nature*, *Vol. 453, No. 15, May 2008*

- Investigation of metabolic phenotype variation across and within four human populations (17 cities from 4 countries: China, Japan, UK, USA)
- ^{1}H NMR spectra of urine specimens from 4630 participants
- PCA plots of median spectra per population (city) and gender

Holmes et al., Nature, *Vol. 453, No. 15, May 2008*

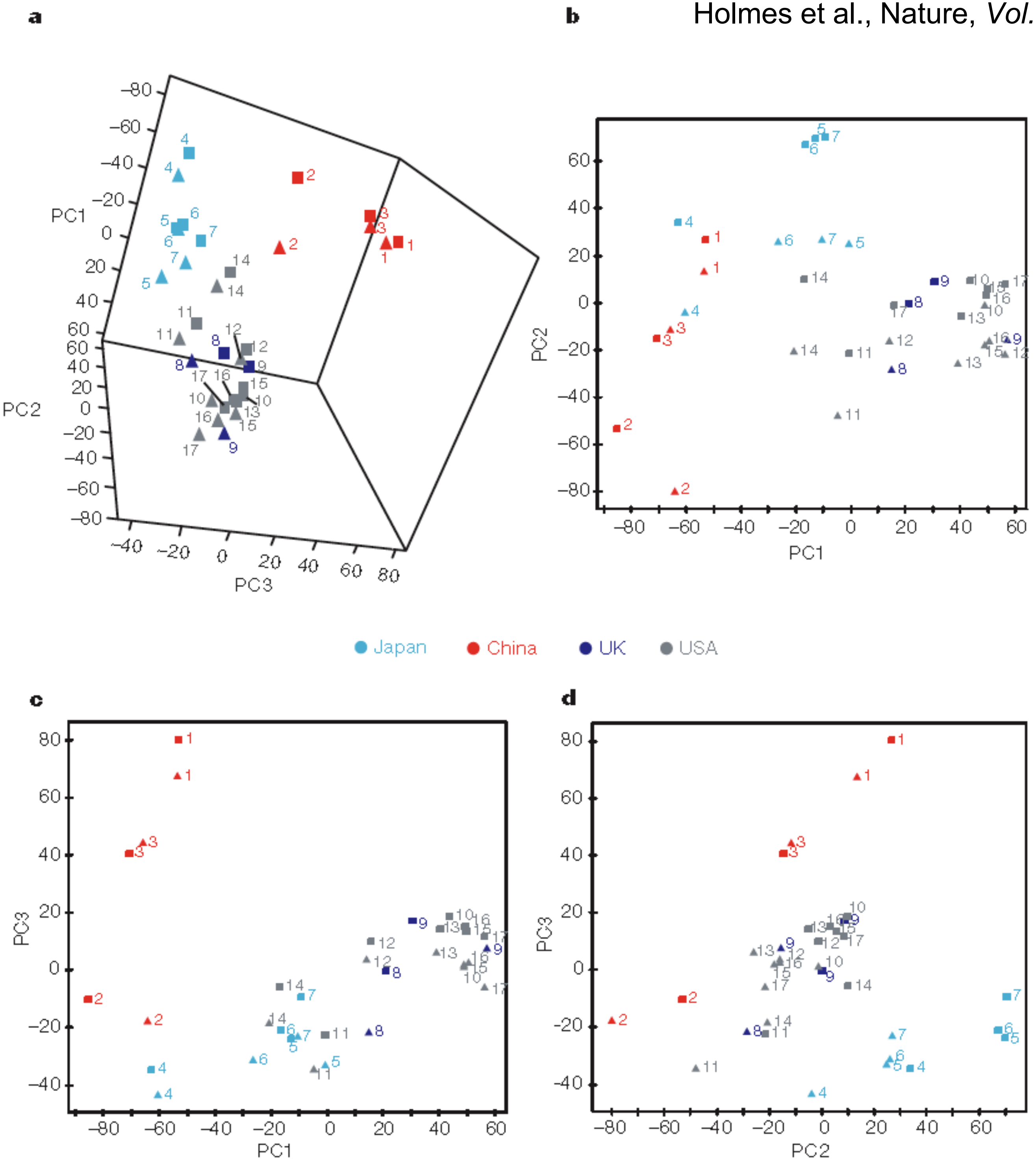


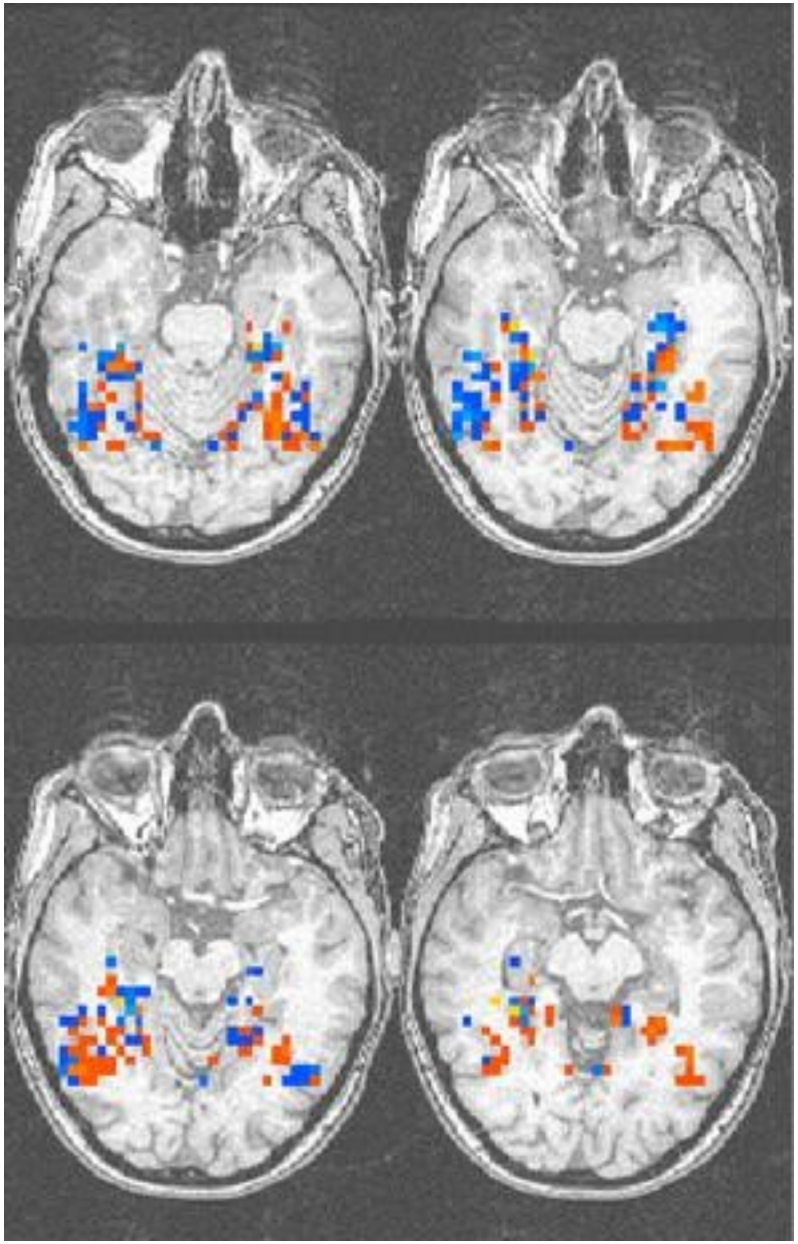
Illustration (3/3)

Neuroimaging

L voxels (brain regions)

A1	A2	A3	A4	A5	...	A7	A8
-0.91	0.74	0.74	0.97	-0.06	...	-0.04	-0.73
-2.3	-1.2	-4.5	0.47	0.13	...	0.16	0.26
-0.98	-0.46	0.98	0.77	-0.14	...	0.44	-0.12
0.97	-0.64	-0.3	-0.14	-0.29	...	-0.43	0.27
-0.64	-0.34	0.21	-0.57	-0.39	...	0.02	-0.61
0.41	-0.95	0.21	-0.17	-0.68	...	0.11	0.49

N patients/brain maps

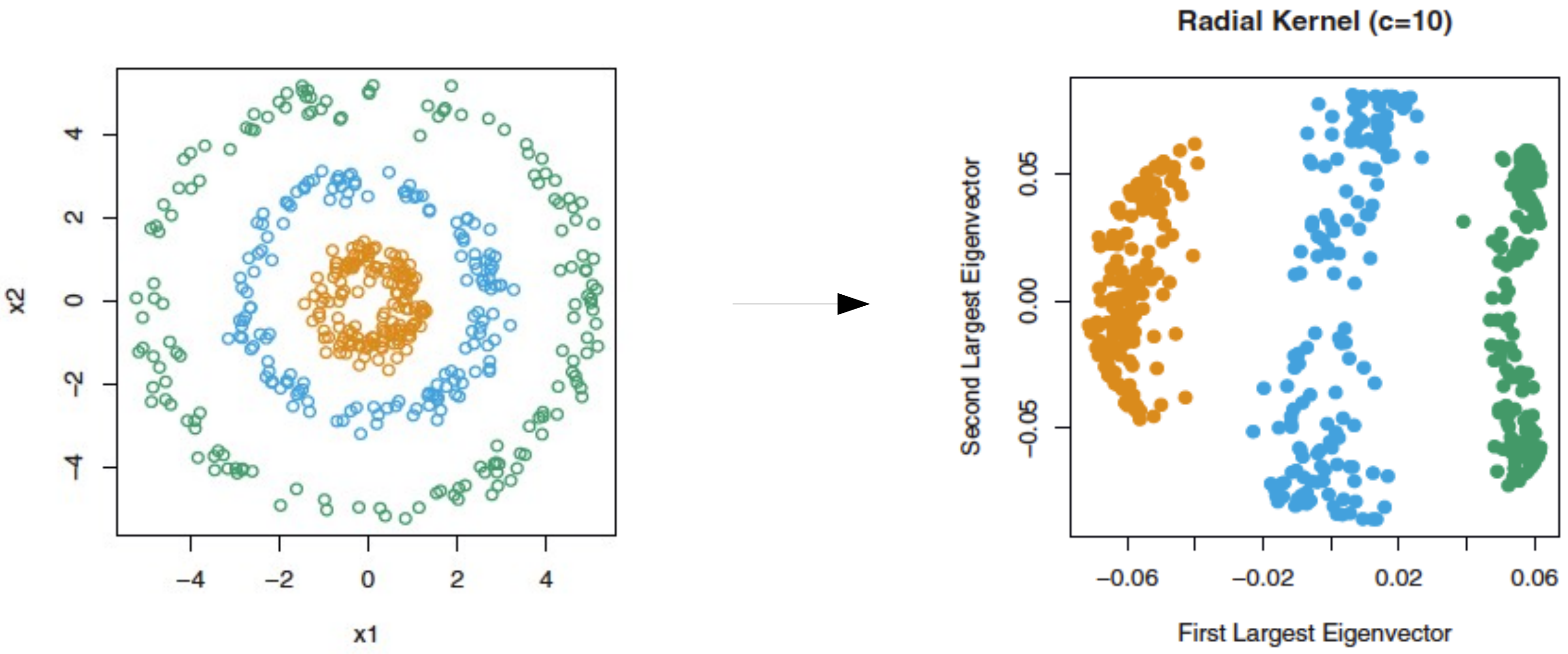


Limitations of PCA

- PCA may be used to retrieve (visually) a priori determined groups, but:
  - If PCA fails at recovering known groups, you can not conclude anything
  - Indirect with respect to clustering methods
- PCA may be used for feature selection:
  - First components may not be related at all to the output
  - Better addressed by (supervised) feature selection methods
- It should be only considered as an exploratory tools

Extensions of PCA

- Kernel PCA: non-linear feature extraction technique based on a kernelization of PCA

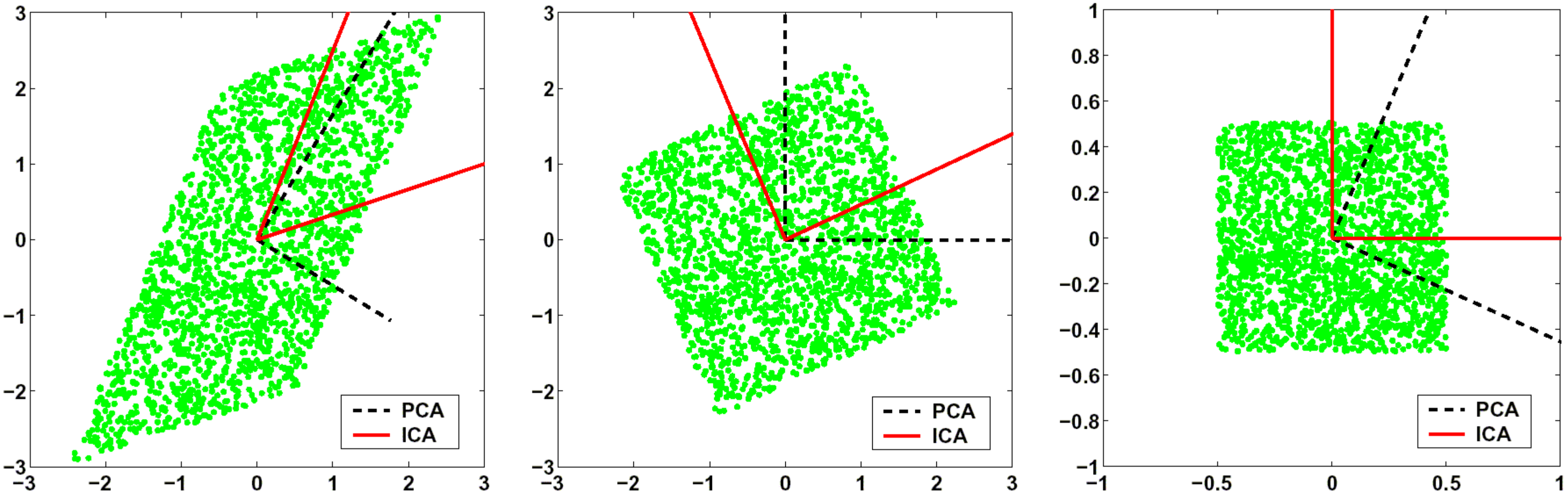


- Sparse PCA: find components with sparse loadings (few components with non-zero weights). eg., uses L1 penalization (like LASSO)

Other dimensionality reduction techniques

Independent Component Analysis:

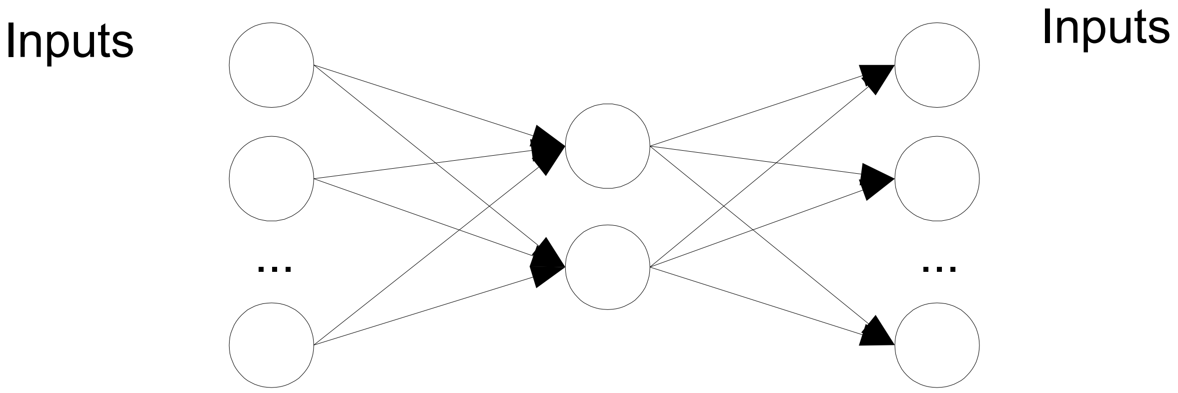
- find independent instead of orthogonal components



(a) Original (b) After PCA pre-whitening (c) After ICA projection

Other dimensionality reduction techniques

- Auto-encoder with neural networks: non-linear embedding



- Multi-dimensional scaling (MDS):
  - find new coordinates such that some distances are respected (in the least-square sense)
  - Find $z_1, z_2, \ldots, z_N \in \mathbb{R}^k$ that minimize:

$$S_M(z_1, z_2, \ldots, z_N) = \sum_{i \neq i'} (d_{ii'} - ||z_i - z_{i'}||)^2.$$

...

Other unsupervised methods

- Association rules
- Density estimation
  - Mixture models
  - Bayesian networks

References and acknowledgements

- Acknowledgements:
  - Several slides borrowed from Jörg Rahnenführer
  - http://www.statistik.tu-dortmund.de/rahnenfuehrer.html
- References:
  - Hastie et al.: chap14 (Clustering: 14.3, PCA: 14.5.1)