Analysis of Microarray Data with Methods from Machine Learning and Network Theory

Summer Lecture 2015

Prof. Dr. A. B. Cremers
Dr. Jörg Zimmermann

Machine Learning and Datamining

We are drowning in information and starving for knowledge.

Rutherford D. Roger

Machine Learning and Datamining

Torture your data until they confess.

Anonymous

Clustering

- Historically, *objects* are clustered into *groups*
  - periodic table of the elements (chemistry)
  - taxonomy (zoology, botany)
- Why cluster?
  - Understand the global structure of the data: see the forest instead of the trees
  - detect heterogeneity in the data, e.g. different tumor classes
  - Find biological pathways (cluster gene expression profiles)
  - Find data outliers (cluster microarray samples)

Classification, Clustering and Prediction

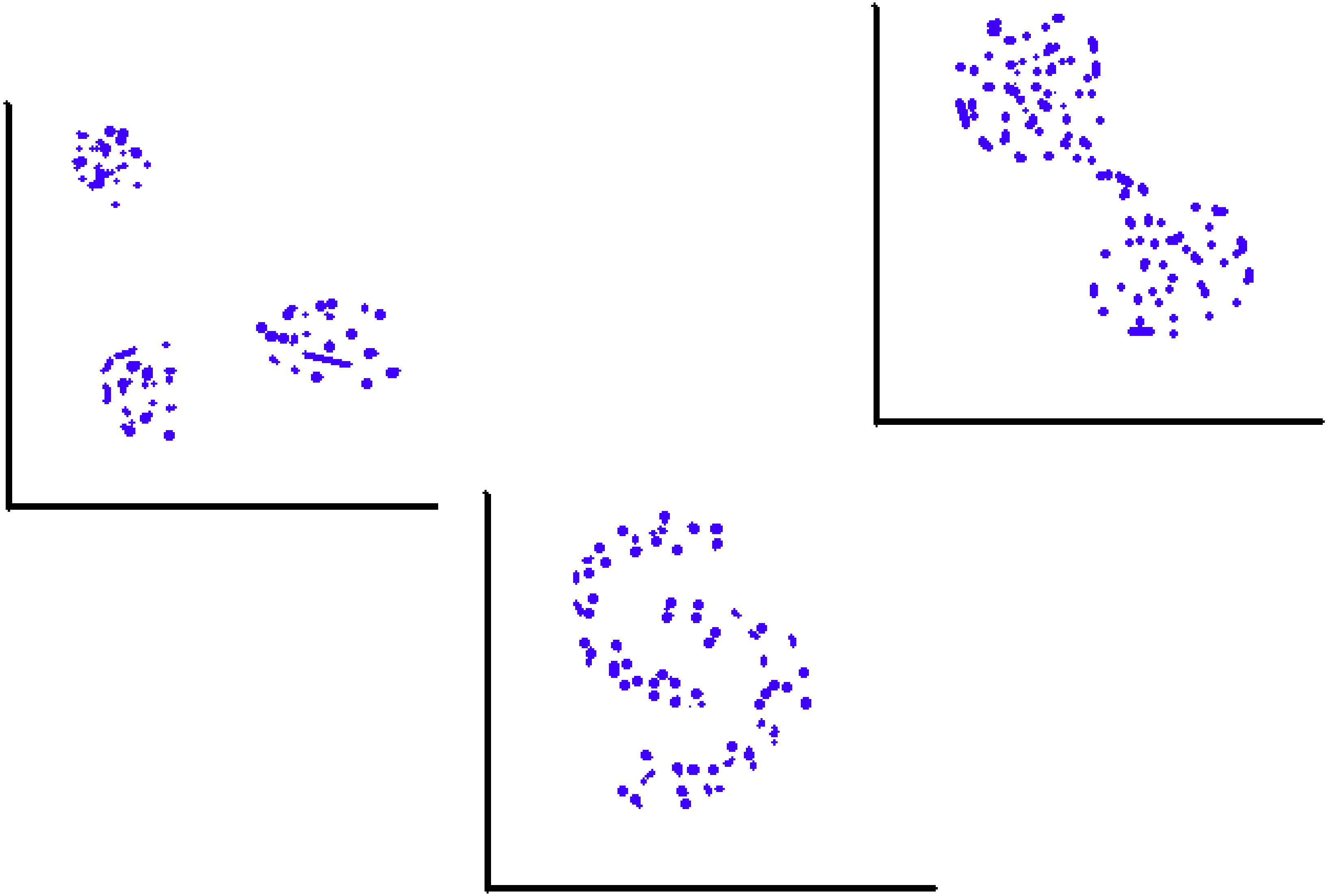
WARNING

- many people talk about “classification” when they mean clustering (unsupervised learning)
- Other people talk about classification when they mean prediction (supervised learning)
- Common denominator: classification divides objects into groups based on a set of values
- Unlike a theory, clustering is neither true nor false, and should be judged largely on the usefulness of results.
- CLUSTERING IS AND ALWAYS WILL BE SOMEWHAT OF AN ARTFORM
- However, a classification (clustering) may be useful for suggesting a theory, which could then be tested

Cluster analysis

- Addresses the problem: Given *n* objects, each described by *p* variables (or features), derive a useful division into a number of classes
- Usually want a *partition* of objects
  - But also ‘fuzzy clustering’
  - Could also take an exploratory perspective
- ‘Unsupervised learning’

Difficulties in defining ‘cluster’



Clustering Gene Expression Data

- Can cluster *genes* (rows), e.g. to (attempt to) identify groups of co-regulated genes
  - Module detection
- Can cluster *samples* (columns), e.g. to identify tumors based on profiles
  - Class discovery
- Can cluster *both* rows and columns at the same time
  - Bi-clustering approaches

Similarity = Proximity

- *Similarity* s_{ij} indicates the strength of relationship between two objects i and j
- Usually $0 \leq s_{ij} \leq 1$
- Ex 1: absolute value of the Pearson correlation coefficient
  - Use of correlation-based similarity is quite common in gene expression studies but is in general contentious...
- Ex 2: co-expression network methods: topological overlap matrix
- Ex 3: random forest similarity

Dissimilarity measures are the input to most clustering algorithms

If the original data were collected as similarities, a monotone-decreasing function can be used to convert them to dissimilarities.

Most algorithms use (symmetric) **dissimilarities** (e.g. distances)
But the triangle inequality does *not* have to hold.
Triangle inequality:

$$d_{ii'} \leq d_{ik} + d_{ki'}$$

Dissimilarity and Distance

- Associated with similarity measures s_{ij} bounded by 0 and 1 is a *dissimilarity* $d_{ij} = 1 - s_{ij}$
- *Distance* measures have the metric property ($d_{ij} + d_{jk} \geq d_{ik}$)
- Many examples: Euclidean (‘as the crow flies’), Manhattan (‘city block’), *etc*.
- Distance measure has a large effect on performance
- Behavior of distance measure related to *scale* of measurement

Partitioning Methods

- Partition the objects into a *prespecified* number of groups K
- Iteratively reallocate objects to clusters until some criterion is met (e.g. minimize within cluster sums of squares)
- Examples: k-means, self-organizing maps (SOM), partitioning around medoids (PAM), model-based clustering

Loss functions for judging clusterings

There are a lot of possibilities to judge the quality of a specific clustering. K-means tries to minimize the within-cluster sum of squares:

$$W(C) = \sum_{i=1}^{k} \sum_{x \in C_i} \left(\left\| x - m_i \right\| \right)^2$$

where C = {C_1, .. C_k} is a partition of the observations {x_1, .., x_n} into k sets and m_i is the mean of data points in C_i.

K-means clustering

- Important: k-means is intended for quantitative variables, it uses a (squared) Euclidean distance (so scale variables suitably before use)
- Pre-specify number of clusters K, and cluster ‘centers’ (means, medoids or centroids)
- Minimize within cluster sum of squares from the centers
- Iterate (until cluster assignments do not change):

1. For a given cluster assignment, find the cluster means
2. For a given set of means, minimize the within cluster sum of squares by allocating each object to the closest cluster mean

Recommendations for k-means clustering

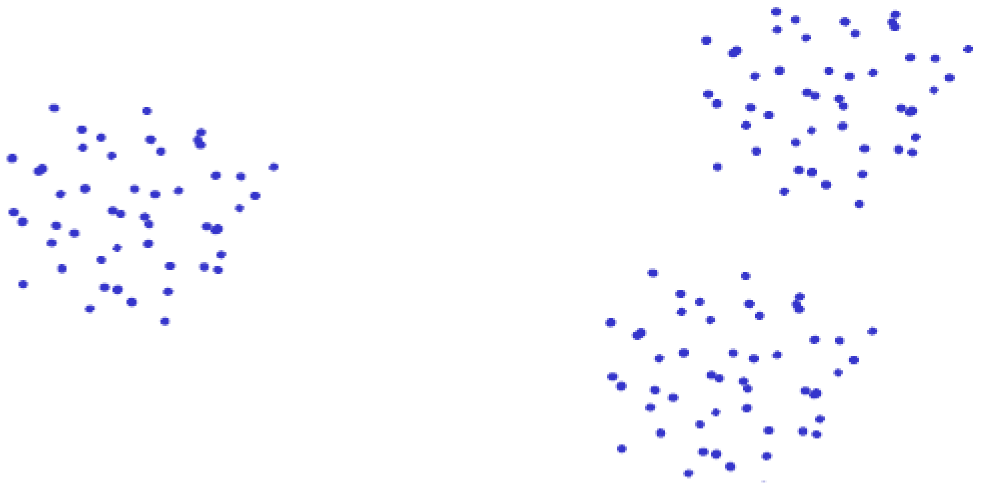
- Either: Start with many different random choices of starting means, and choose the solution having smallest value of the objective function.

- Or use another clustering method (e.g. hierarchical clustering) to determine an initial set of cluster centers.

K-Means Clustering

The K-Means algorithm is a local (or greedy) procedure, thus the result may not be the global optimal solution.

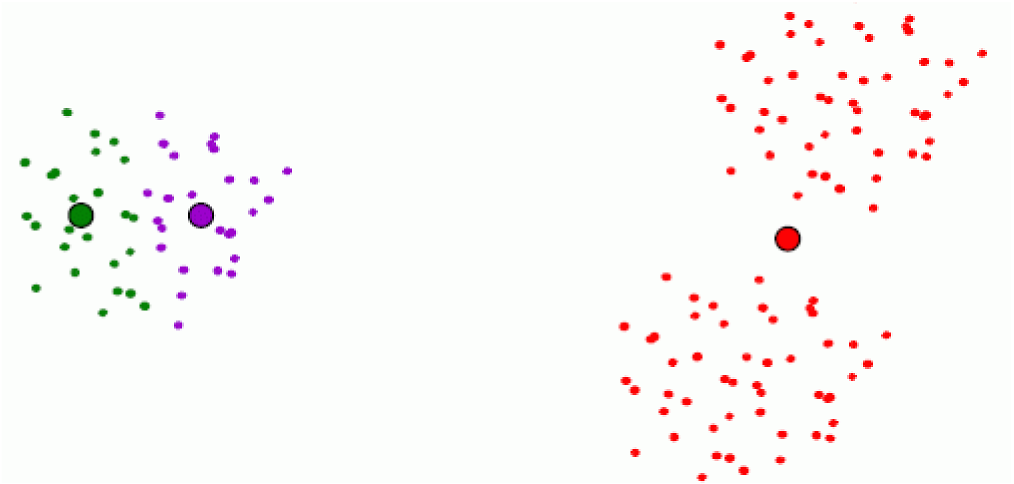
Suppose we have measured the following 2D-dataset:



What would you expect K-means to do for K=3?

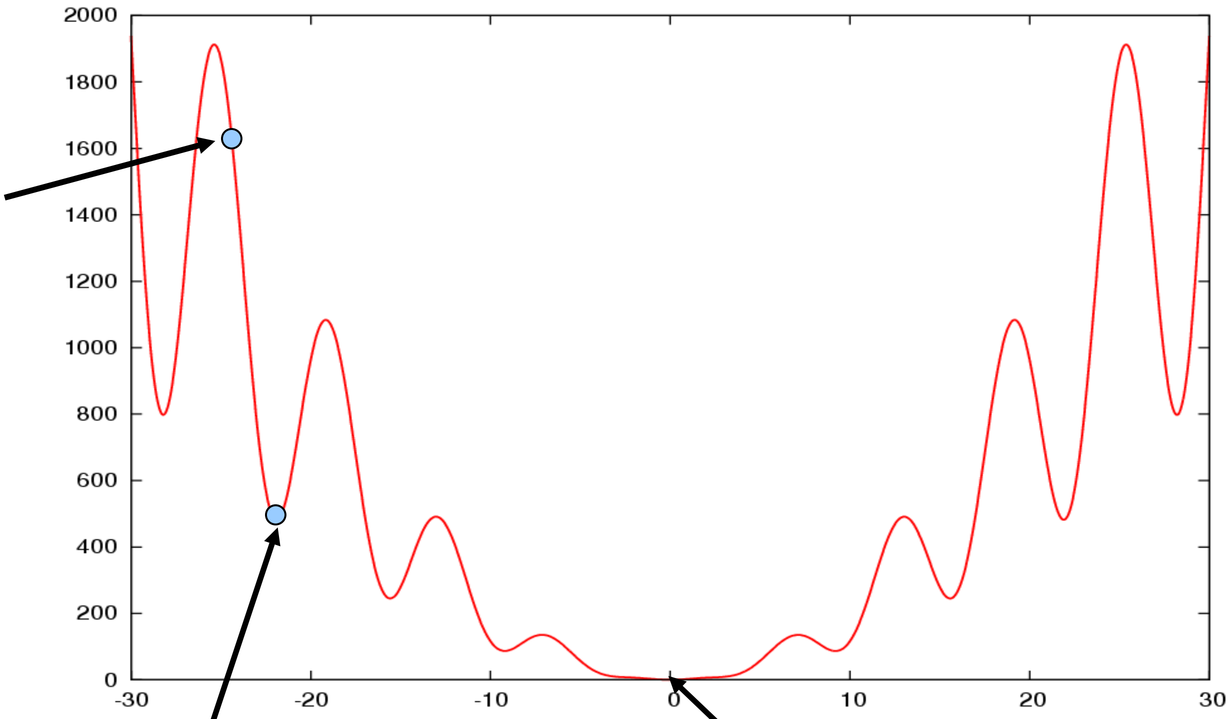
K-Means Clustering

In some cases, the randomly chosen starting points for cluster centers can lead to the following converged configuration, which has not minimal total cluster variance:



K-Means Clustering

General fact: local optimization algorithms can get stuck in local optima



If you start here

... you will end there

global minimum

K-Means Clustering

How to deal with this situation?

1. Make as many random restarts (new start configurations) as reasonably possible, and return the configuration of the best run (according to your target function, e.g. total cluster variance)

2. Use a heuristic to select starting points for cluster centers, e.g.:

   Select first starting point randomly among all data points.

   Select data point which is as far as possible from first starting point as second starting point.

   .

   .

   Select data point which is as far as possible from nearest already selected starting point as *j*. starting point.

   .

   .

Hierarchical clustering

- Hierarchical clustering algorithms begin with every observation representing a singleton cluster.

- At each of N-1 steps the closest 2 (least dissimilar) clusters are merged into a single cluster.

- Therefore a measure of “inter-group dissimilarity” between 2 clusters A and B must be defined.
- average linkage = mean distance between pairs of objects
- complete linkage = maximal distance between pairs of objects
- single linkage = minimal distance between pairs of objects

Here pairs of objects consist of objects of cluster A and objects of cluster B.

Dendrogram

Recursive binary splitting/agglomeration can be represented by a rooted binary tree.

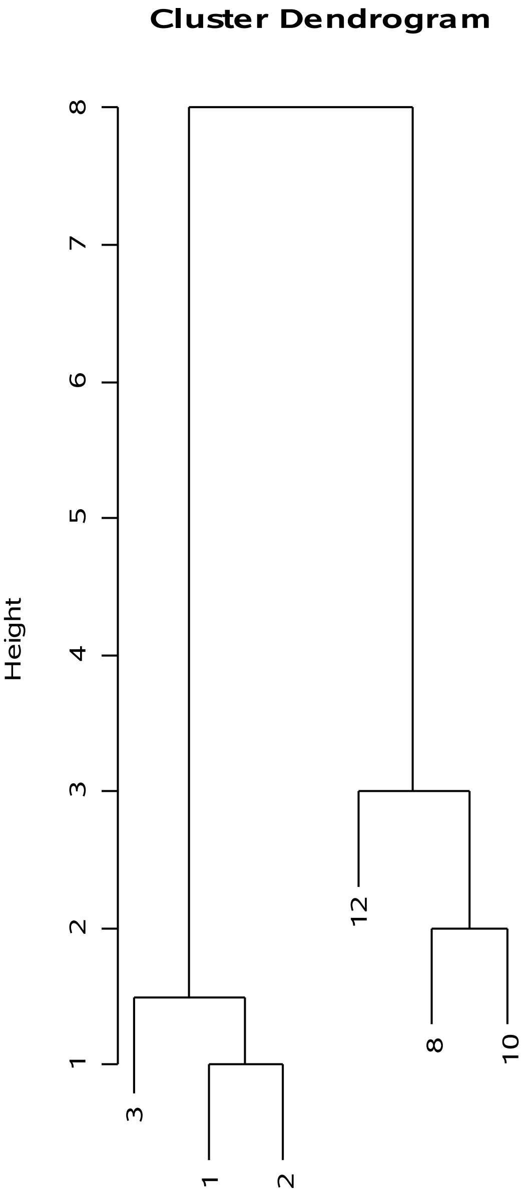
The root node represents the entire data set.

The N terminal nodes of the trees represent individual observations.

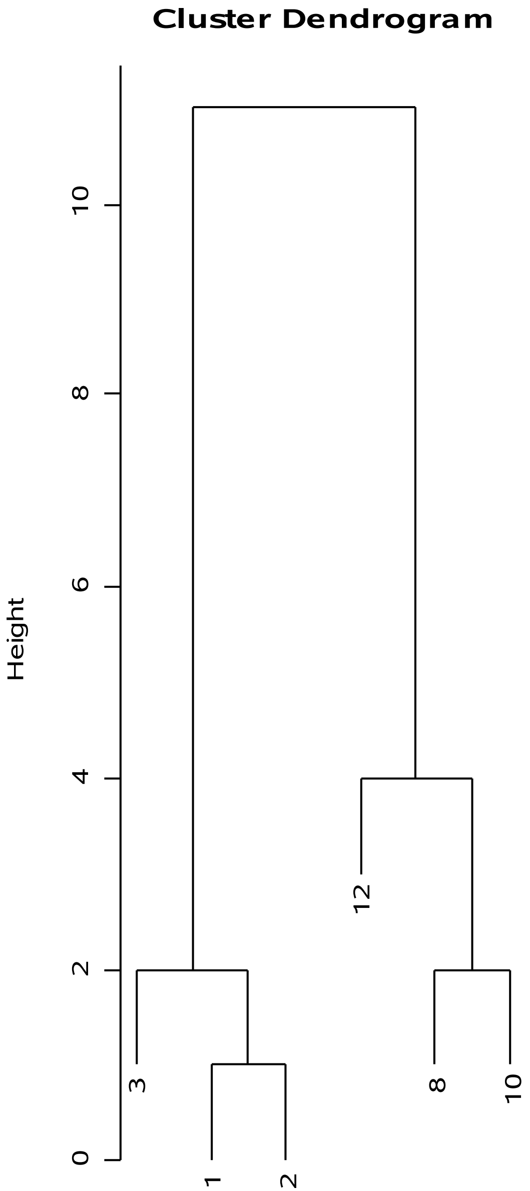
Each nonterminal node ("parent") has two daughter nodes.

Thus the binary tree can be plotted so that the height of each node is proportional to the value of the intergroup dissimilarity between its 2 daughters.

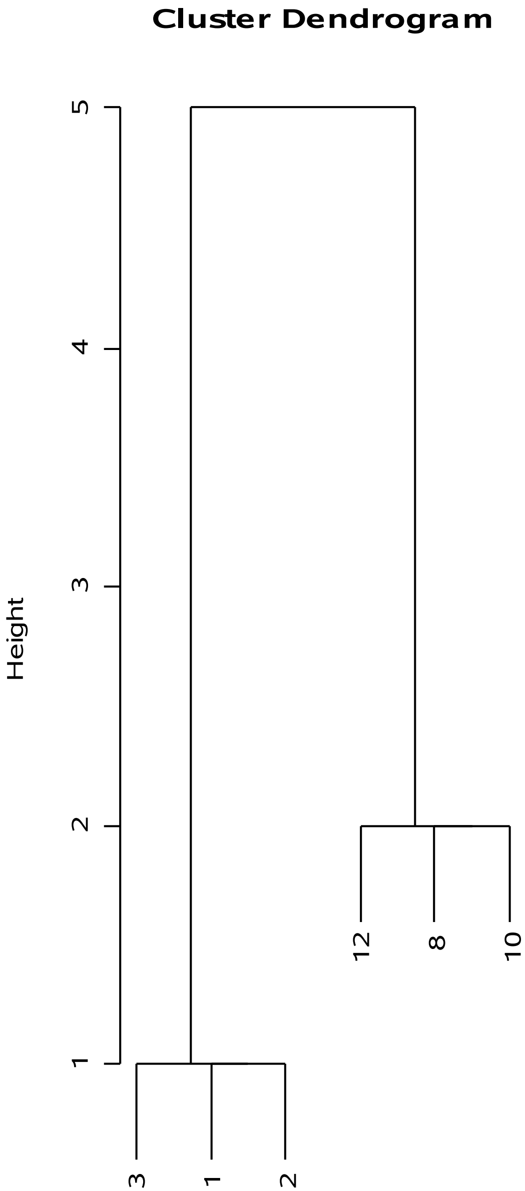
A dendrogram provides a complete description of the hierarchical clustering in graphical format.



dist(xsimple)
hclust (*, "average")



dist(xsimple)
hclust (*, "complete")



dist(xsimple)
hclust (*, "single")

Partitioning vs. Hierarchical

- *Partitioning*
  - Advantage: Provides clusters that satisfy some optimality criterion (approximately)
  - Disadvantages: Need initial K, long computation time
- *Hierarchical*
  - Advantage: Fast computation (agglomerative)
  - Disadvantages: harder to interpret, need for additional criteria
- Word on the street: most data analysts prefer hierarchical clustering over partitioning methods when it comes to gene expression data

Generic Clustering Tasks

- Estimating number of clusters
- Assigning each object to a cluster
- Assessing strength/confidence of cluster assignments for individual objects
- Assessing cluster homogeneity

R: clustering

- A number of **R** packages (libraries) contain functions to carry out clustering.
- Important R functions for clustering
  - **`Kmeans`**
  - **`hclust`**
  - **`pam`** (part of the cluster package)
  - **`mclust`**: model-based clustering