Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Cluster Analysis

Applied Multivariate Statistics – Spring 2012

Overview

- Hierarchical Clustering: Agglomerative Clustering
- Partitioning Methods: K-Means and PAM
- Gaussian Mixture Models

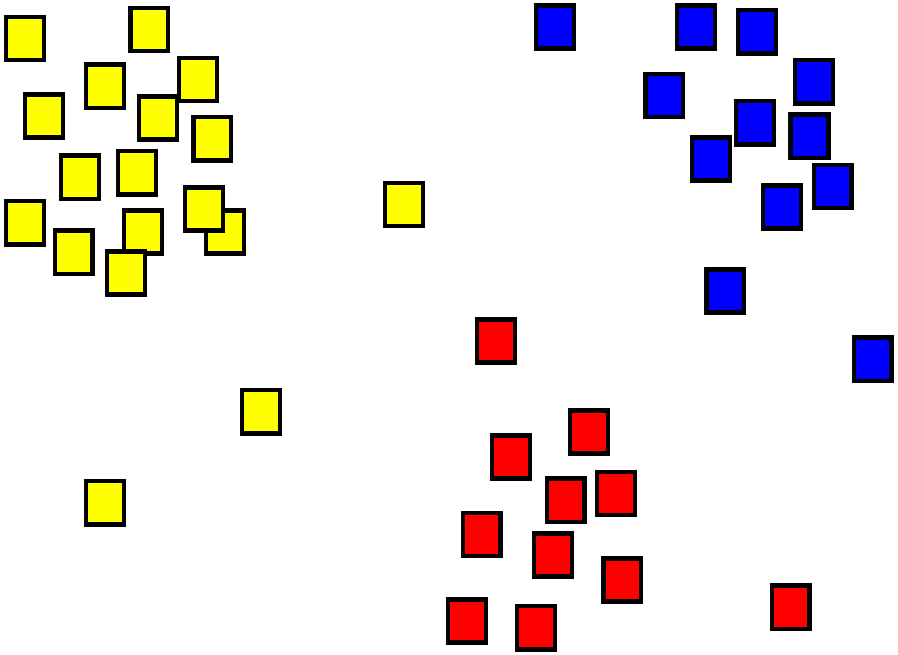
Goal of clustering

- Find groups, so that elements within cluster are very similar and elements between cluster are very different
  Problem: Need to interpret meaning of a group

- Examples:
  - Find customer groups to adjust advertisement
  - Find subtypes of diseases to fine-tune treatment
- Unsupervised technique: No class labels necessary

- N samples, k cluster: k^N possible assignments
  E.g. N=100, k=5: $5^{100} = 7*10^{69}$!!
  Thus, impossible to search through all assignments

Clustering is useful in 3+ dimensions



Human eye is extremely good at clustering

Use clustering only,
if you can not look at
the data
(i.e. more than 2 dimensions)

Hierarchical Clustering

- Agglomerative: Build up cluster from individual observations

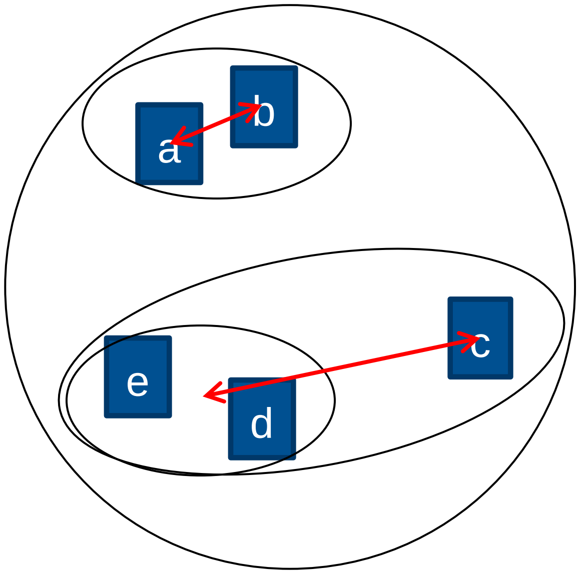
- Divisive: Start with whole group of observations and split off clusters

- Divisive clustering has much larger computational burden
  We will focus on agglomerative clustering

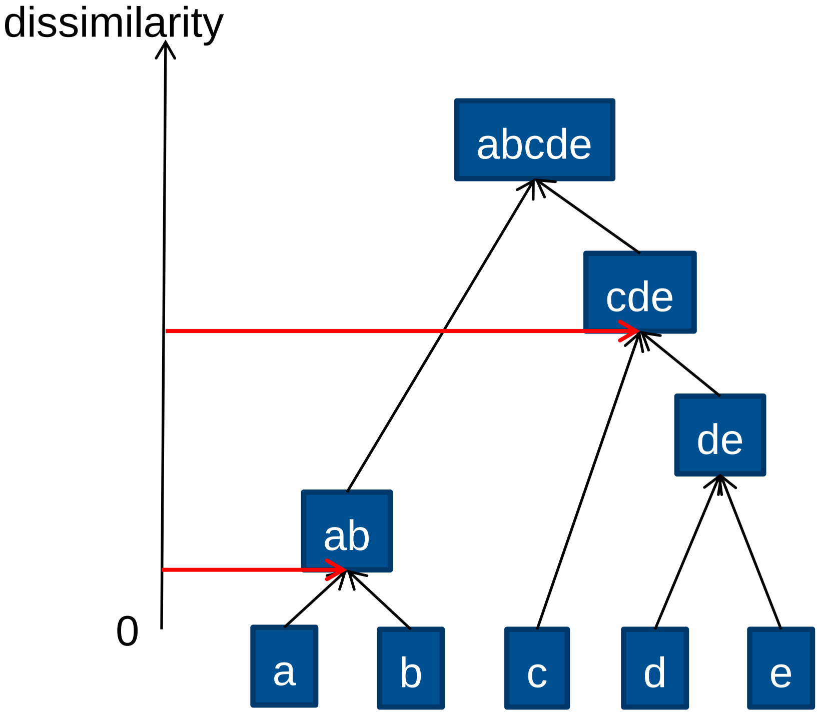
- Solve clustering for all possible numbers of cluster (1, 2, …, N) at once
  Choose desired number of cluster later

Agglomerative Clustering

Data in 2 dimensions



Clustering tree = Dendrogramm



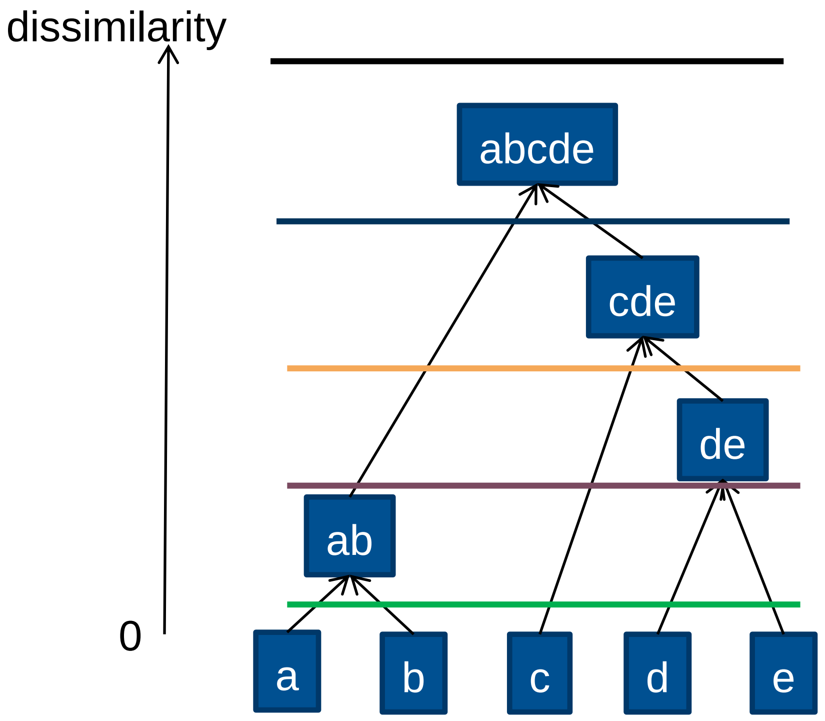
Join samples/cluster that are closest until only one cluster is left

Agglomerative Clustering: Cutting the tree

Get cluster solutions by cutting the tree:

- 1 Cluster: abcde (trivial)
- 2 Cluster: ab - cde
- 3 Cluster: ab – c – de
- 4 Cluster: ab – c – d – e
- 5 Cluster: a – b – c – d – e

Clustering tree = Dendrogramm



Dissimilarity between samples

- Any dissimilarity we have seen before can be used
  - euclidean
  - manhattan
  - simple matching coefficent
  - Jaccard dissimilarity
  - Gower’s dissimilarity
  - etc.

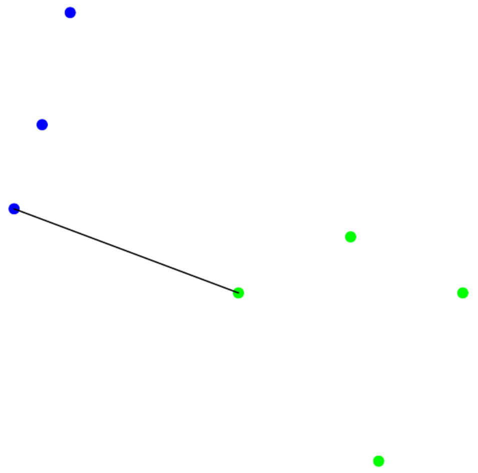
Dissimilarity between cluster

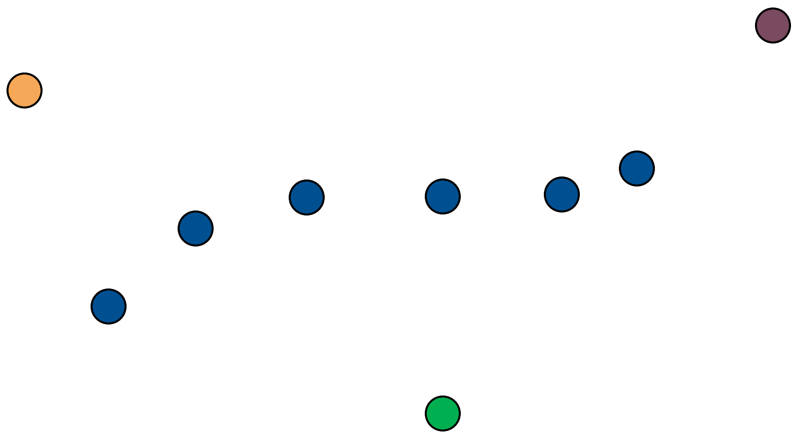
- Based on dissimilarity between samples
- Most common methods:
  - single linkage
  - complete linkage
  - average linkage
- No right or wrong: All methods show one aspect of reality
- If in doubt, I use complete linkage

Single linkage

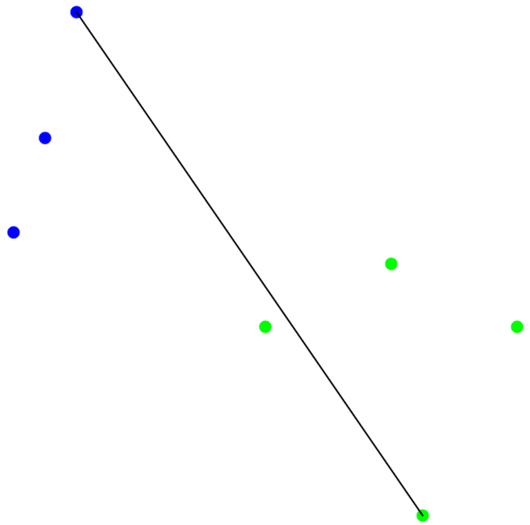
- Distance between two cluster = minimal distance of all element pairs of both cluster

- Suitable for finding elongated cluster

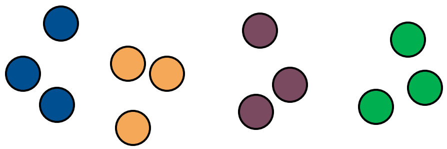




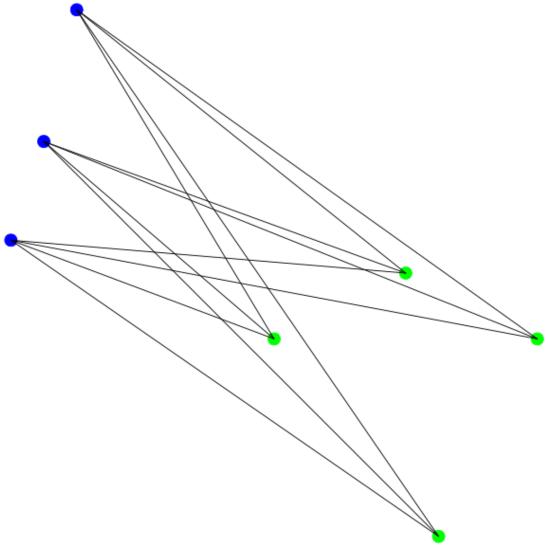
Complete linkage



- Distance between two cluster = maximal distance of all element pairs of both cluster
- Suitable for finding compact but not well separated cluster

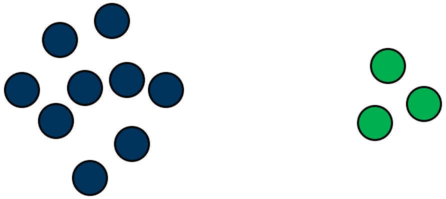


Average linkage



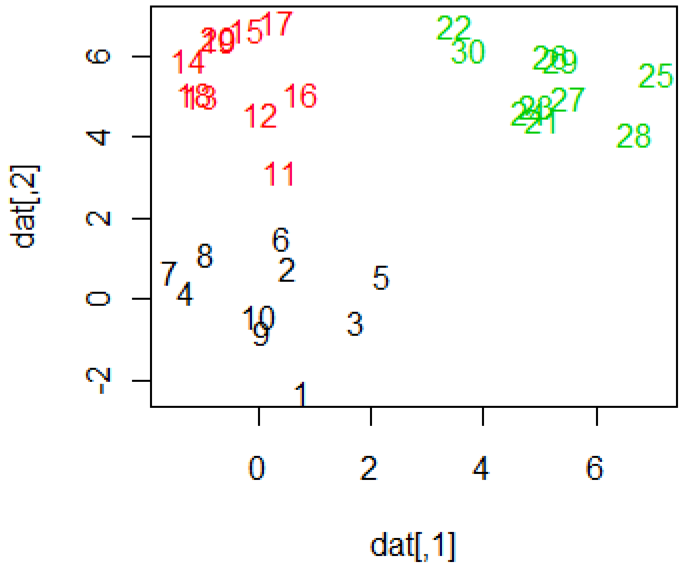
- Distance between two cluster = average distance of all element pairs of both cluster

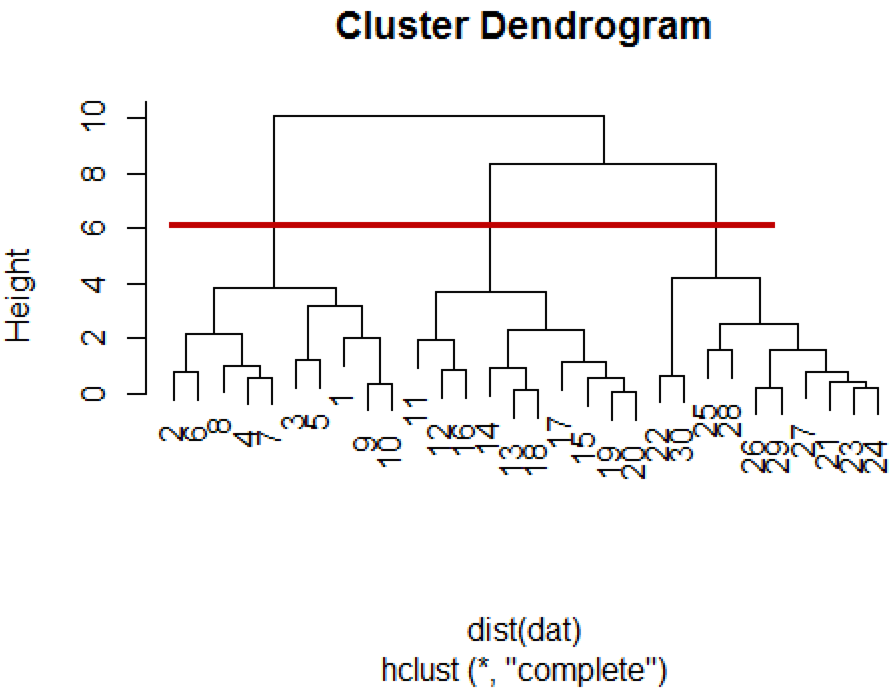
- Suitable for finding well separated, potato-shaped cluster



Choosing the number of cluster

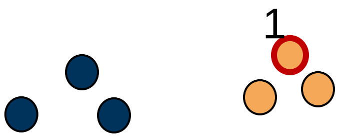
- No strict rule
- Find the largest vertical “drop” in the tree





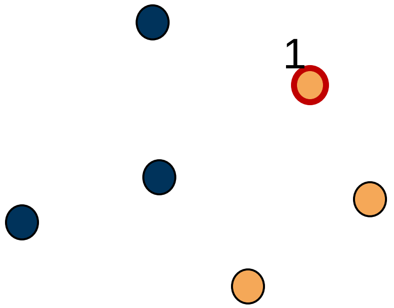
Quality of clustering: Silhouette plot

- One value S(i) in [0,1] for each observation
- Compute for each observation i:
  a(i) = average dissimilarity between i and all other points of the cluster to which i belongs
  b(i) = average dissimilarity between i and its “neighbor” cluster, i.e., the nearest one to which it does *not* belong.
  Then, $S(i) = \frac{(b(i)-a(i))}{\max(a(i),b(i))}$
- S(i) large: well clustered; S(i) small: badly clustered
  S(i) negative: assigned to wrong cluster



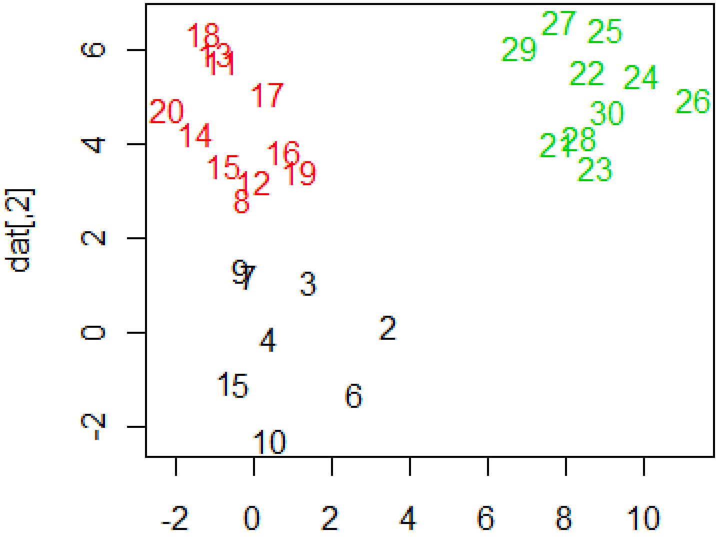
S(1) large

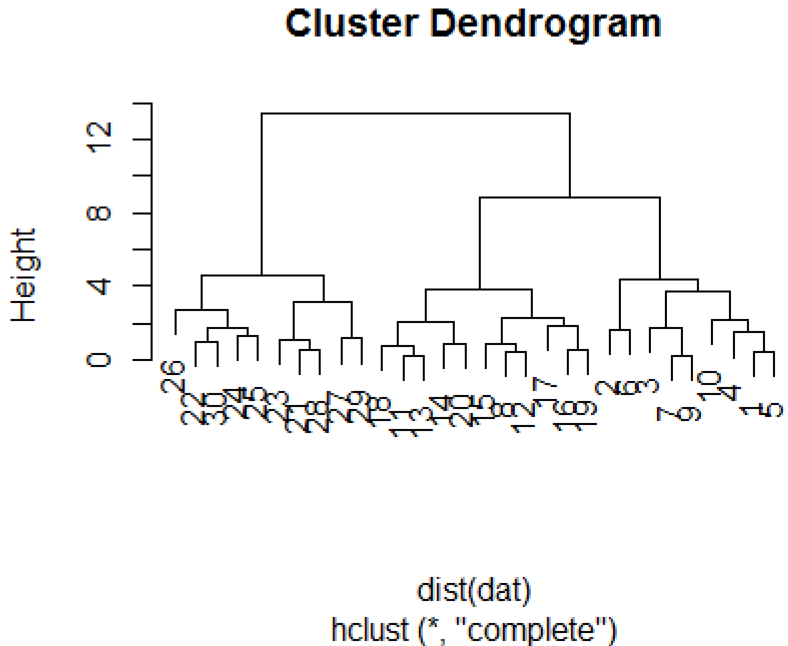
Average S over 0.5 is acceptable

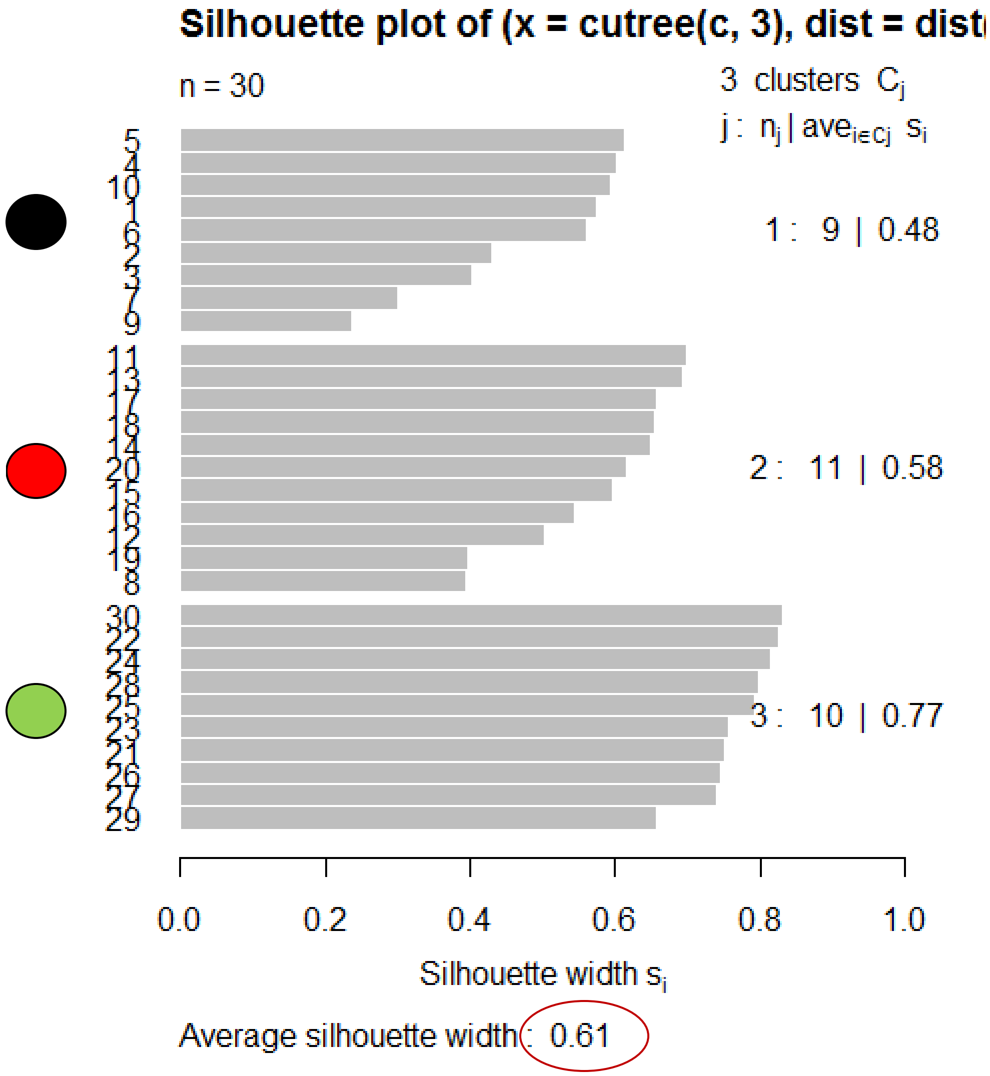


S(1) small

Silhouette plot: Example







Agglomerative Clustering in R

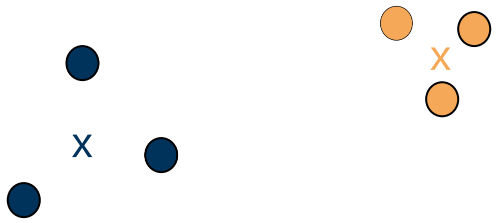
- Pottery Example

- Functions “hclust”, “cutree” in package “stats”
- Alternative: Function “agnes” in package “cluster”
- Function “silhouette” in package “cluster”

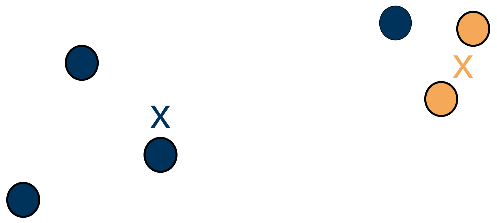
Partitioning Methods: K-Means

- Number of clusters K is fixed in advance

- Find K cluster centers μ_i and assignments, so that within-groups Sum of Squares (WGSS) is minimal
- $WGSS = \sum_{all\ Cluster\ C} \sum_{Point\ i\ in\ Cluster\ C} (x_i - \mu_i)^2$



WGSS small

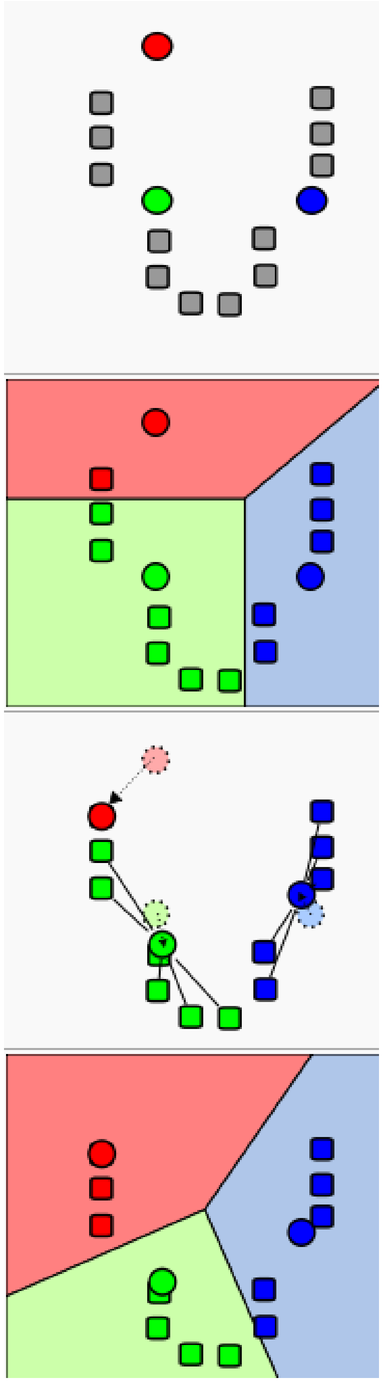


WGSS large

K-Means

- Exact solution computationally infeasible
- Approximate solutions, e.g. Lloyd’s algorithm

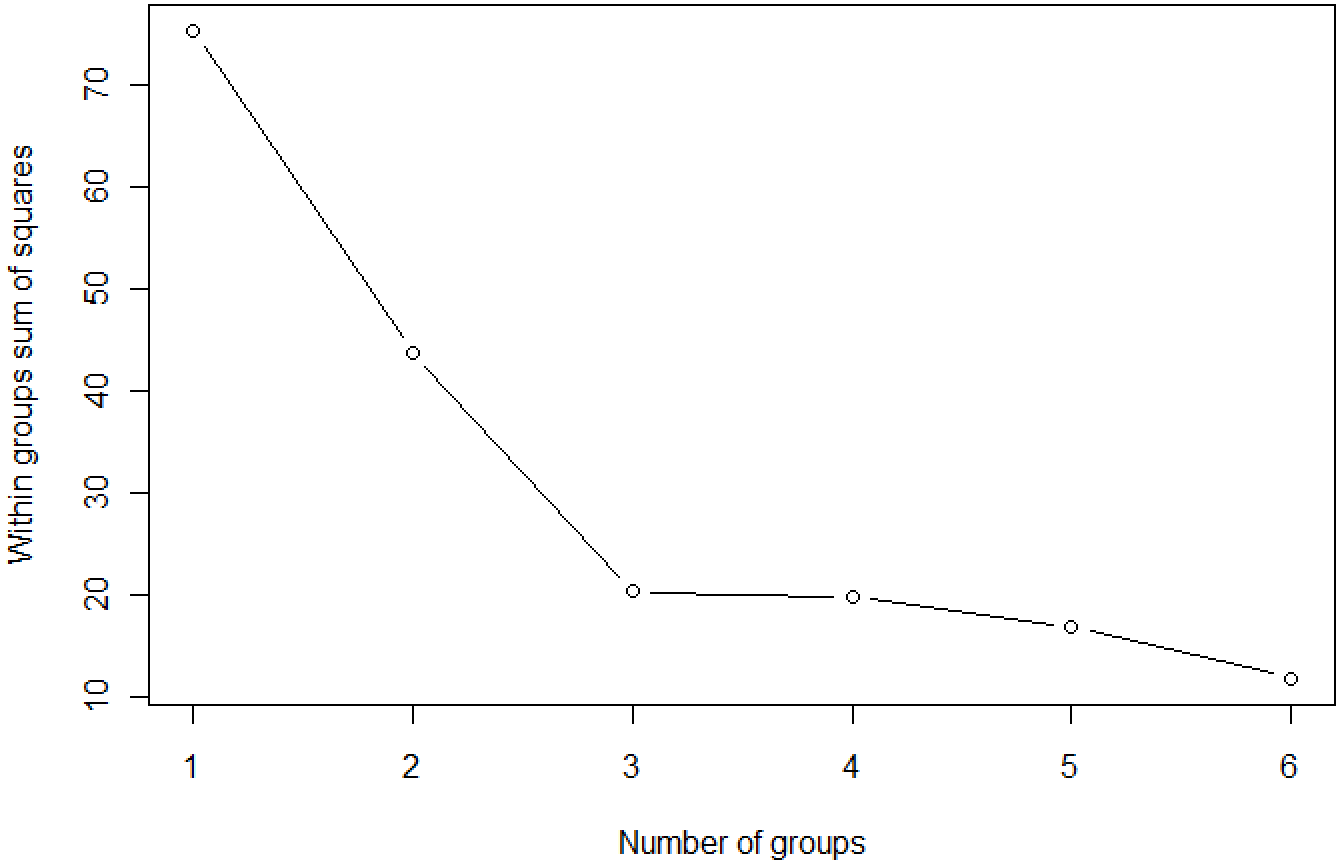
- Different starting assignments will give different solutions
  Random restarts to avoid local optima



Iterate until convergence

K-Means: Number of clusters

- Run k-Means for several number of groups
- Plot WGSS vs. number of groups
- Choose number of groups after the last big drop of



Robust alternative: PAM

- Partinioning around Medoids (PAM)
- K-Means: Cluster center can be an arbitrary point in space
  PAM: Cluster center must be an observation (“medoid”)

- Advantages over K-means:
  - more robust against outliers
  - can deal with any dissimilarity measure
  - easy to find representative objects per cluster
    (e.g. for easy interpretation)

Partitioning Methods in R

- Function “kmeans” in package “stats”
- Function “pam” in package “cluster”

- Pottery revisited

Gaussian Mixture Models (GMM)

- Up to now: Heuristics using distances to find cluster
- Now: Assume underlying statistical model
- Gaussian Mixture Model:
  $f(x; p, \theta) = \sum_{j=1}^{K} p_j g_j(x; \theta_j)$
  K populations with different probability distributions
- Example: X_1 ~ N(0,1), X_2 ~ N(2,1); p_1 = 0.2, p_2 = 0.8

$$f(x; p, \theta) = 0.2 \cdot \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) + 0.8 \cdot \frac{1}{\sqrt{2\pi}} \exp(-(x-2)^2/2)$$

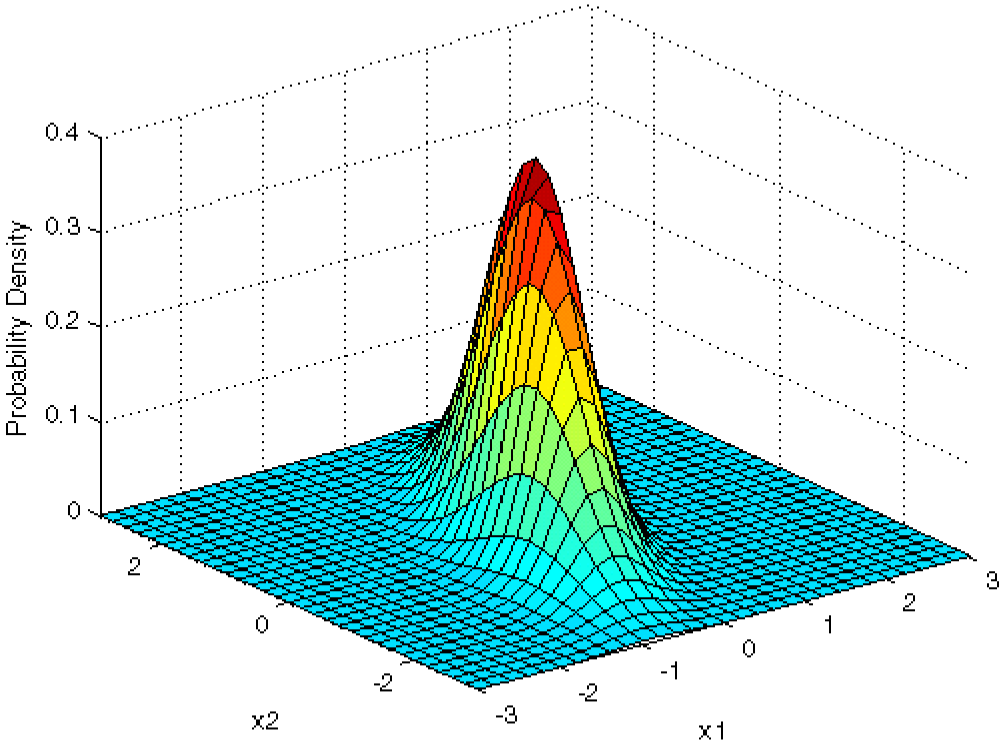
- Find number of classes and parameters p_j and θ_j given data
- Assign observation x to cluster j, where estimated value of

$$P(cluster\ j|x) = \frac{p_j g_j(x; \theta_j)}{f(x; p, \theta)}$$

is largest

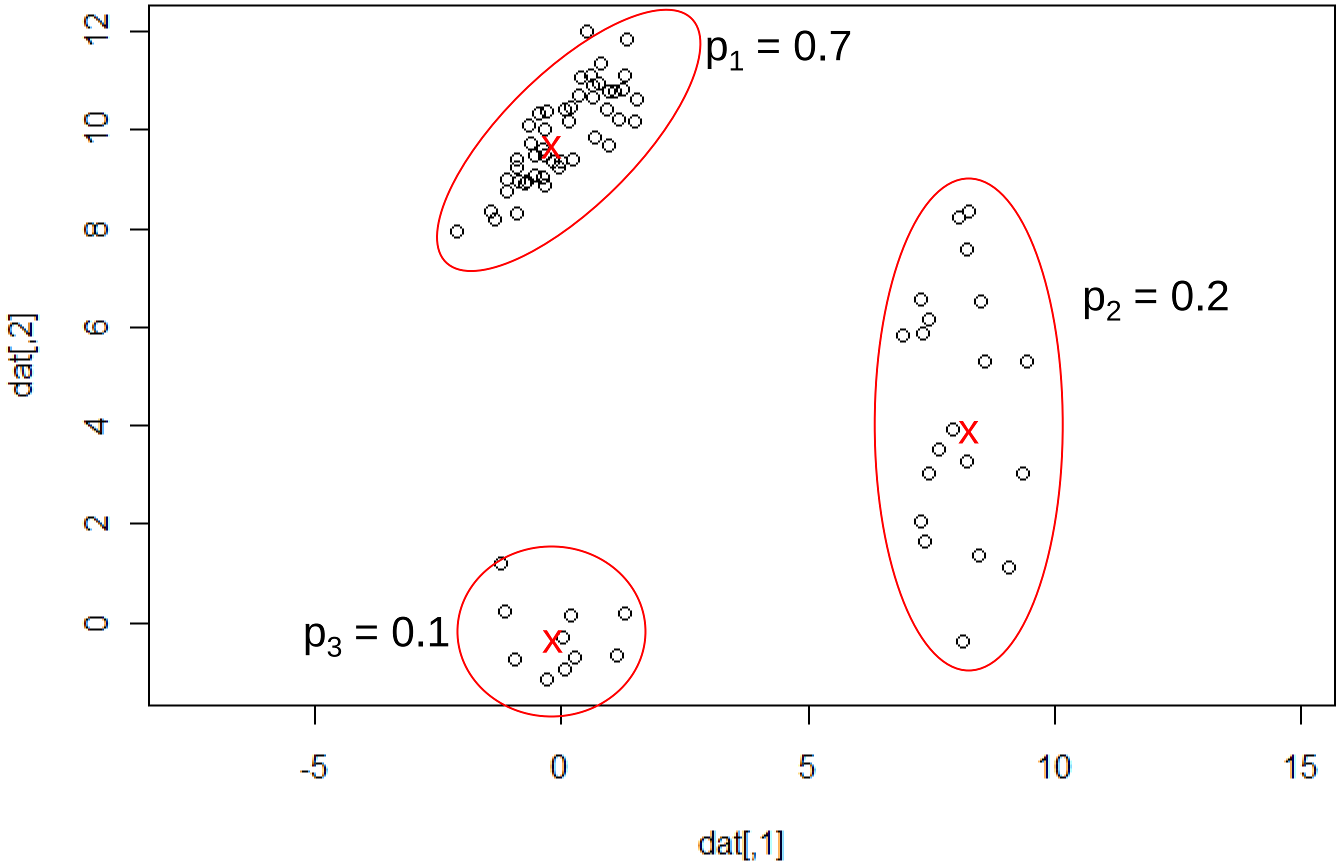
Revision: Multivariate Normal Distribution

$$f(x;\mu,\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\tfrac{1}{2}\cdot(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$



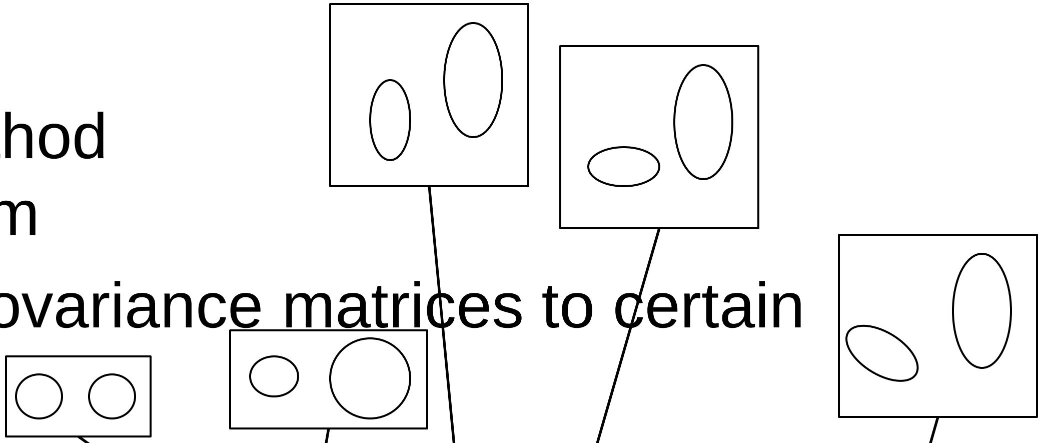
GMM: Example estimated manually

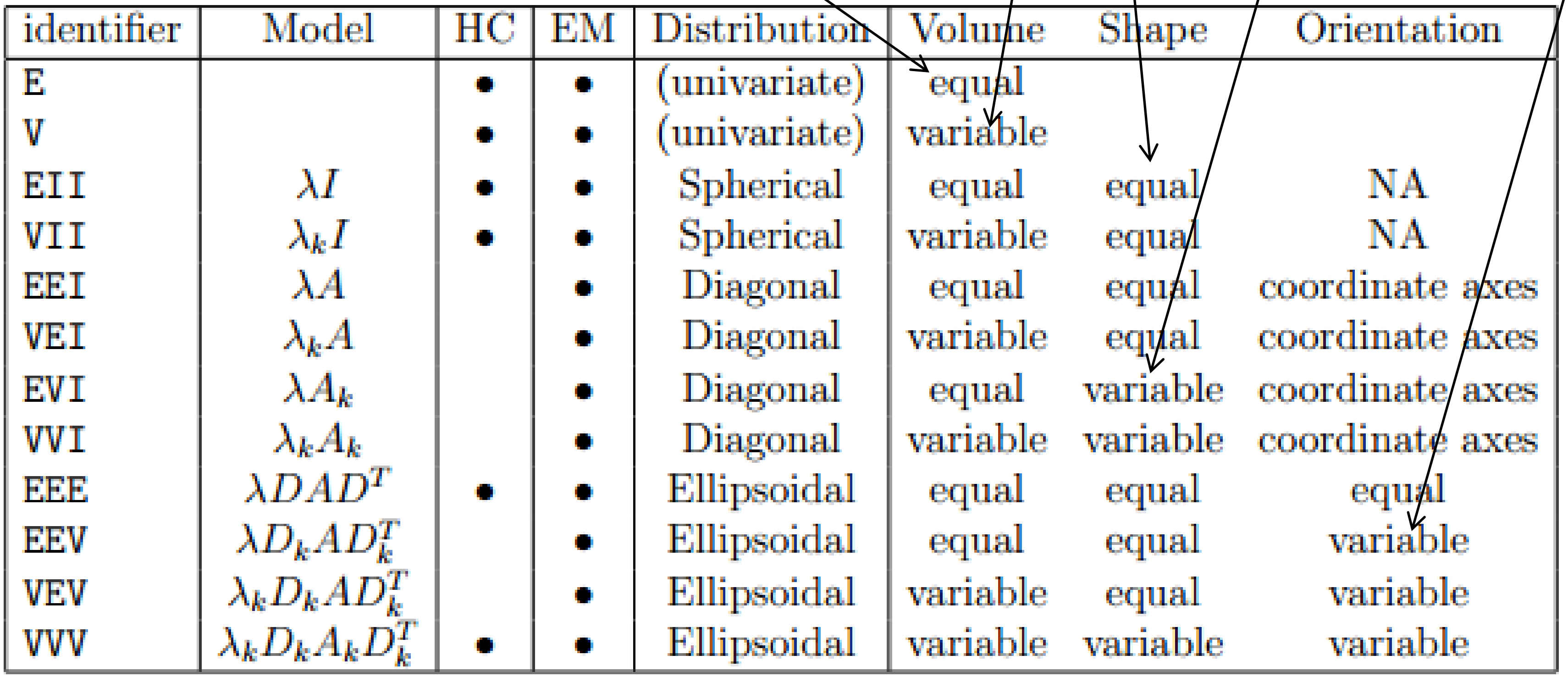
- 3 clusters
- $p_1 = 0.7$, $p_2 = 0.2$, $p_3 = 0.1$
- Mean vector and cov. Matrix per cluster



Fitting GMMs 1/2

- Maximum Likelihood Method
  Hard optimization problem
- Simplification: Restrict Covariance matrices to certain patterns (e.g. diagonal)





identifier	Model	HC	EM	Distribution	Volume	Shape	Orientation
E		●	●	(univariate)	equal		
V		●	●	(univariate)	variable		
EII	λI	●	●	Spherical	equal	equal	NA
VII	$\lambda_k I$	●	●	Spherical	variable	equal	NA
EEI	λA		●	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_k A$		●	Diagonal	variable	equal	coordinate axes
EVI	λA_k		●	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_k A_k$		●	Diagonal	variable	variable	coordinate axes
EEE	$\lambda D A D^T$	●	●	Ellipsoidal	equal	equal	equal
EEV	$\lambda D_k A D_k^T$		●	Ellipsoidal	equal	equal	variable
VEV	$\lambda_k D_k A D_k^T$		●	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k^T$	●	●	Ellipsoidal	variable	variable	variable

Fitting GMMs 2/2

- Problem: Fit will never get worse if you use more cluster or allow more complex covariance matrices
  → How to choose optimal model ?
- Solution: Trade-off between model fit and model complexity

  BIC = log-likelihood – log(n)/2*(number of parameters)

  Find solution with maximal BIC

GMMs in R

- Function “Mclust” in package “mclust”
- Pottery revisited

Giving meaning to clusters

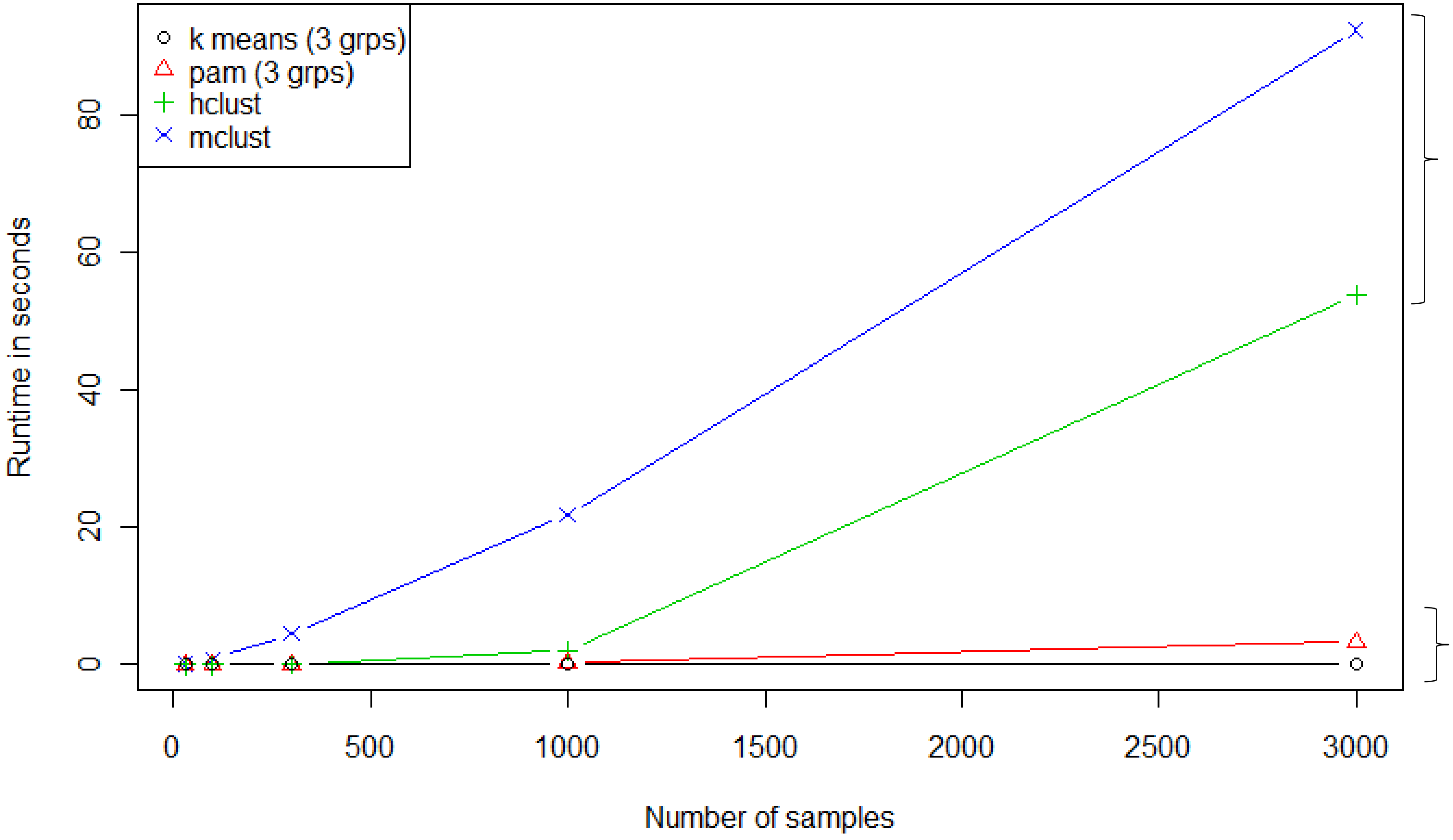
- Generally hard in many dimensions
- Look at position of cluster centers or cluster representatives (esp. easy in PAM)

(Very) small runtime study

Uniformly distributed points in $[0,1]^5$ on my desktop

1 Mio samples with k-means: 5 sec

(always just one replicate; just to give you a rough idea…)



Good for small / medium data sets

Good for huge data sets

Comparing methods

- Partitioning Methods:
  + Super fast (“millions of samples”)
  - No underlying Model
- Agglomerative Methods:
  + Get solutions for all possible numbers of cluster at once
  - slow (“thousands of samples”)
- GMMs:
  + Get statistical model for data generating process
  + Statistically justified selection of number of clusters
  - very slow (“hundreds of samples”)

Concepts to know

- Agglomerative clustering, dendrogram, cutting a dendrogram, dissimilarity measures between cluster
- Partitioning methods: k-Means, PAM
- GMM
- Choosing number of clusters:
  - drop in dendrogram
  - drop in WGSS
  - BIC
- Quality of clustering: Silhouette plot

R functions to know

- Functions “kmeans”, “hclust”, “cutree” in package “stats”
- Functions “pam”, “agnes”, “shilouette” in package “cluster”
- Function “Mclust” in package “mclust”