Machine Learning for Bioinformatics & Systems Biology

2. Classification and clustering

Perry Moerland *Amsterdam UMC, University of Amsterdam*
Marcel Reinders *Delft University of Technology*
Lodewyk Wessels *Netherlands Cancer Institute*

Some material courtesy of Robert Duin and David Tax

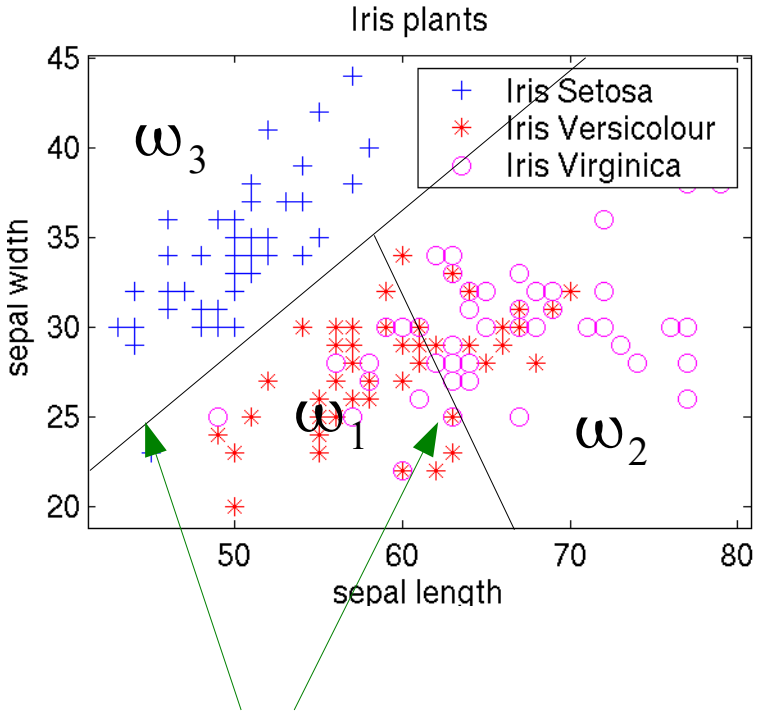
Classification

- Logistic classifier
- Plug-in Bayes classifiers
  - Density-based classification: Parzen, nearest neighbour, Gaussian
- Decision trees and random forests

BioSB

Classification (2)

- Given labeled data: $\boldsymbol{x}$
- Assign to each object a class label ω
- In effect splits the feature space in separate regions



Description of a classifier

There are several ways to describe the classifier:

- If $p(\omega=h|\mathbf{x})>p(\omega=d|\mathbf{x})$ then assign to h otherwise to d
- If $p(\omega=h|\mathbf{x})-p(\omega=d|\mathbf{x})>0$ then assign to h
- If $\frac{p(\omega=h|\mathbf{x})}{p(\omega=d|\mathbf{x})}>1$ then assign to h
- If $\ln(p(\omega=h|\mathbf{x}))-\ln(p(\omega=d|\mathbf{x}))>0$ then assign to h

A Bayesian classifier is a *threshold* on the difference between *posterior probabilities*

BioSB

Logistic classifier

- We can rewrite:

$$\ln\left(p(\omega=h|\boldsymbol{x})\right)-\ln\left(p(\omega=d|\boldsymbol{x})\right)=\ln\left(\frac{p(\omega=h|\boldsymbol{x})}{p(\omega=d|\boldsymbol{x})}\right)$$

logit, log-odds

- Assume we can approximate:

$$\ln\left(\frac{p(\omega=h|\boldsymbol{x})}{p(\omega=d|\boldsymbol{x})}\right)=w_0+\boldsymbol{w}^T\boldsymbol{x}$$

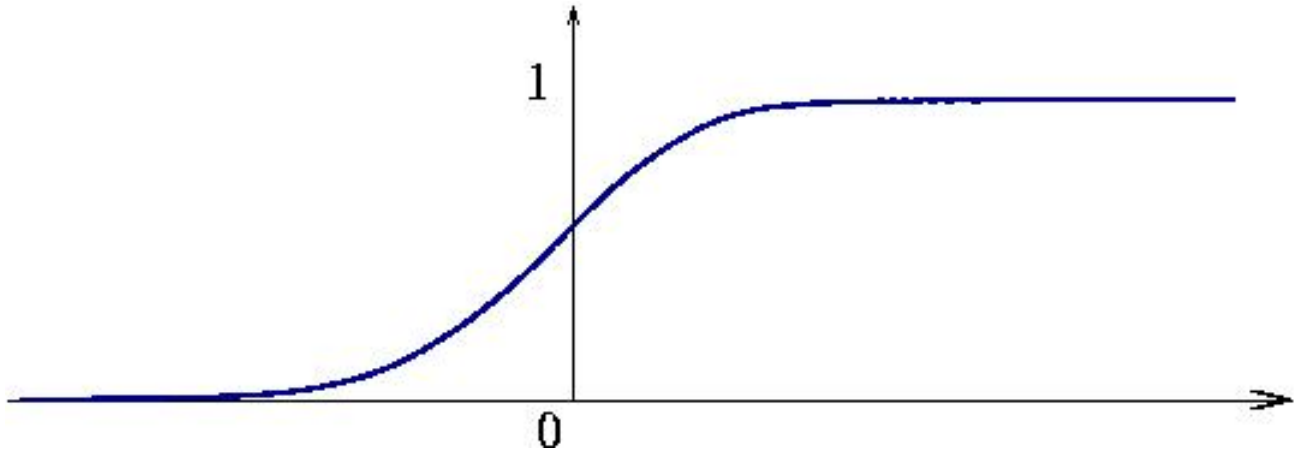
- The classifier becomes (computer lab exercise):

$$p(\omega=d|\boldsymbol{x})=\frac{1}{1+\exp(\boldsymbol{w}^T\boldsymbol{x}+w_0)}$$

BioSB

Logistic function

- The function looks like:



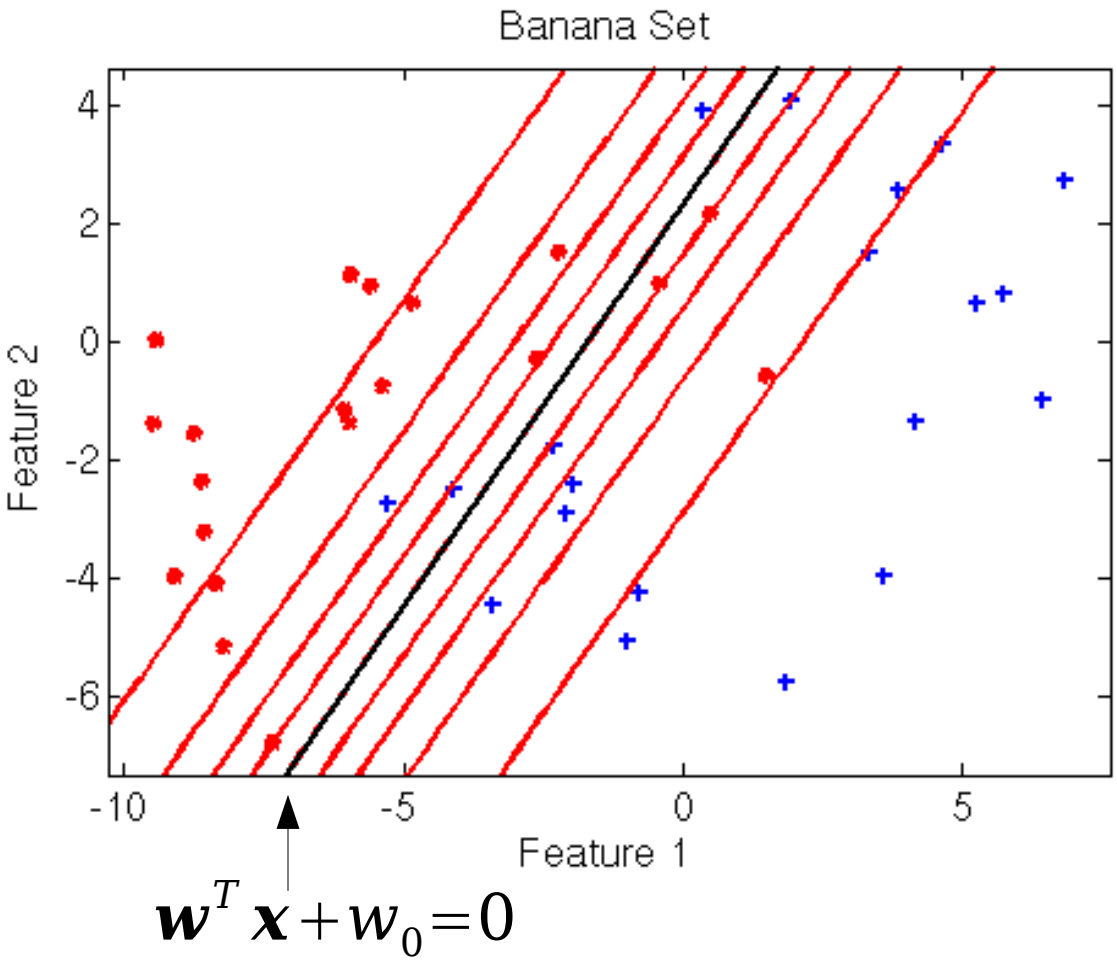
$$f(x)=\frac{1}{1+\exp(-x)}$$

logistic (sigmoid) function

BioSB

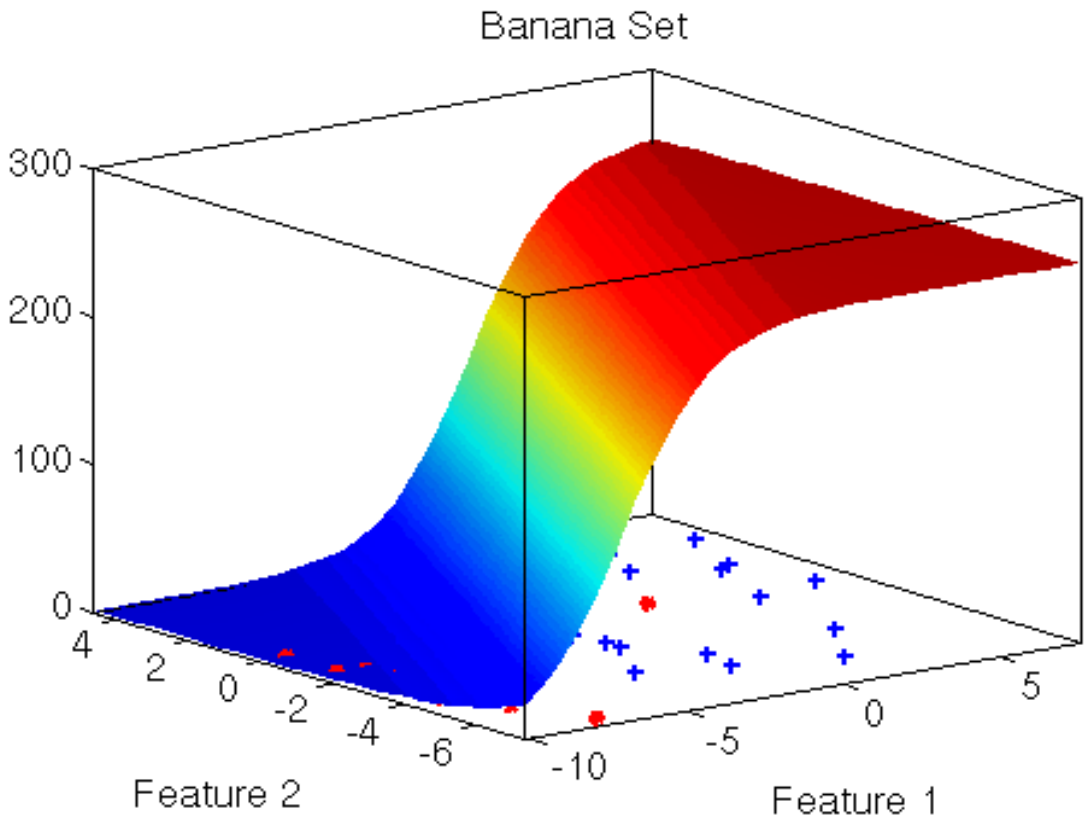
Logistic classifier (2)

- On a two-dimensional dataset it looks like:



Logistic classifier (3)

- On a two-dimensional dataset it looks like:



Optimizing the logistic classifier

- To optimize the parameters on a training set, maximize the likelihood

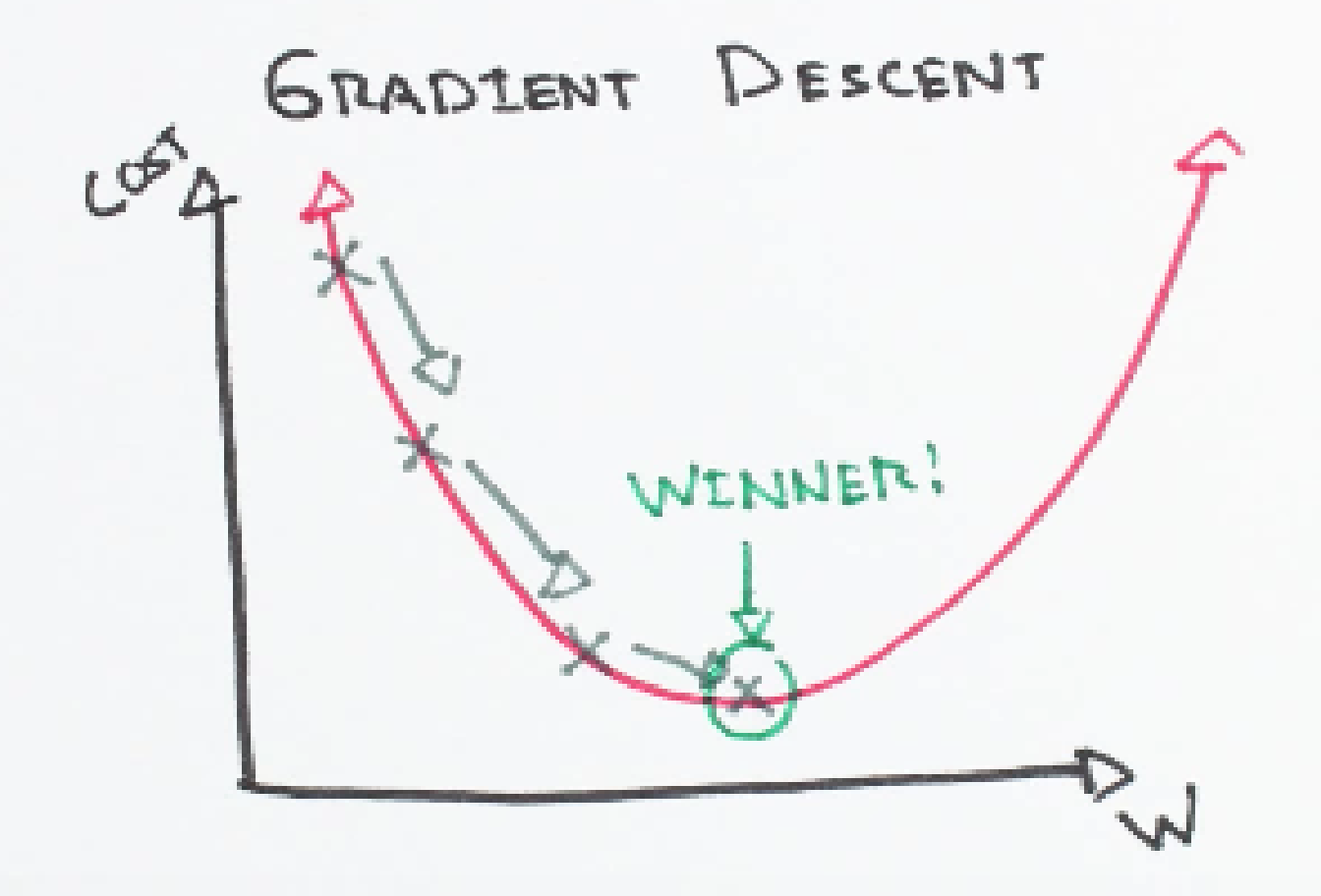
$$L = \prod_{i=1}^{n_1} p(\mathbf{x}_i^{(1)} | \omega_1) \prod_{j=1}^{n_2} p(\mathbf{x}_j^{(2)} | \omega_2)$$

where $\mathbf{x}_i^{(j)}$ is the i-th object from class j

- Maximization using gradient ascent
- Appears to be easier to maximize log(L)
- Weights are iteratively updated as:

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \eta \frac{\partial \log(L)}{\partial \mathbf{w}}$$

BioSB


GRADIENT DESCENT
LOSS
WINNER!
W

Optimizing the logistic classifier (2)

- Function to maximize

$$L = \prod_{i=1}^{n_1} p(\boldsymbol{x}_i^{(1)}|\omega_1) \prod_{j=1}^{n_2} p(\boldsymbol{x}_j^{(2)}|\omega_2)$$

- Use log(L)

$$\log(L) = \sum_{i=1}^{n_1} \log(p(\boldsymbol{x}_i^{(1)}|\omega_1)) + \sum_{j=1}^{n_2} \log(p(\boldsymbol{x}_j^{(2)}|\omega_2))$$

- Use Bayes' theorem

$$\log p(\boldsymbol{x}_i^{(1)}|\omega_1) = \log p(\omega_1|\boldsymbol{x}_i^{(1)}) - \log p(\omega_1) + \log p(\boldsymbol{x}_i^{(1)})$$

constant

- Therefore

$$\log(L) = \sum_{i=1}^{n_1} \log(p(\omega_1|\boldsymbol{x}_i^{(1)})) + \sum_{j=1}^{n_2} \log(p(\omega_2|\boldsymbol{x}_j^{(2)})) + C$$

BioSB

Optimizing the logistic classifier (3)

- Filling in that

$$p(\omega_2|\boldsymbol{x}) = \frac{1}{1+\exp(\boldsymbol{w}^T\boldsymbol{x}+w_0)}$$

gives

$$\log(L) = \sum_{i=1}^{n_1}(w_0+\boldsymbol{w}^T\boldsymbol{x}_i^{(1)}) - \sum_{j=1}^{n_1+n_2}\log(1+\exp(w_0+\boldsymbol{w}^T\boldsymbol{x}_j))$$

BioSB

Derivative of the log-likelihood

- The gradient of log(L) is

$$\frac{\partial \log(L)}{\partial w_0} = n_1 - \sum_{i=1}^{n_1+n_2} p(\omega_1|\boldsymbol{x}_i)$$

$$\frac{\partial \log(L)}{\partial w_j} = \sum_{i=1}^{n_1} (\boldsymbol{x}_i^{(1)})_j - \sum_{i=1}^{n_1+n_2} p(\omega_1|\boldsymbol{x}_i)(\boldsymbol{x}_i)_j, \; j=1,\ldots,p$$

- Take initial values: $w_0 = 0, \boldsymbol{w} = \boldsymbol{0}$
- Keep iterating $\boldsymbol{w}_{new} = \boldsymbol{w}_{old} + \eta \frac{\partial \log(L)}{\partial \boldsymbol{w}}$

till convergence

BioSB

Plug-in Bayes classification

- In many cases the posterior is hard to estimate
- Often a functional form of the class distributions can be assumed
- Use Bayes' theorem to rewrite one into the other:

$$p(\omega|\mathbf{x}) = \frac{p(\mathbf{x}|\omega)\,p(\omega)}{p(\mathbf{x})}$$

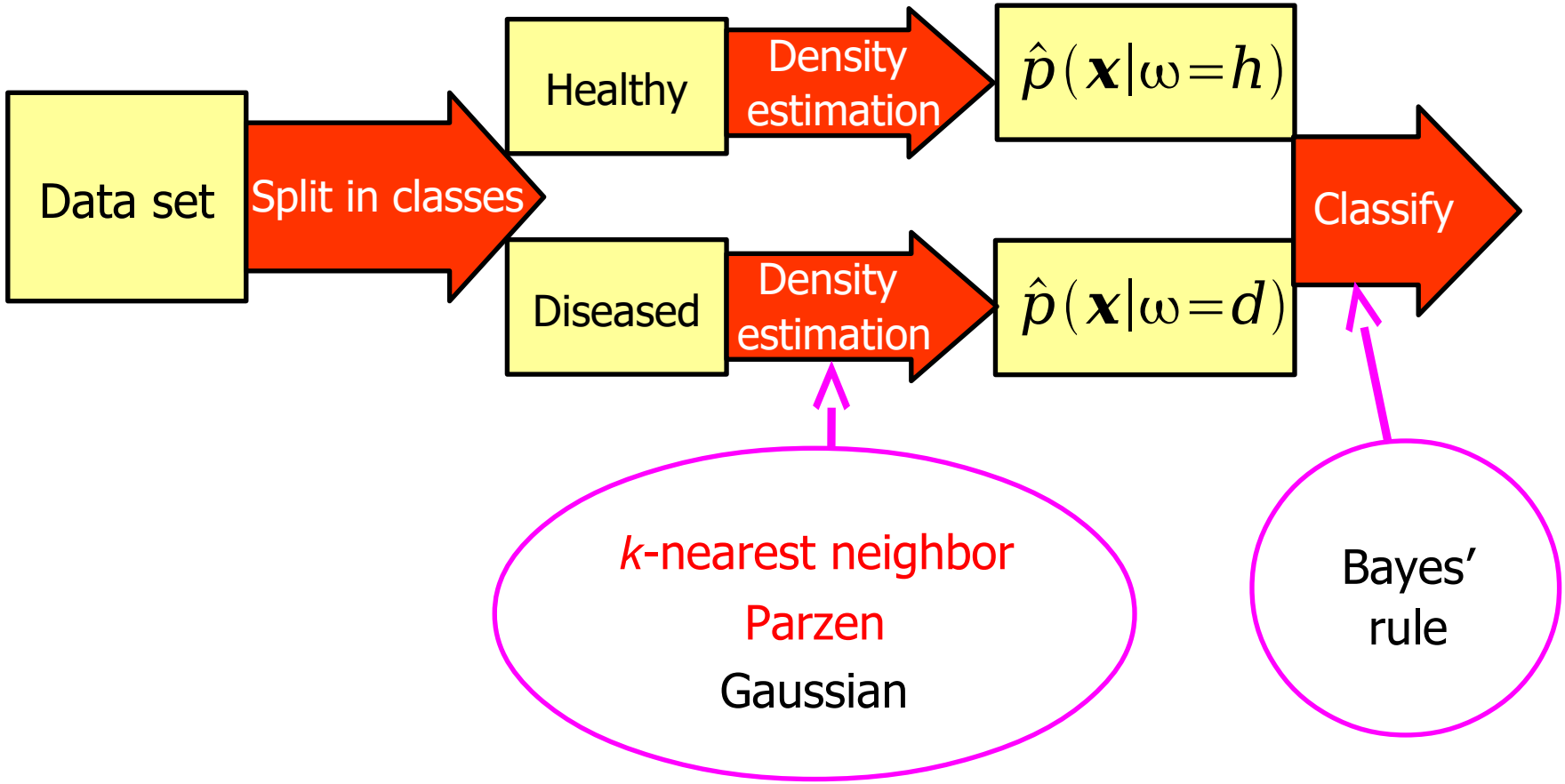
class-conditional distribution: $p(\mathbf{x}|\omega)$

prior distribution: $p(\omega)$

data distribution: $p(\mathbf{x})$

BioSB

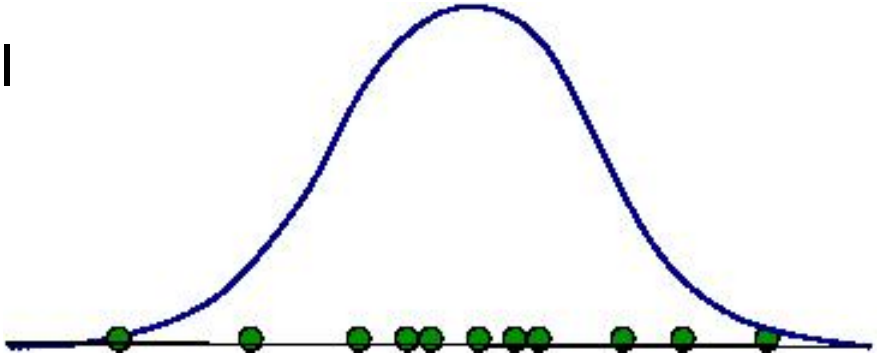
Plug-in Bayes classification (2)



Bayes decision making

$$\hat{p}(\boldsymbol{x}|\omega_i) = N(\boldsymbol{x}; \mu, \sigma)$$

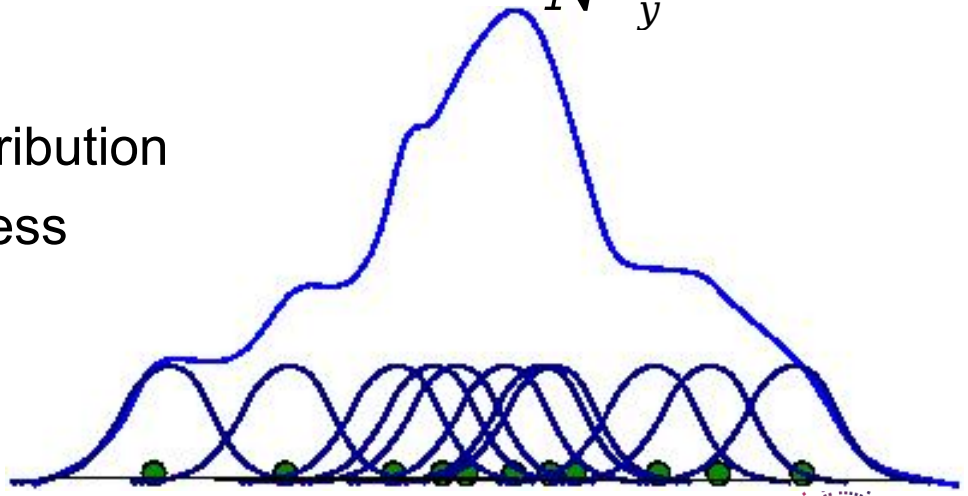
- Estimate the class-conditional density (Day 1) $\hat{p}(\boldsymbol{x}|\omega_i)$



- Parametric
  - Known distribution
  - Estimate parameters on training set

$$\hat{p}(\boldsymbol{x}|\omega_i) = \frac{1}{N}\sum_{y} K(\boldsymbol{x}, \boldsymbol{y})$$

- Non-parametric
  - No knowledge on distribution
  - Manage the smoothness of the distribution

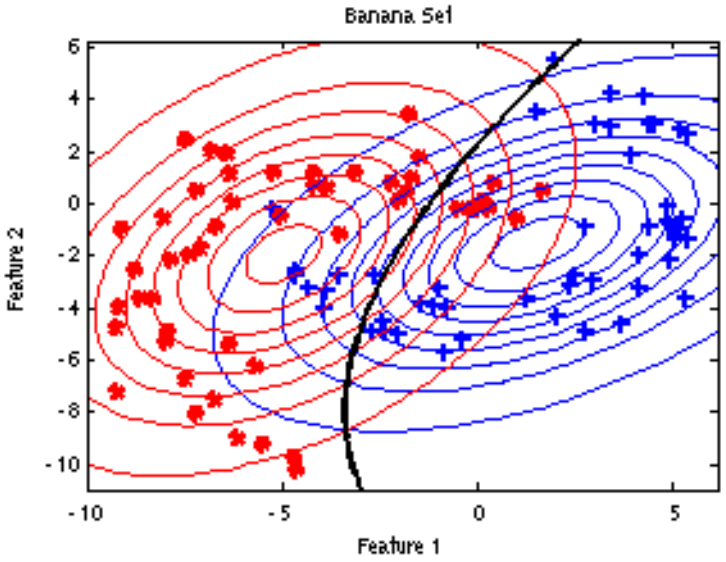


BioSB

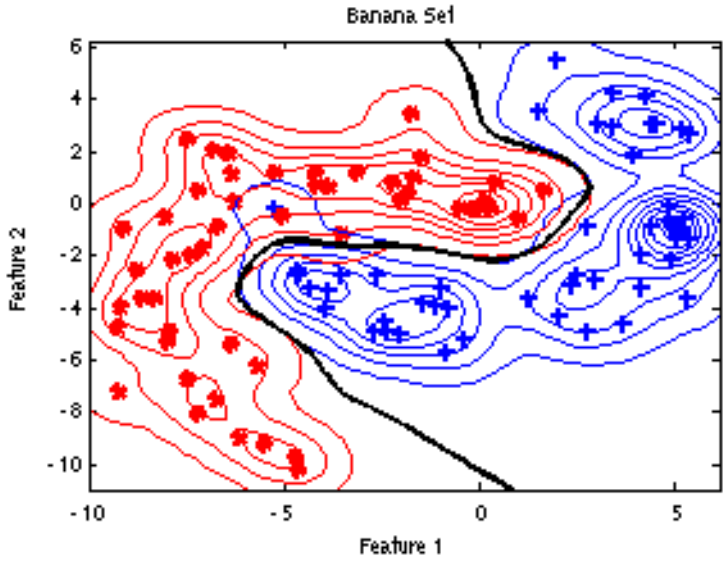
Example plugin

- Two examples

Normal density estimation



Parzen density estimation

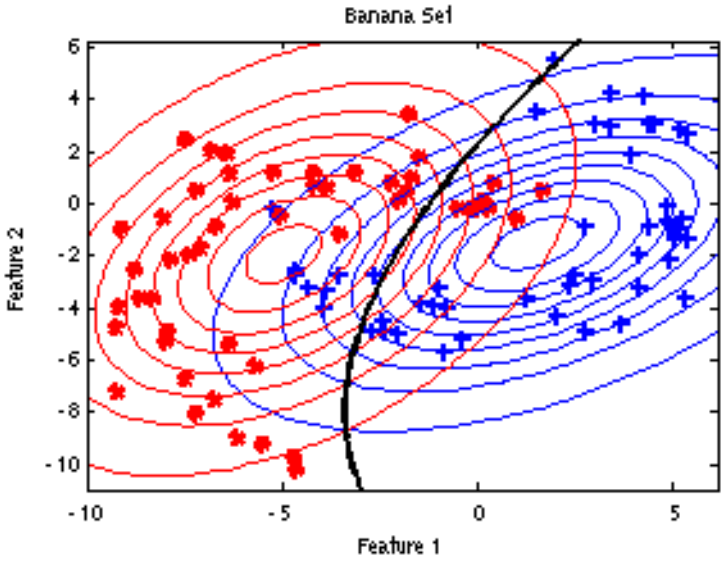


BioSB

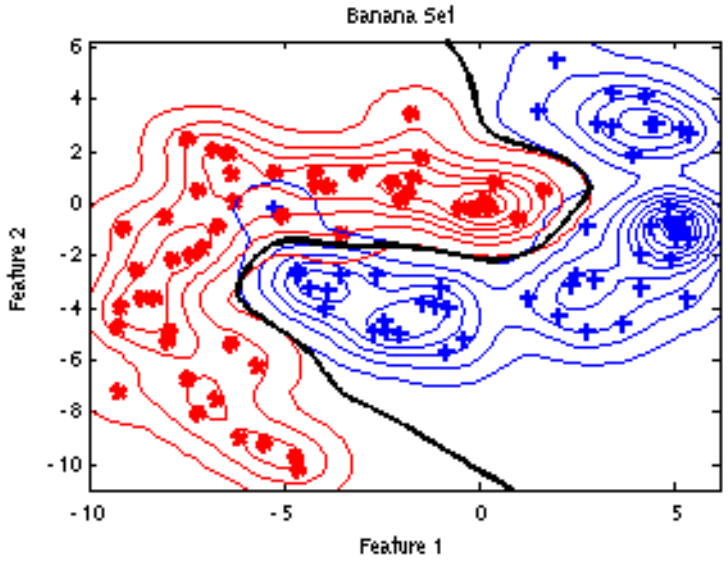
Example plugin

- Two examples

Normal density estimation



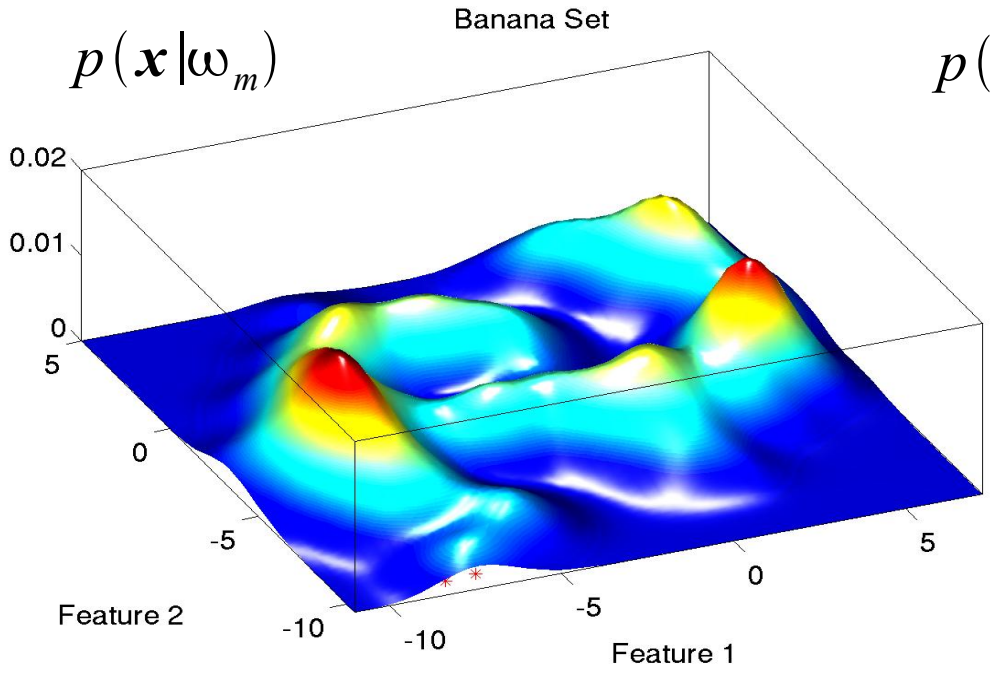
Parzen density estimation

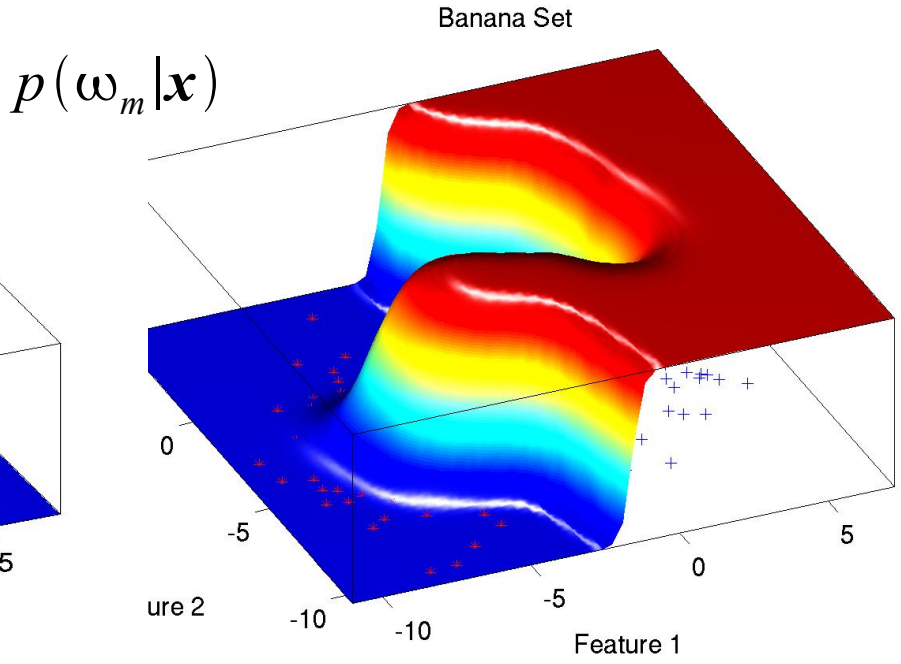


BioSB

Parzen classifier

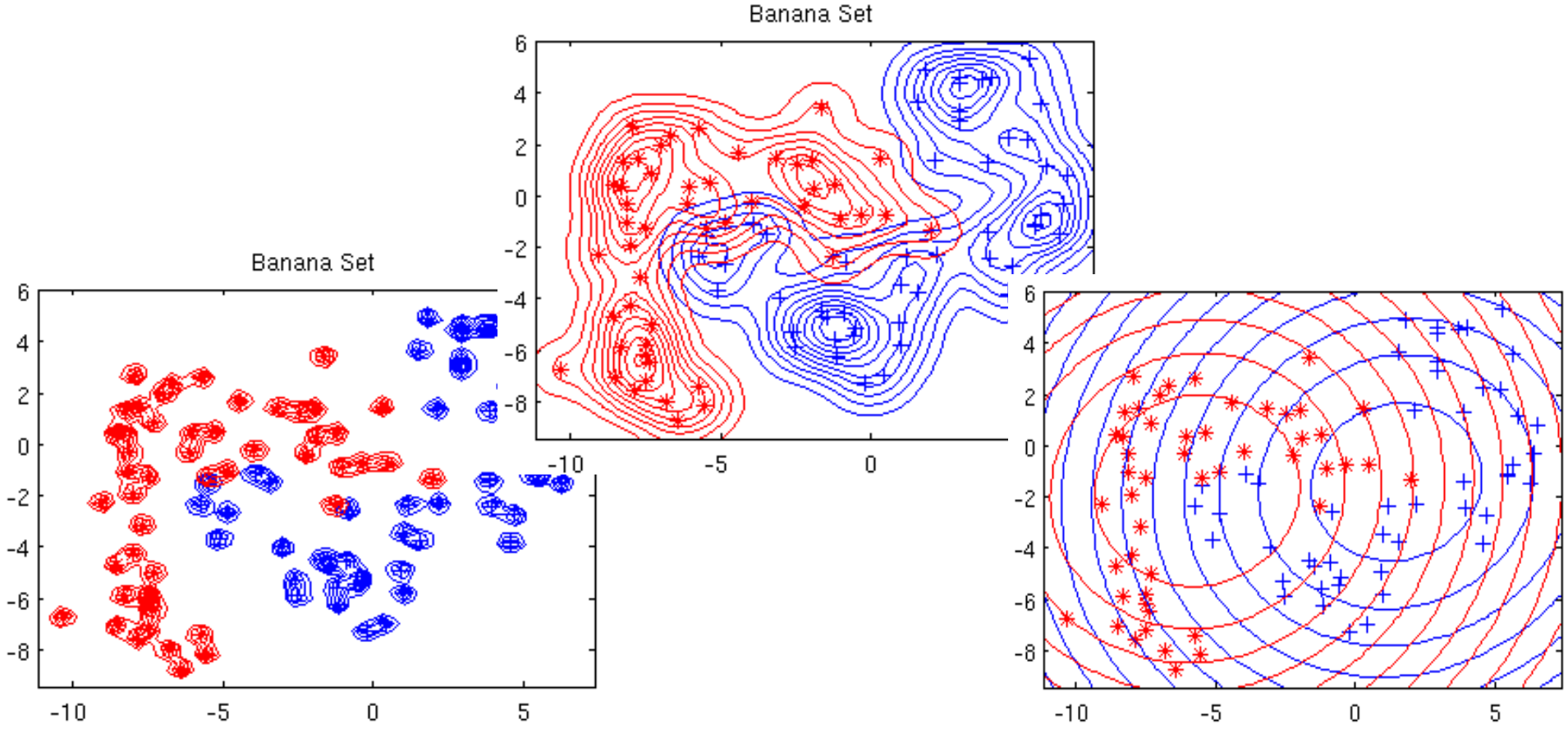
$$p(\boldsymbol{x}|\omega_m) = \frac{1}{N}\sum_{i=1}^{N_m} N(\boldsymbol{x}; \boldsymbol{x}_i, h\boldsymbol{I})$$





Parzen width parameter

- The width parameter *h* has a large influence



Optimization of *h*

- Use the average *k*-nearest neighbor distance
  (*k*=10 is suggested...)
- Use a heuristic $$h = \sigma \left(\frac{4}{p+2} \right)^{\frac{1}{p+4}} n^{\frac{-1}{p+4}}$$

$$\sigma^2 = \frac{1}{p} \sum_{i=1}^{p} s_{ii}$$

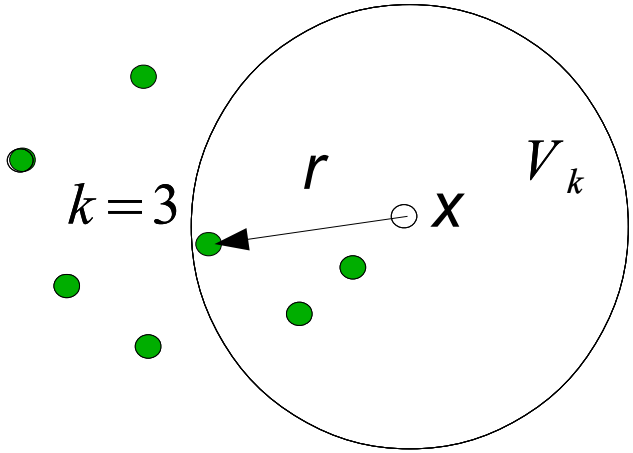
- Optimize the likelihood using cross-validation

$$\prod_{i=1}^{n} \hat{p}(\boldsymbol{x}_i)$$

- and more...

BioSB

Nearest neighbor classification

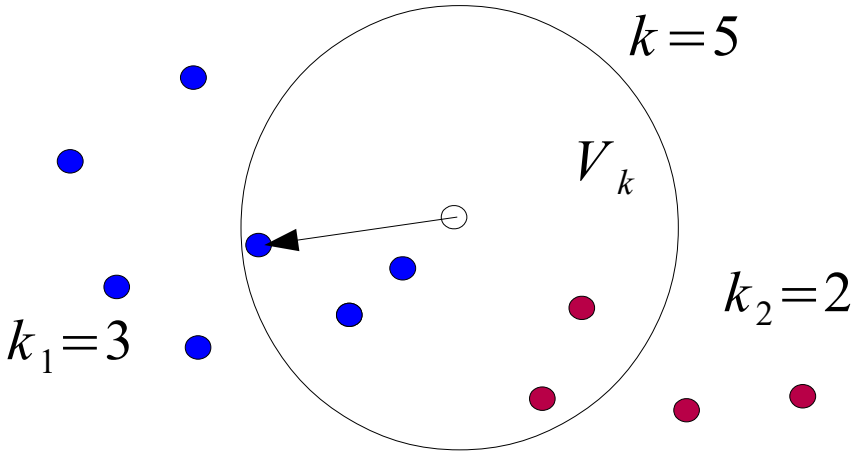


- For the k-nearest neighbor density we defined:

$$\hat{p}(\boldsymbol{x}) = \frac{k}{n V_k}$$

where V_k is the volume of the sphere centered at x, with radius r the distance to the k-th nearest neighbor

Nearest neighbor classification (2)



- When more classes are present, count how many objects of each of the classes are members of the *k* neighbors
- Class-conditional density:

$$\hat{p}(\boldsymbol{x}|\omega_m)=\frac{k_m}{n_m V_k}$$

Nearest neighbor classification (3)

- Using Bayes: $\hat{p}(\boldsymbol{x}|\omega_m)\,\hat{p}(\omega_m) \geq \hat{p}(\boldsymbol{x}|\omega_i)\,\hat{p}(\omega_i)$
- Estimate the prior probability by counting:

$$\hat{p}(\omega_m) = \frac{n_m}{n}$$

- Fill in:

$$\frac{k_m}{n_m V_k}\frac{n_m}{n} \geq \frac{k_i}{n_i V_k}\frac{n_i}{n} \quad \Rightarrow \quad k_m \geq k_i$$

- No density estimation is needed!

BioSB

The choice of *k*

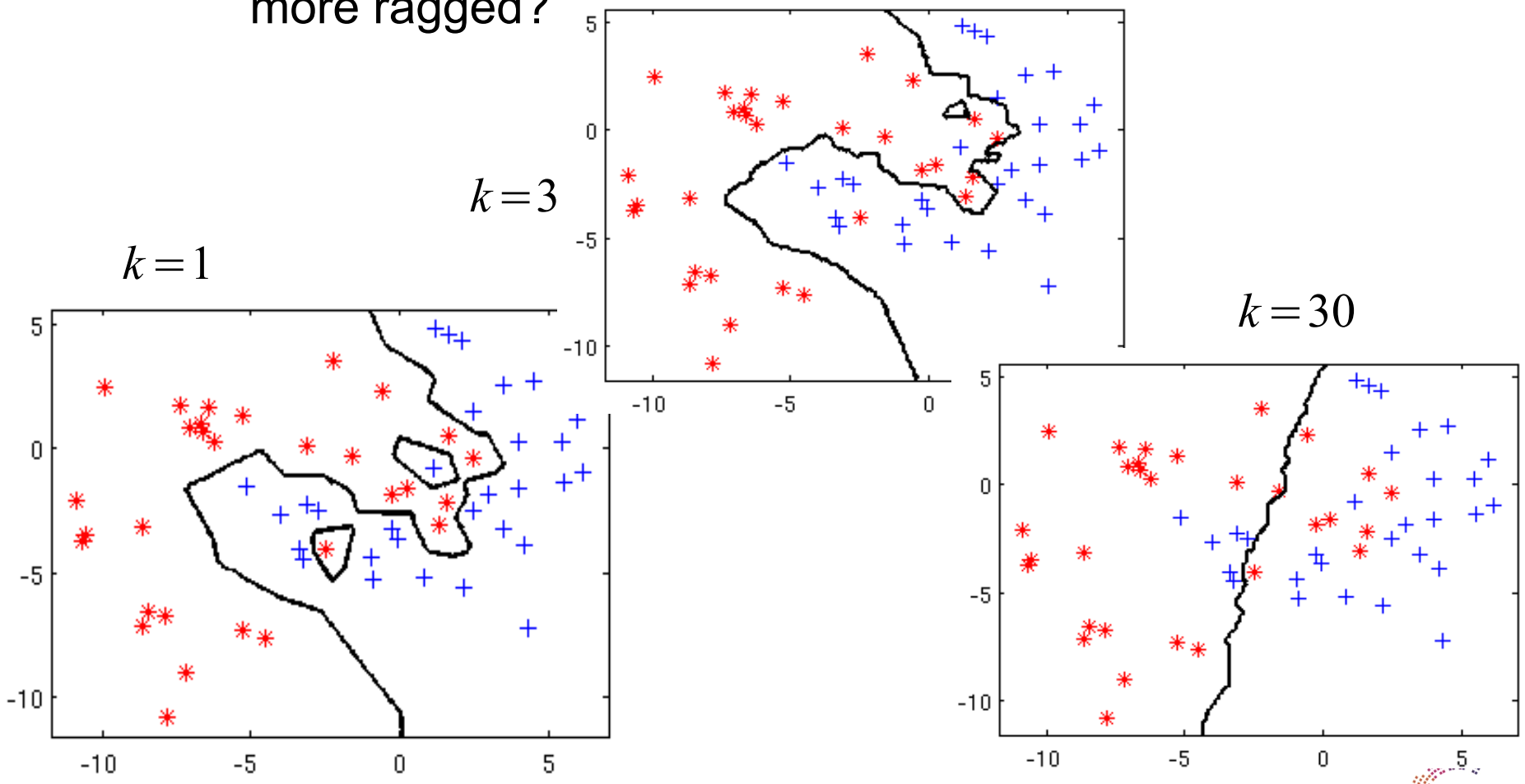
- When does the classifier become more smooth? When more ragged?

- What happens for $k = 1$, and $k = n$?

BioSB

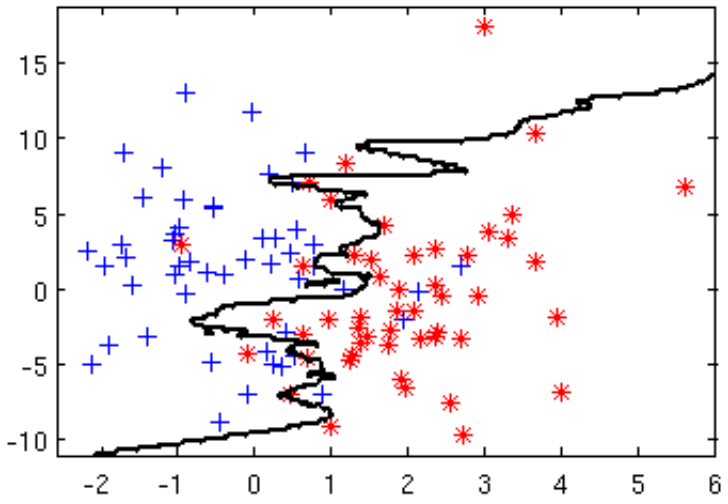
The choice of *k* (2)

- When does the classifier become more smooth? When more ragged?



Sometimes strange results:

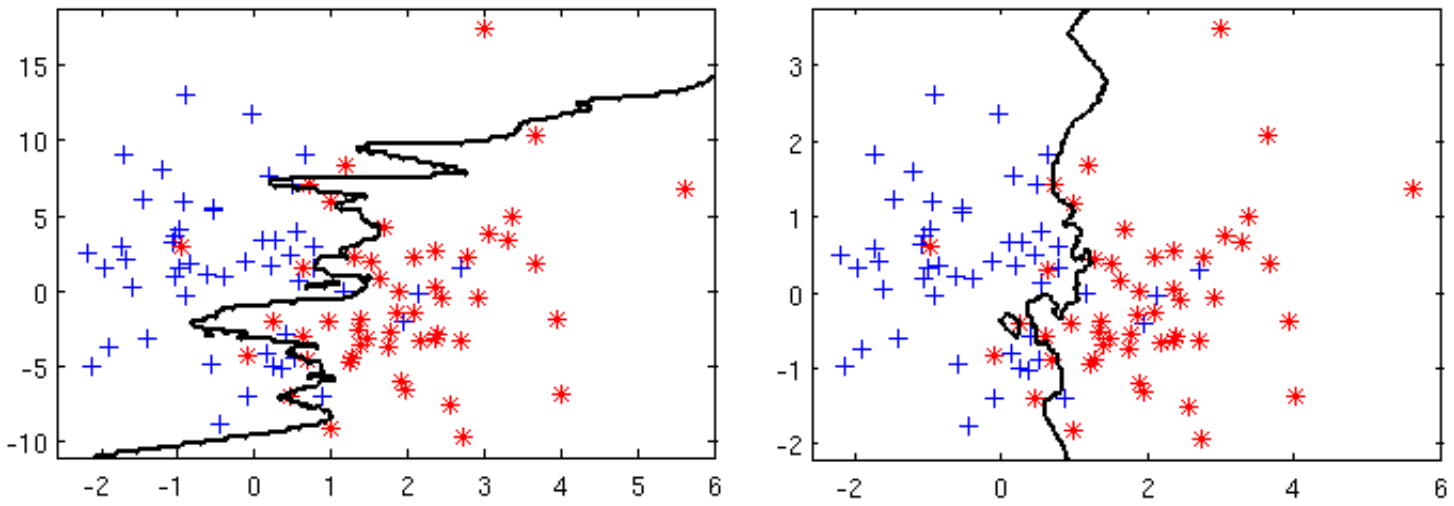
$k = 5$



BioSB

Sometimes strange results (2):

$k = 5$



Rescaling the features has large influence!

BioSB

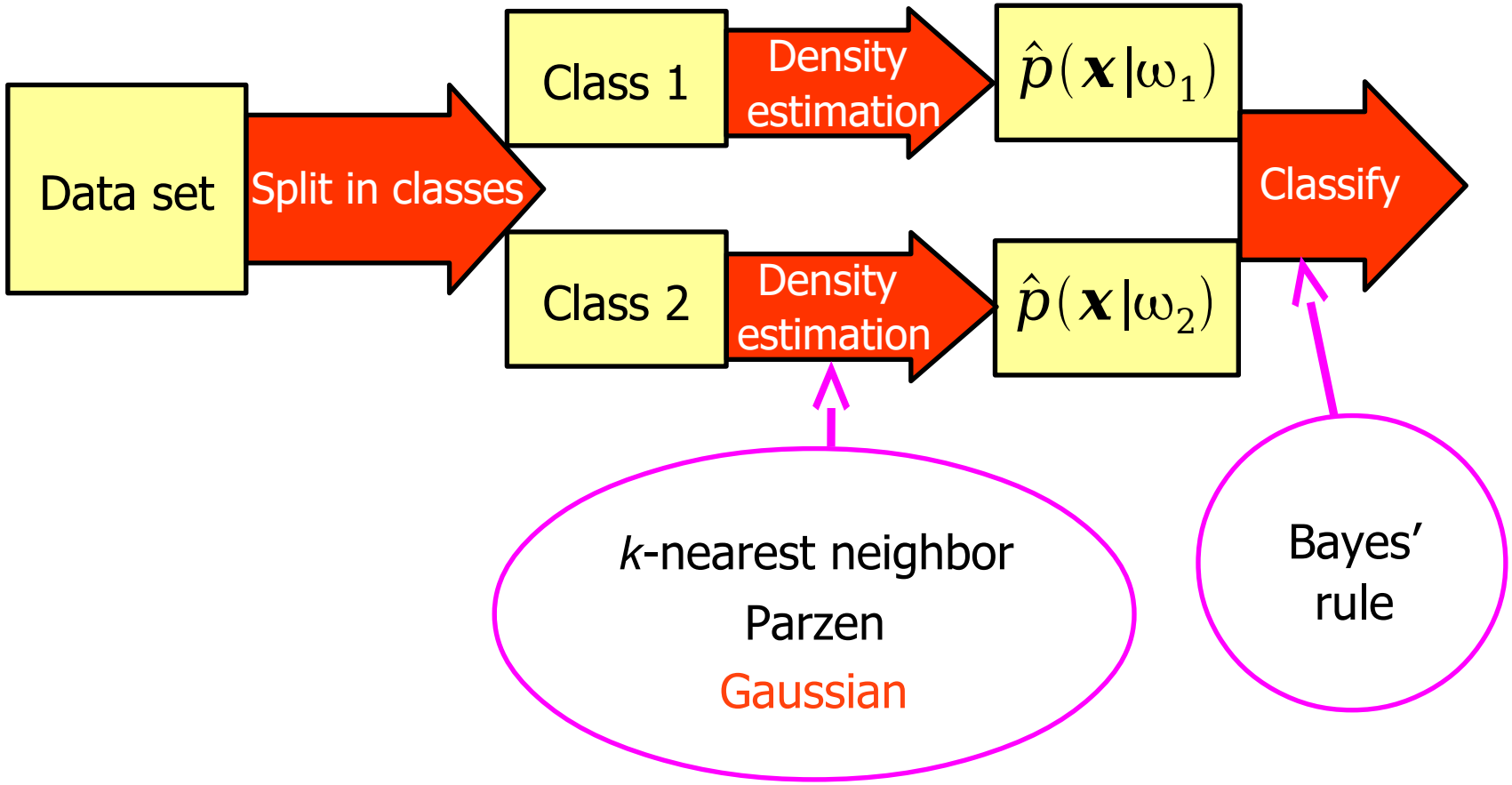
Advantages/disadvantages

- simple and flexible classifier
- often a very good classification performance
- it is simple to adapt the complexity of the classifier

- you have to store the complete training set
- distances to all training objects have to be computed
- scaling of the features should be sensible
- you have to optimize *k* or *h*

BioSB

Classifying with densities



BioSB

Plug-in Gaussian distribution

- Now take the most obvious choice: the Gaussian distribution

$$\hat{p}(\mathbf{x}|\omega) = \frac{1}{\sqrt{2\pi^p det(\hat{\Sigma}_\omega)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\hat{\mu}_\omega)^T \hat{\Sigma}_\omega^{-1}(\mathbf{x}-\hat{\mu}_\omega)\right)$$

- So-called parametric density estimation
- We have to estimate the parameters via maximum likelihood:

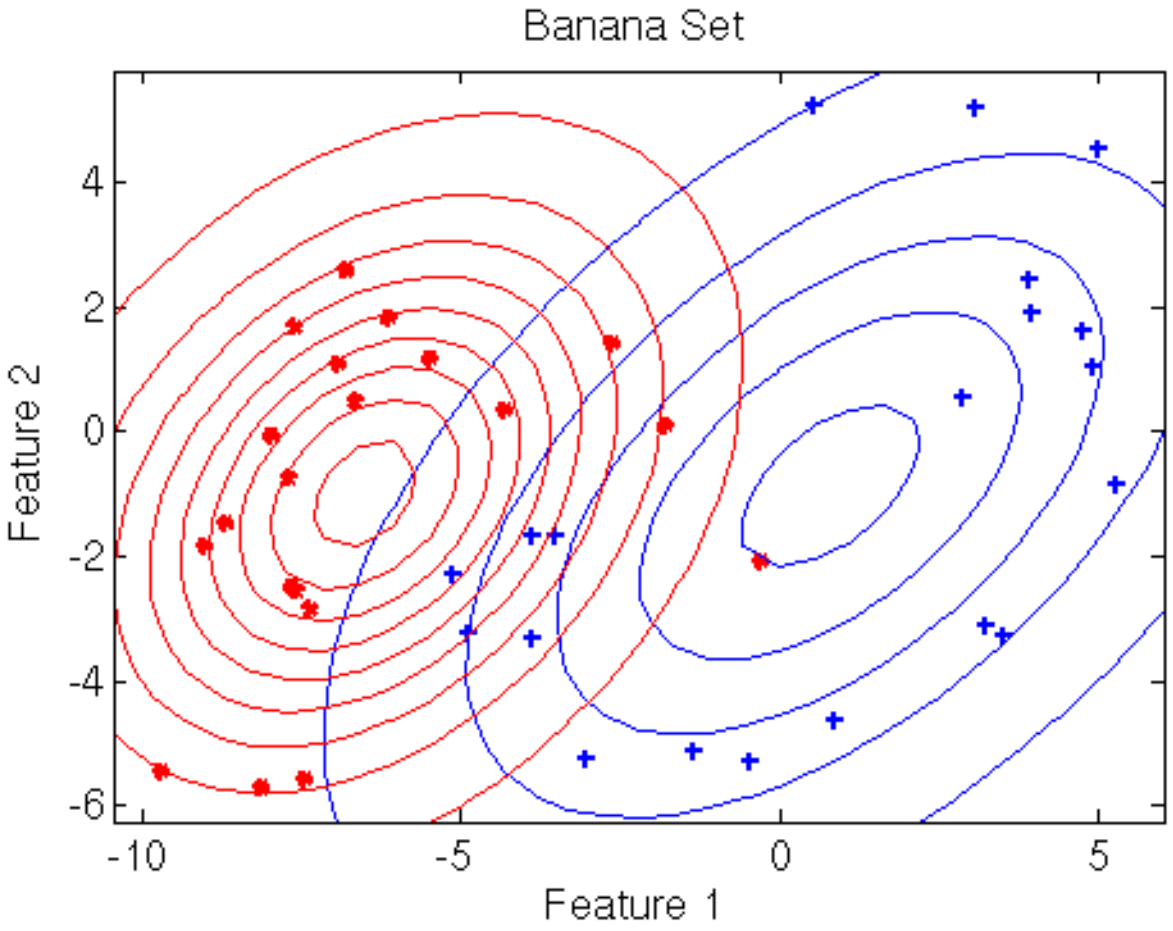
$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^T$$

BioSB

Example on banana data

- A single Gaussian distribution on each class:



BioSB

Class-conditional densities

- Combining

$$\hat{p}(\mathbf{x}|\omega_i) = \frac{1}{\sqrt{2\pi^p det(\Sigma_i)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)\right)$$

$$p(\omega|\mathbf{x}) = \frac{p(\mathbf{x}|\omega)\,p(\omega)}{p(\mathbf{x})}$$

we can derive for log(p):

$$\log(\hat{p}(\omega_i|\mathbf{x})) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log(det\,\Sigma_i)$$

$$-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i) + \log(p(\omega_i)) - \log(p(\mathbf{x}))$$

BioSB

Normal density-based classifier

- $p(\mathbf{x})$ is independent of the classes and can be dropped

$$g_i(\mathbf{x}) = -\frac{1}{2}\log(det\,\Sigma_i) - \frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i) + \log(p(\omega_i))$$

- Classifier becomes:

assign $\mathbf{x}$ to class ω_i when for all $i \neq j$: $g_i(\mathbf{x}) > g_j(\mathbf{x})$

BioSB

The two-class case

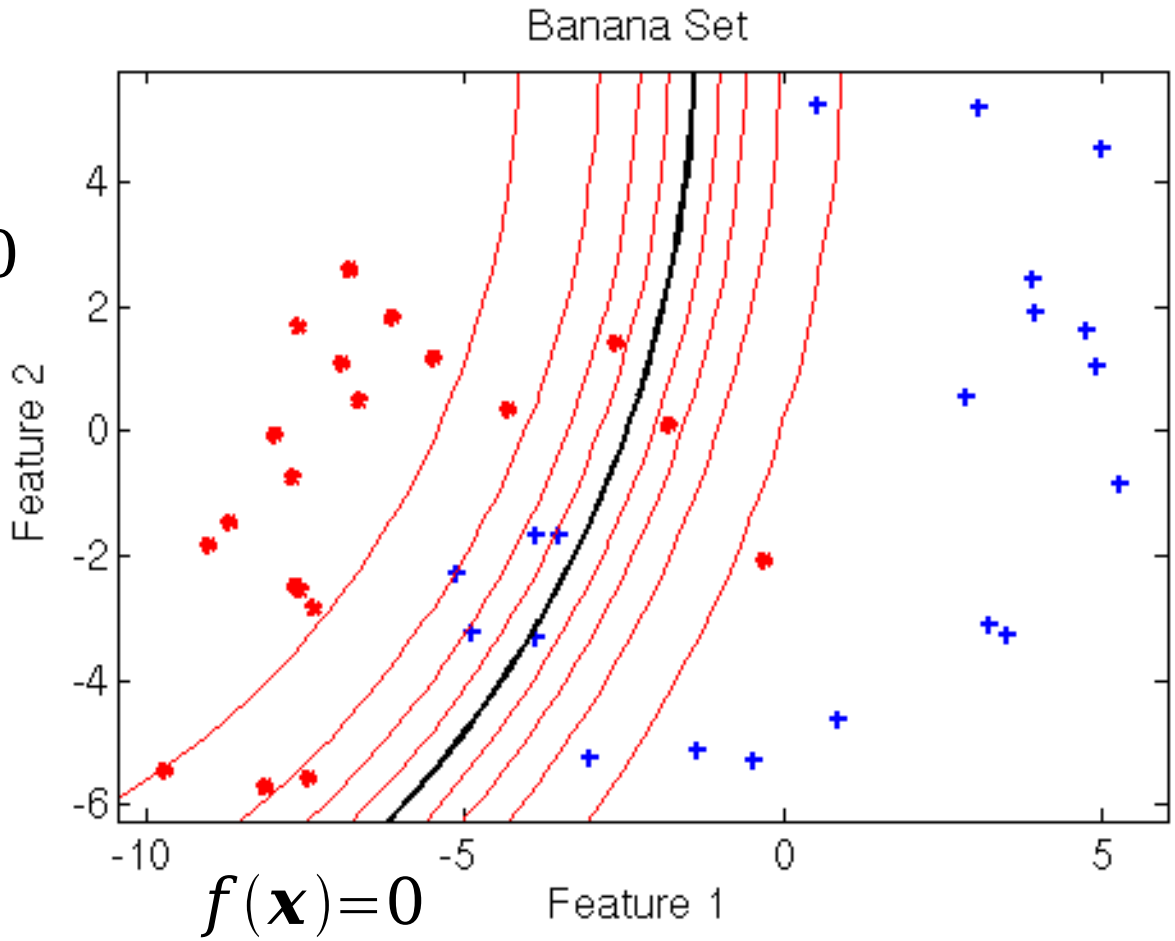
- Define the discriminant $f(\boldsymbol{x}) = p(\omega_1|\boldsymbol{x}) - p(\omega_2|\boldsymbol{x}) > 0$
- We get (computer lab exercise):

$$f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x} + \boldsymbol{w}^T \boldsymbol{x} + w_0$$

- This is a quadratic classifier because
  the decision boundary is a quadratic function of x

Quadratic classifier on banana data

$f(\mathbf{x})>0$



$f(\mathbf{x})<0$

$f(\mathbf{x})=0$

Estimating the covariance matrix

- For the quadratic classifier you need to estimate

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{k}} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\mu}_k)(\boldsymbol{x}_i - \hat{\mu}_k)^T$$

  for each of the classes!

- When you have insufficient data, this covariance matrix cannot be inverted
- Average over the covariance matrices of different classes:

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{c} \sum_{k=1}^{c} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}$$

BioSB

Average covariance matrix

- When we use the averaged covariance matrix:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\log(det\,\hat{\Sigma}) - \frac{1}{2}(\boldsymbol{x}-\hat{\mu}_i)^T\hat{\Sigma}^{-1}(\boldsymbol{x}-\hat{\mu}_i) + \log(p(\omega_i))$$

- First term and quadratic term are always the same for all classes
- We end up with:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\hat{\mu}_i^T\hat{\Sigma}^{-1}\hat{\mu}_i - \frac{1}{2}\hat{\mu}_i^T\hat{\Sigma}^{-1}\boldsymbol{x} + \log(p(\omega_i))$$

- This classifier is *linear*:
  the linear normal density-based classifier.

BioSB

The two-class case (2)

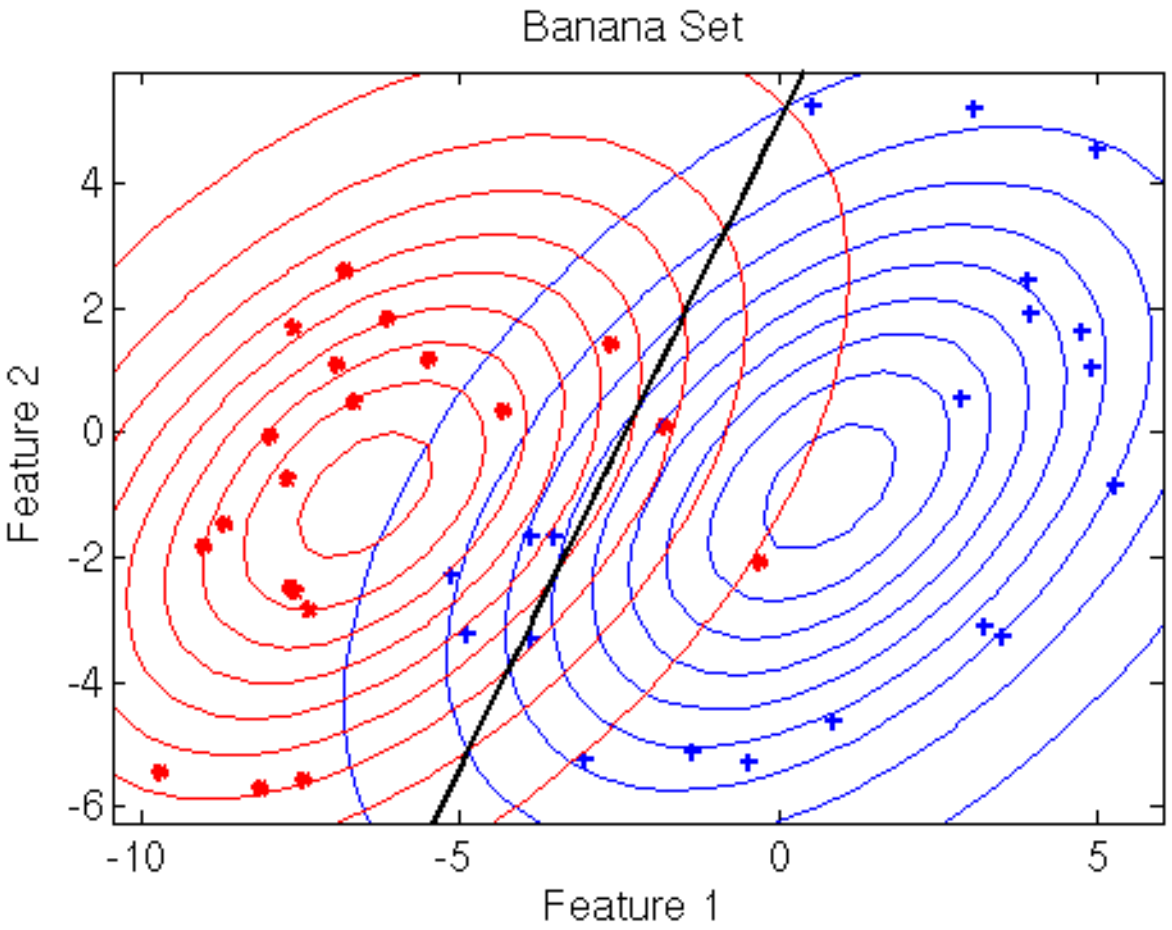
- Define the discriminant $f(\boldsymbol{x}) = p(\omega_1|\boldsymbol{x}) - p(\omega_2|\boldsymbol{x}) > 0$
- We get $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$

with $\boldsymbol{w} = \hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_2)$

$$w_0 = \frac{1}{2}\hat{\mu}_2^T \hat{\Sigma}^{-1} \hat{\mu}_2 - \frac{1}{2}\hat{\mu}_1^T \hat{\Sigma}^{-1} \hat{\mu}_1 + \log \frac{p(\omega_1)}{p(\omega_2)}$$

BioSB

Linear classifier on banana data



No estimated full covariance matrix

- In some cases even the averaged covariance matrix is too much to estimate
- Assume that all features have the same variance, and are uncorrelated:

$$\hat{\Sigma} = \sigma^2 I$$

- Then it becomes even simpler:

$$g_i(\mathbf{x}) = -\frac{1}{2\hat{\sigma}^2}(\hat{\mu}_i^T \hat{\mu}_i - \hat{\mu}_i^T \mathbf{x}) + \log(p(\omega_i))$$

Nearest mean classifier

- Define the discriminant: $f(\boldsymbol{x}) = p(\omega_1|\boldsymbol{x}) - p(\omega_2|\boldsymbol{x}) > 0$
- We get $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$

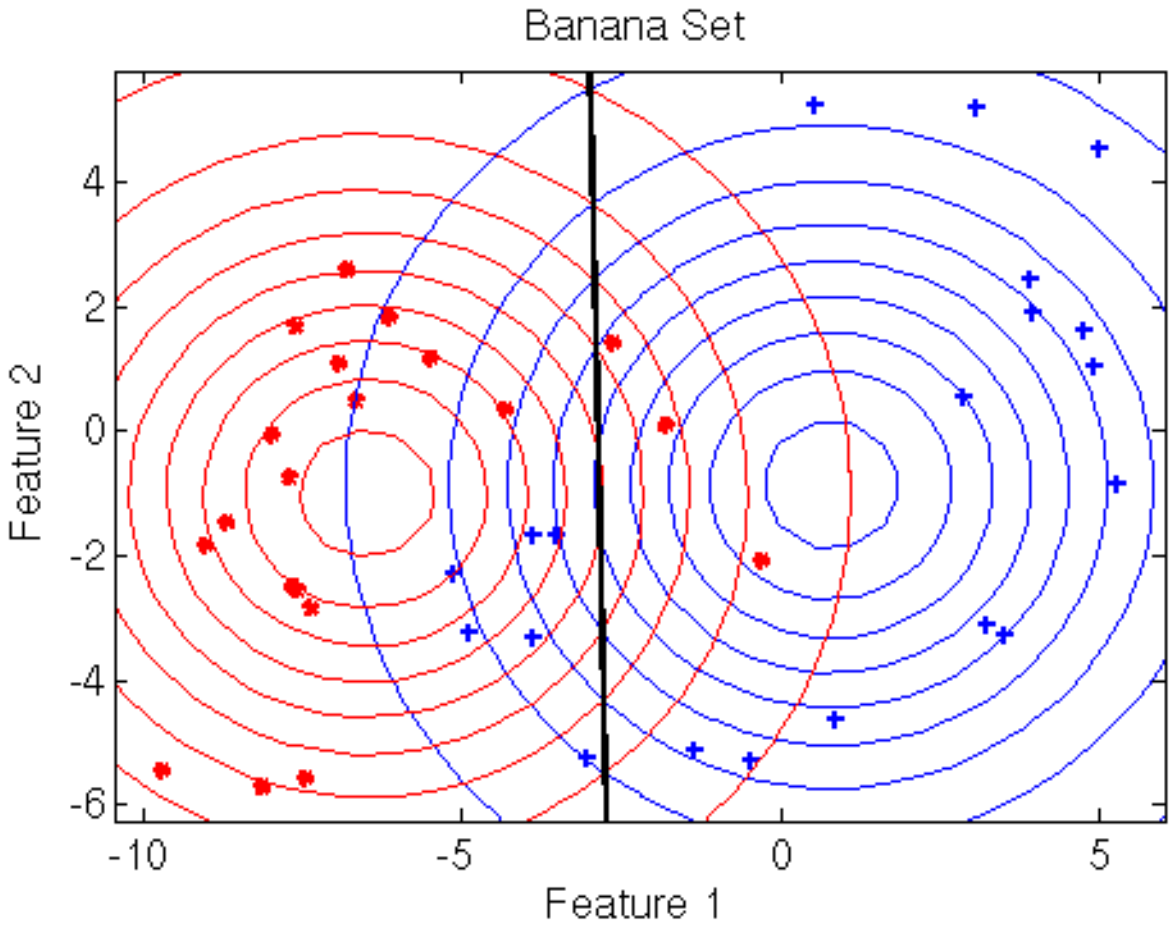
with $\boldsymbol{w} = \hat{\mu}_1 - \hat{\mu}_2$

$$w_0 = \frac{1}{2}\hat{\mu}_2^T \hat{\mu}_2 - \frac{1}{2}\hat{\mu}_1^T \hat{\mu}_1 + \hat{\sigma}^2 \log \frac{p(\omega_1)}{p(\omega_2)}$$

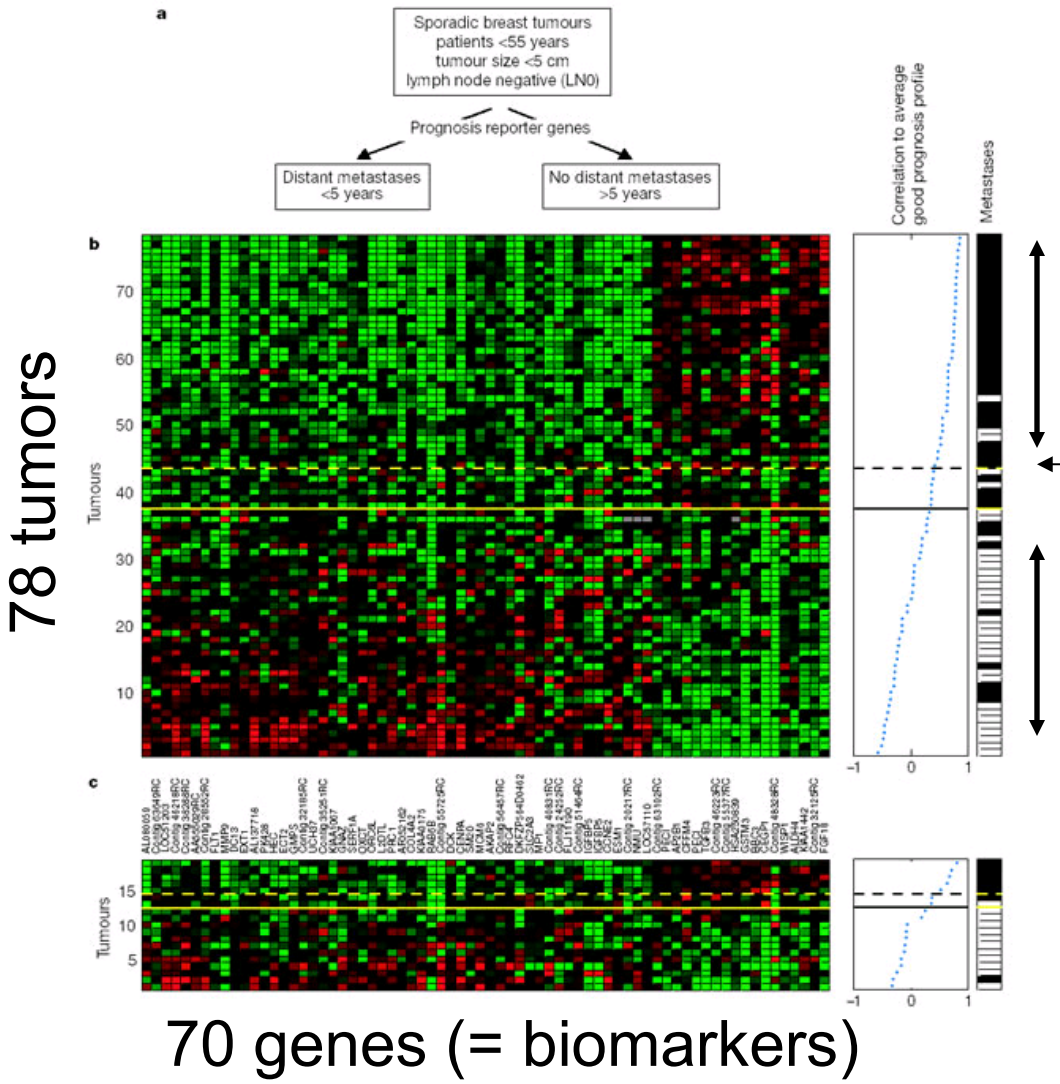
- Again a linear classifier, but it only uses the distance to the mean of each of the classes: *nearest mean* classifier

BioSB

Nearest mean on banana data



Nearest mean on gene expression data



78 tumors

70 genes (= biomarkers)

Two classes:

no metastases > 5yr

metastases < 5yr

good prognosis

threshold

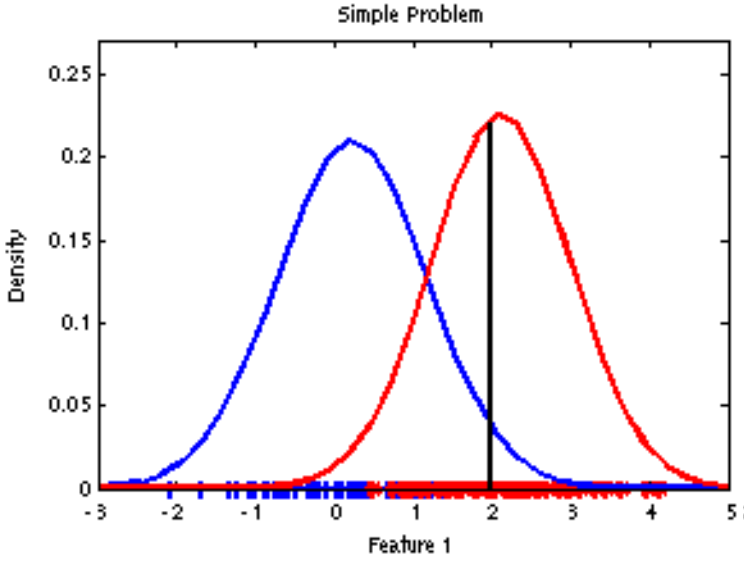
bad prognosis

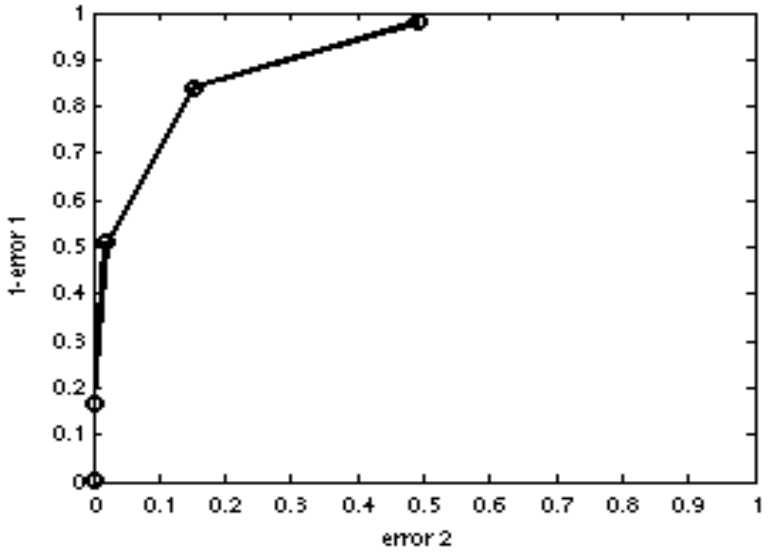
Van 't Veer et al, Nature **415**, 530 (2002)

BioSB

ROC curve

- Error as a function of the threshold gives an overview of all possible (cost/prior) scenarios: *receiver-operator characteristic curve*
- Classifier: any x left of the threshold belongs to the blue class, any x to the right to the red class

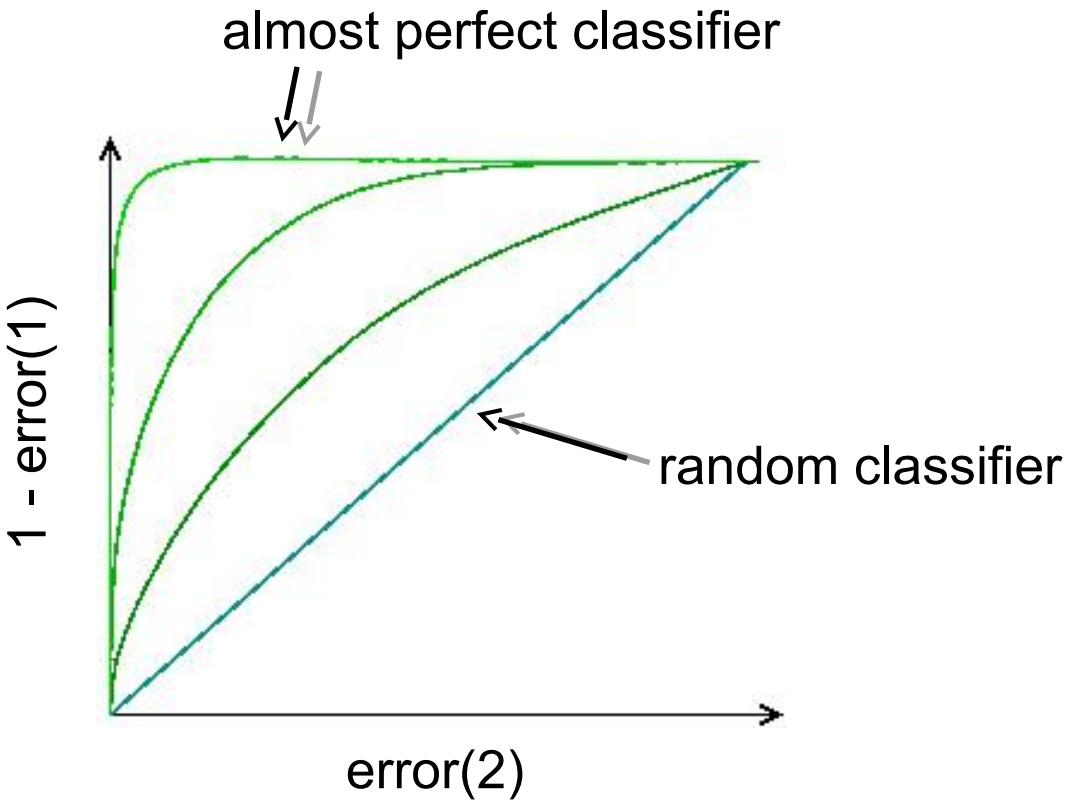




BioSB

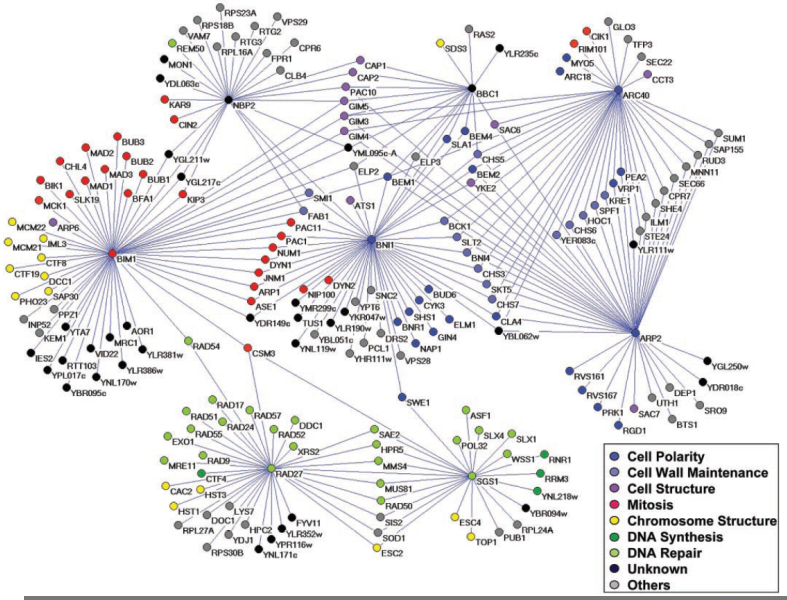
ROC curve (2)

- Different classifiers have different ROC curves

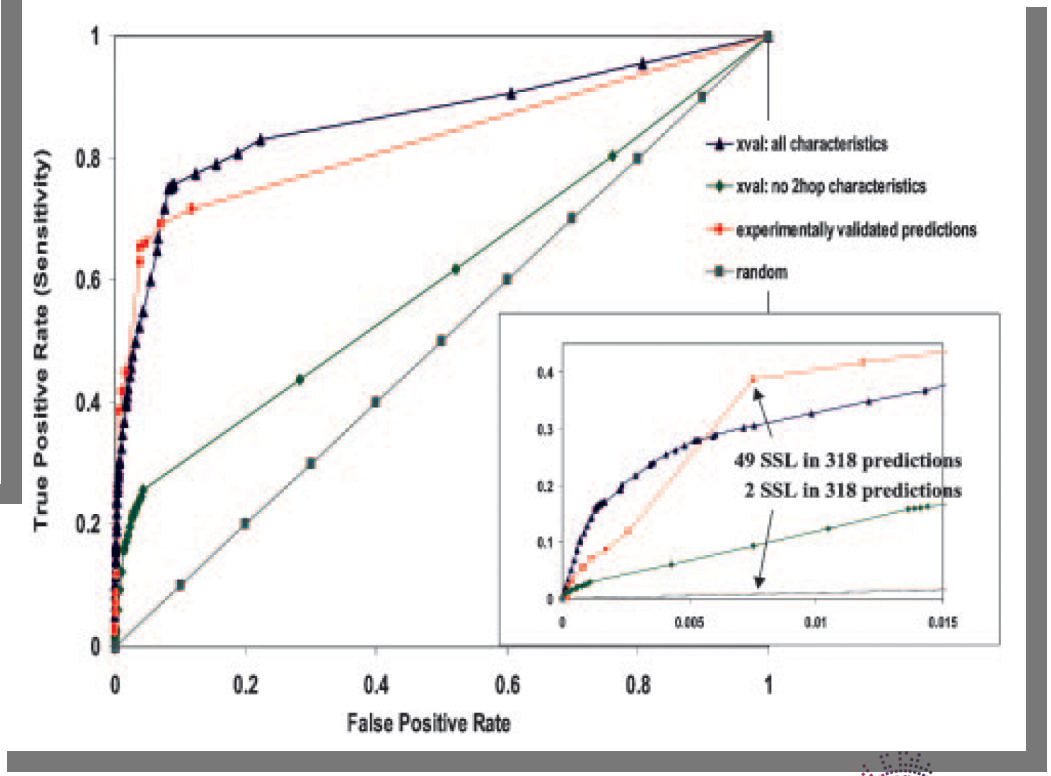


ROC curve (3)

- Example: prediction of synthetic genetic interactions (SGAs)



Wong et al., PNAS 2004



Recapitulation

- Using the Parzen density and nearest neighbor density we can derive the Parzen classifier and nearest neighbor classifier
- Using the plug-in Bayes' rule with a normal distribution for each of the classes gives different classifiers
  - Separate mean and covariance matrix per class gives the quadratic classifier
  - Separate mean, equal covariance matrix per class gives the linear classifier (see Fisher classifier, for two classes)
  - Separate mean, identity covariance matrix per class gives the nearest mean classifier
- By changing the thresholds a ROC curve is obtained, showing the error on both classes.

BioSB

Exercises 2.1-2.11

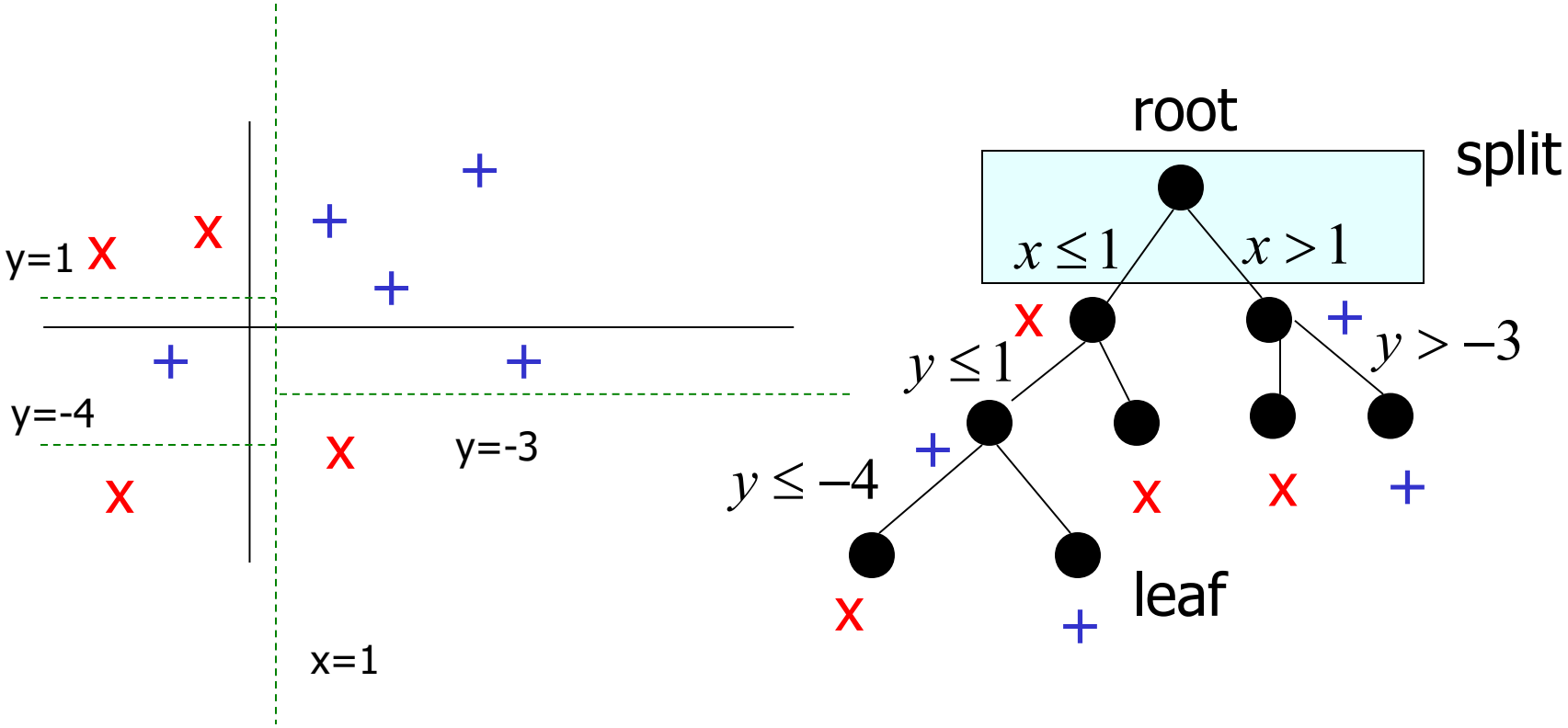
Tree-based models

- Until now: mainly linear and quadratic decision surfaces, often real data is more complex

- Classification trees
  - Feature selection
- Random forests
  - Ensemble of trees
  - Randomization
  - Bootstrapping
- More on Day 5: neural networks, support vector machines

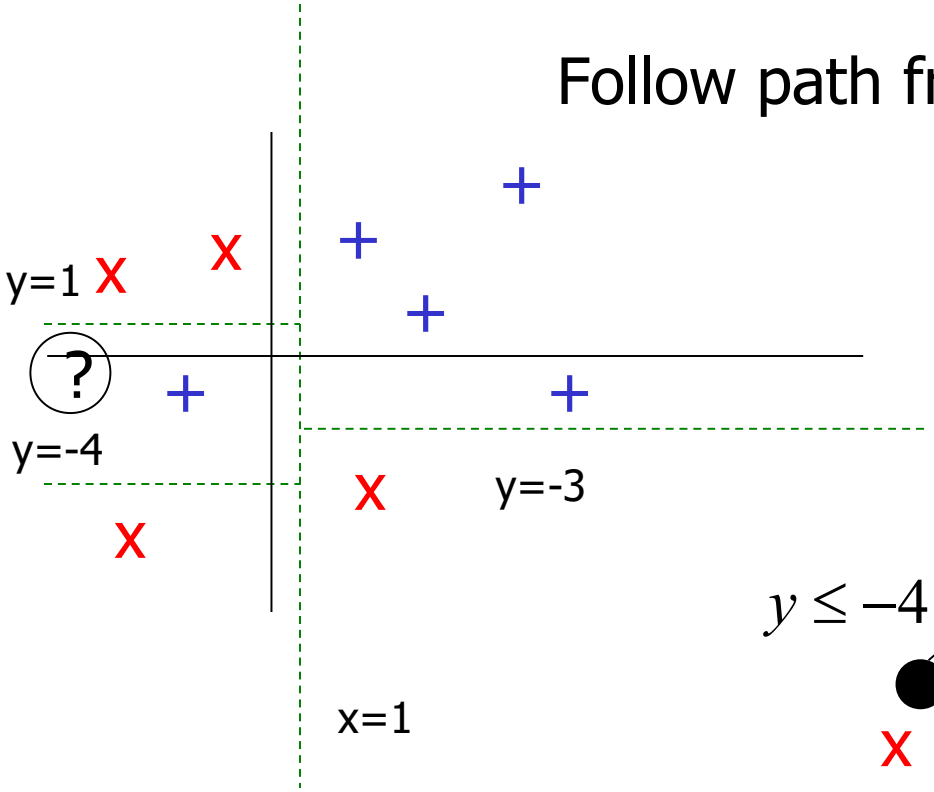
BioSB

Classification trees

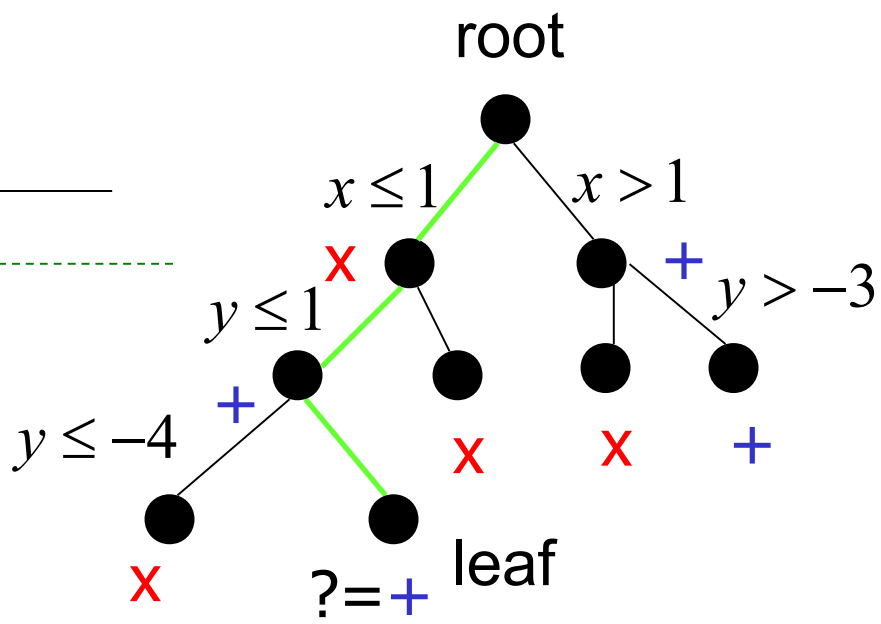


Build a tree of (binary) splits parallel to the axes in a greedy (=one by one) way.

Classification trees: new data

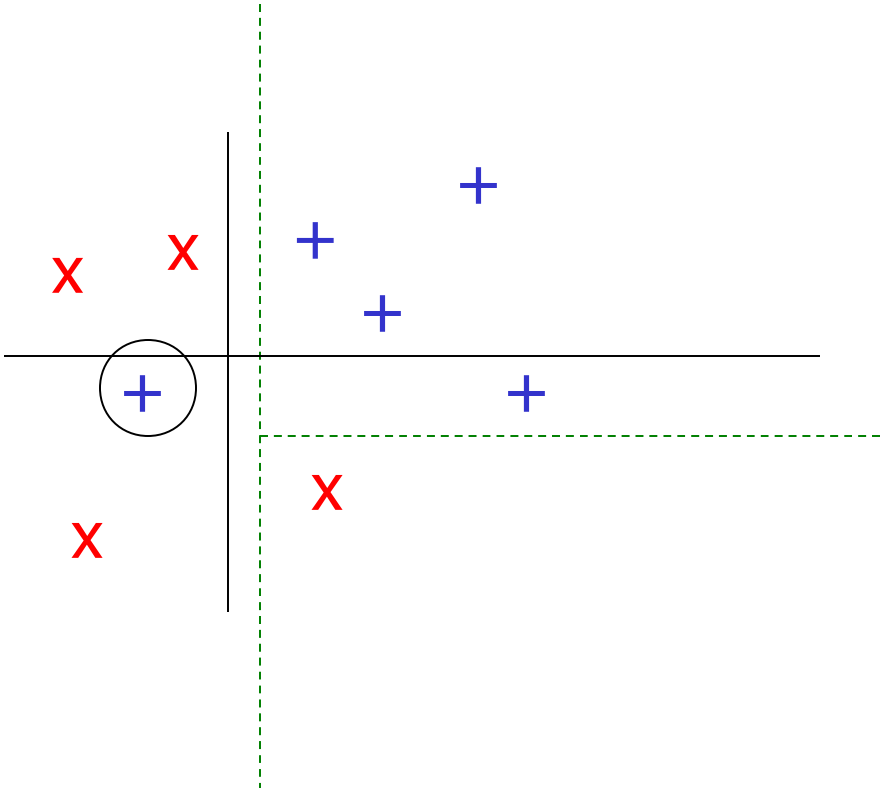


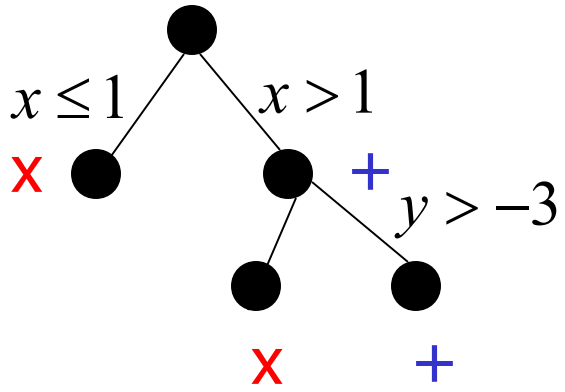
Follow path from root to one of the leaves



Can perfectly fit the data: overfitting

Classification trees: pruning





Allow errors on training data in order to reduce overfitting

Tree ingredients

Trees are constructed in a greedy way:
starting with an empty tree and adding splits one by one
(and never coming back on a decision taken)

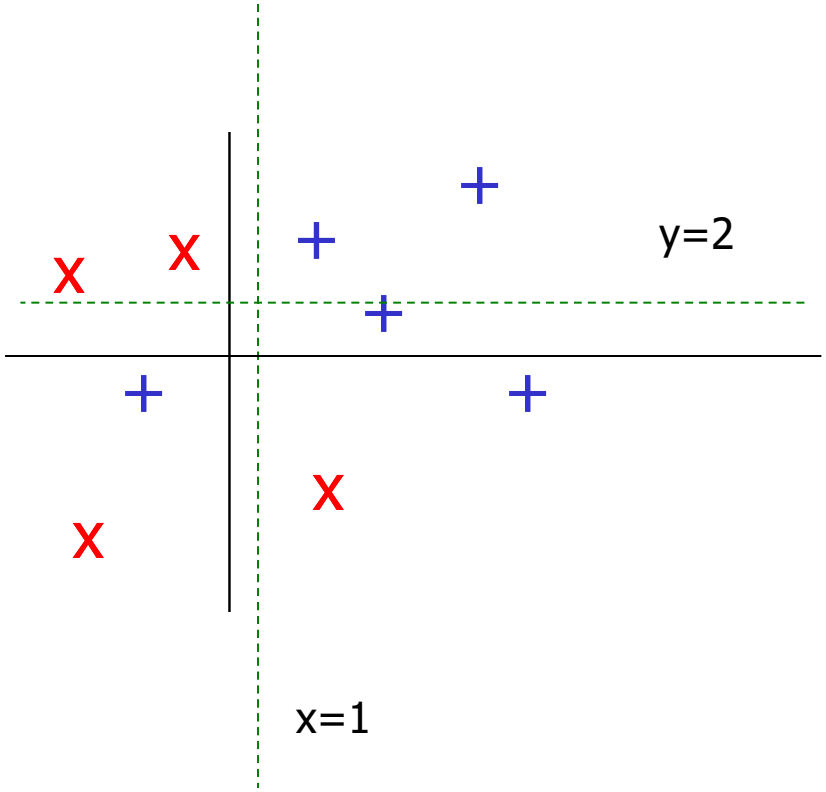
Main questions:

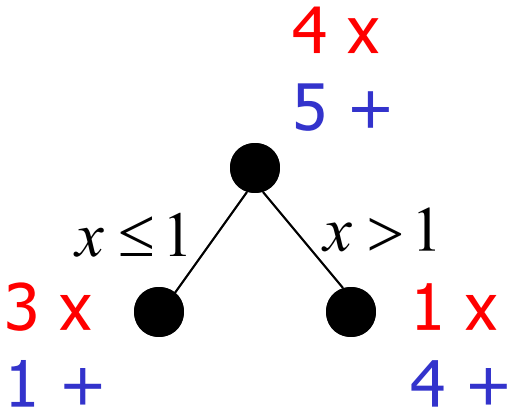
- How to choose a split
- How to choose a final tree?
  - Amount of pruning

Rest: details (but might be important …)

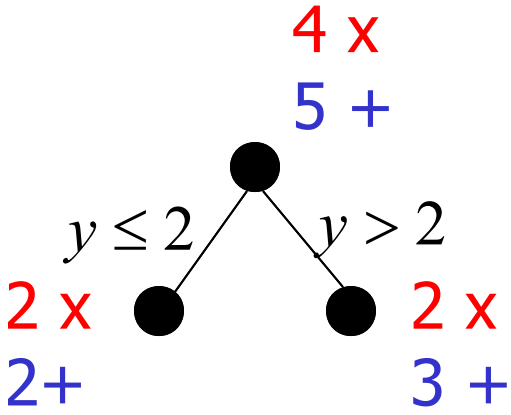
BioSB

How to choose a split?



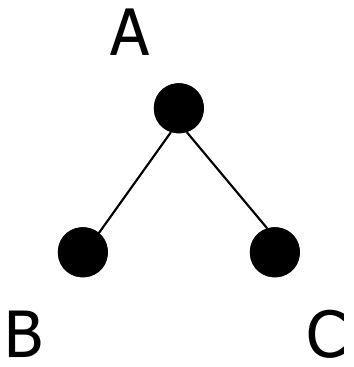


good



bad

How to choose a split? (2)

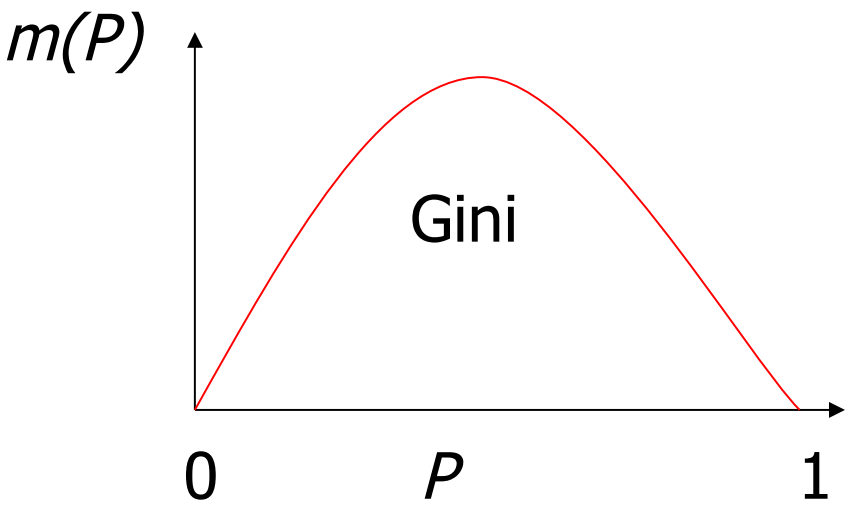


Good split at A:

- few x & many + in B, C

- many x & few + in B, C

Find some measure *m* that captures goodness



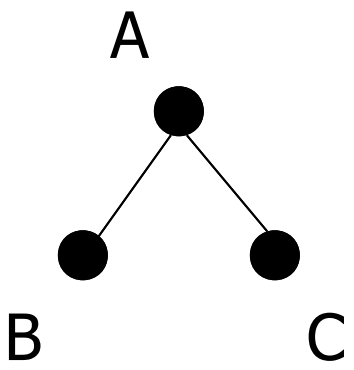
$m(0)=0$

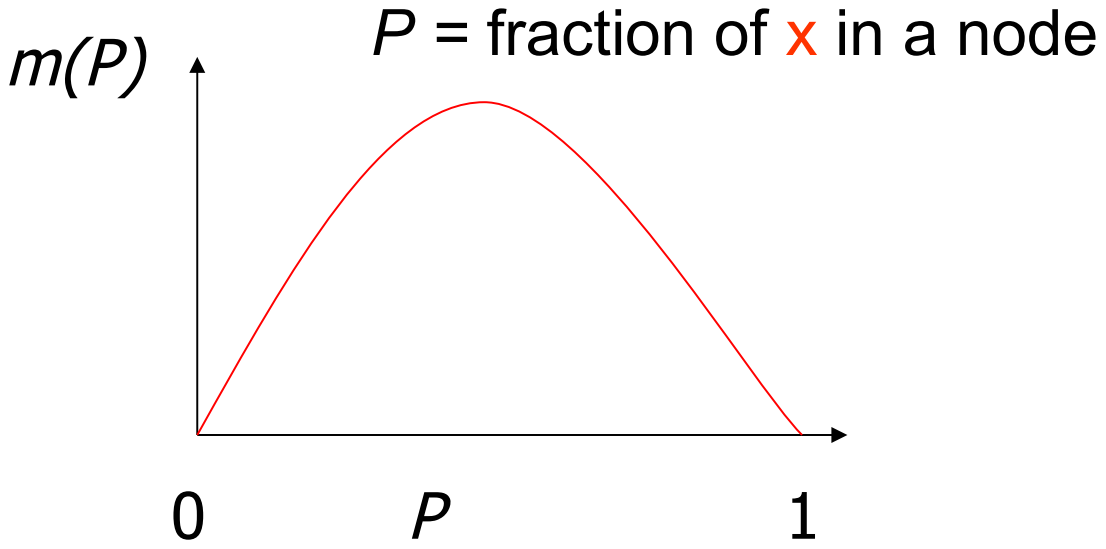
$m(1)=0$

$m(P)=P(1-P)$

P = fraction of x in a node

How to choose a split? (3)

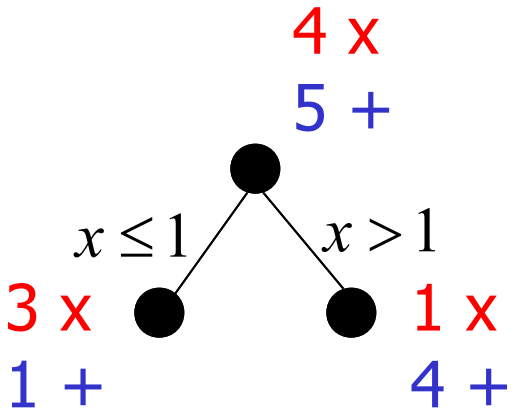




maximize $m(P_A) - P(B)m(P_B) - P(C)m(P_C)$

$P(X)$: determined by number of x and + at node X

How to choose a split? (4)



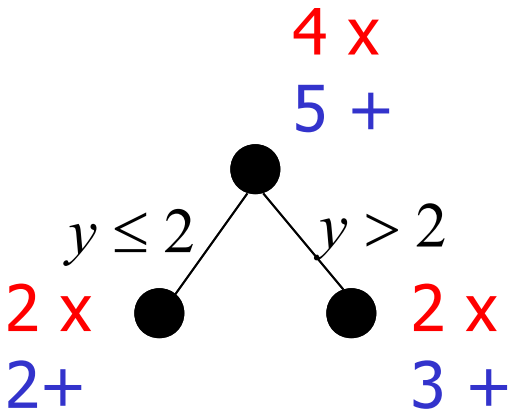
good

$$m(P_A) - P(B)m(P_B) - P(C)m(P_C)$$

$$\frac{4}{9}\frac{5}{9} - \frac{4}{9}\frac{3}{4}\frac{1}{4} - \frac{5}{9}\frac{1}{5}\frac{4}{5} =$$

0.075

maximum



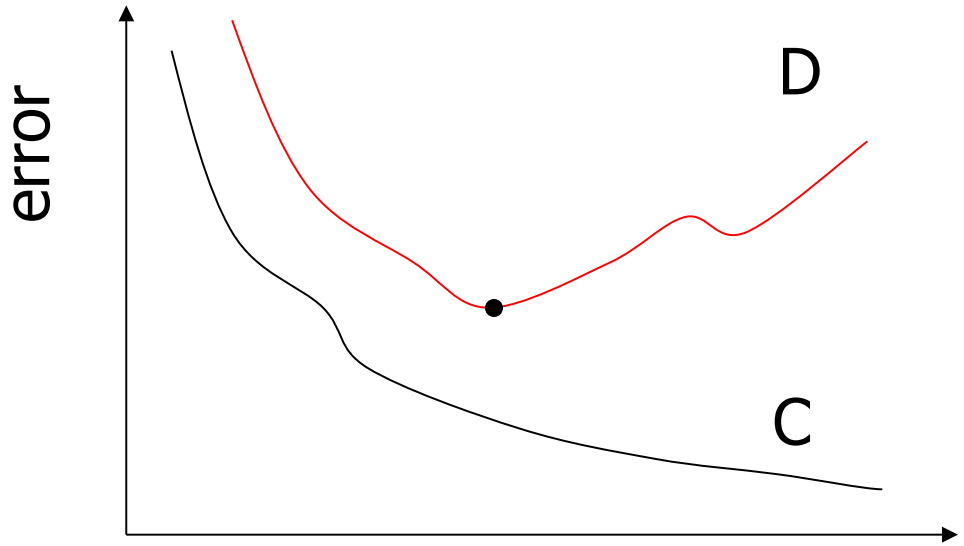
bad

$$\frac{4}{9}\frac{5}{9} - \frac{4}{9}\frac{2}{4}\frac{2}{4} - \frac{5}{9}\frac{2}{5}\frac{3}{5} =$$

0.0025

BioSB

Pruning: one step back

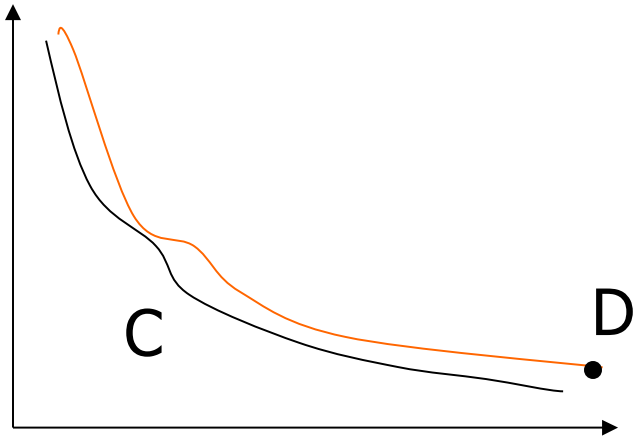


minimize: D = C + *k*(# of leaf nodes in the tree)

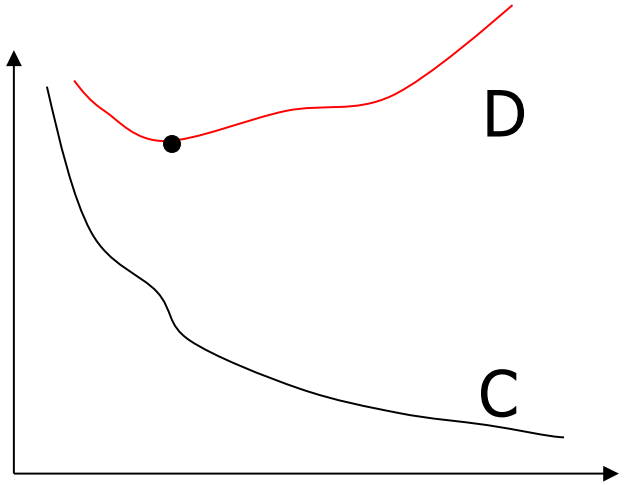
$0 \leq k$ *k*: complexity parameter

k penalizes big trees

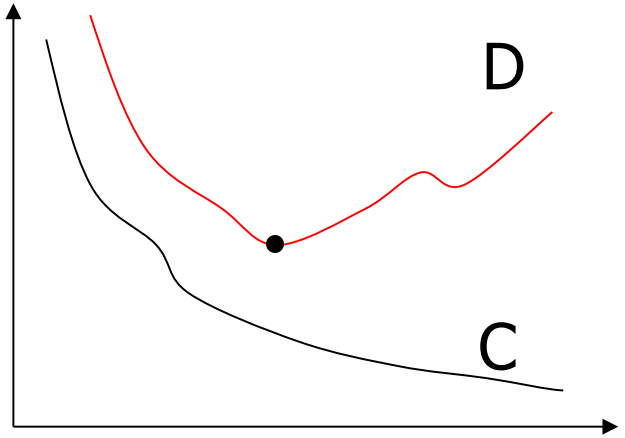
Pruning: one step back (2)



small *k*: big tree



large *k*: small tree



medium *k*: medium tree

BioSB

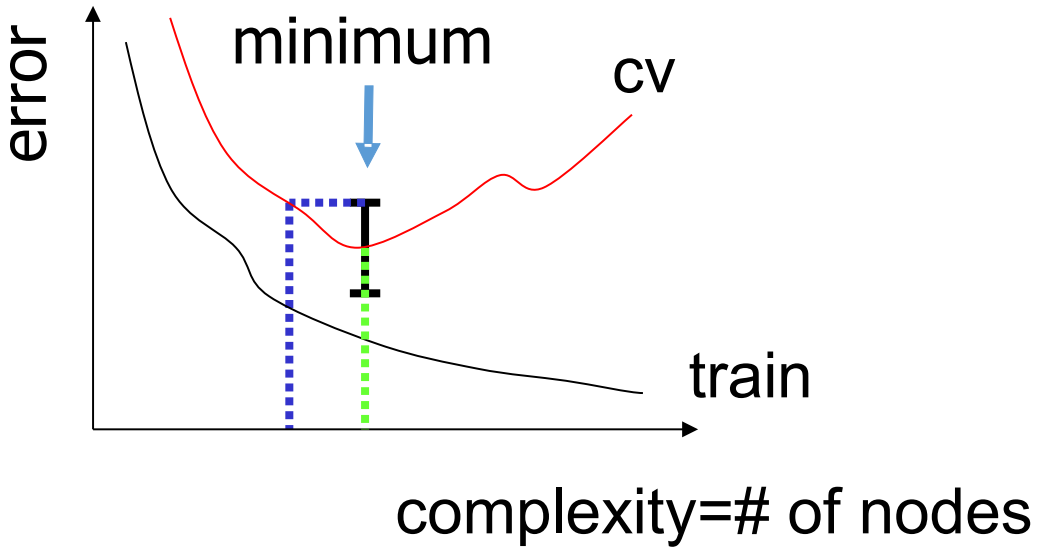
Pruning: CART

- Build a complete tree T
- With each subtree of T corresponds a choice of k

Cannot make choice of k on training set: overfitting

Optimal choice of k is made by cross-validation

Pruning: model selection



minimum

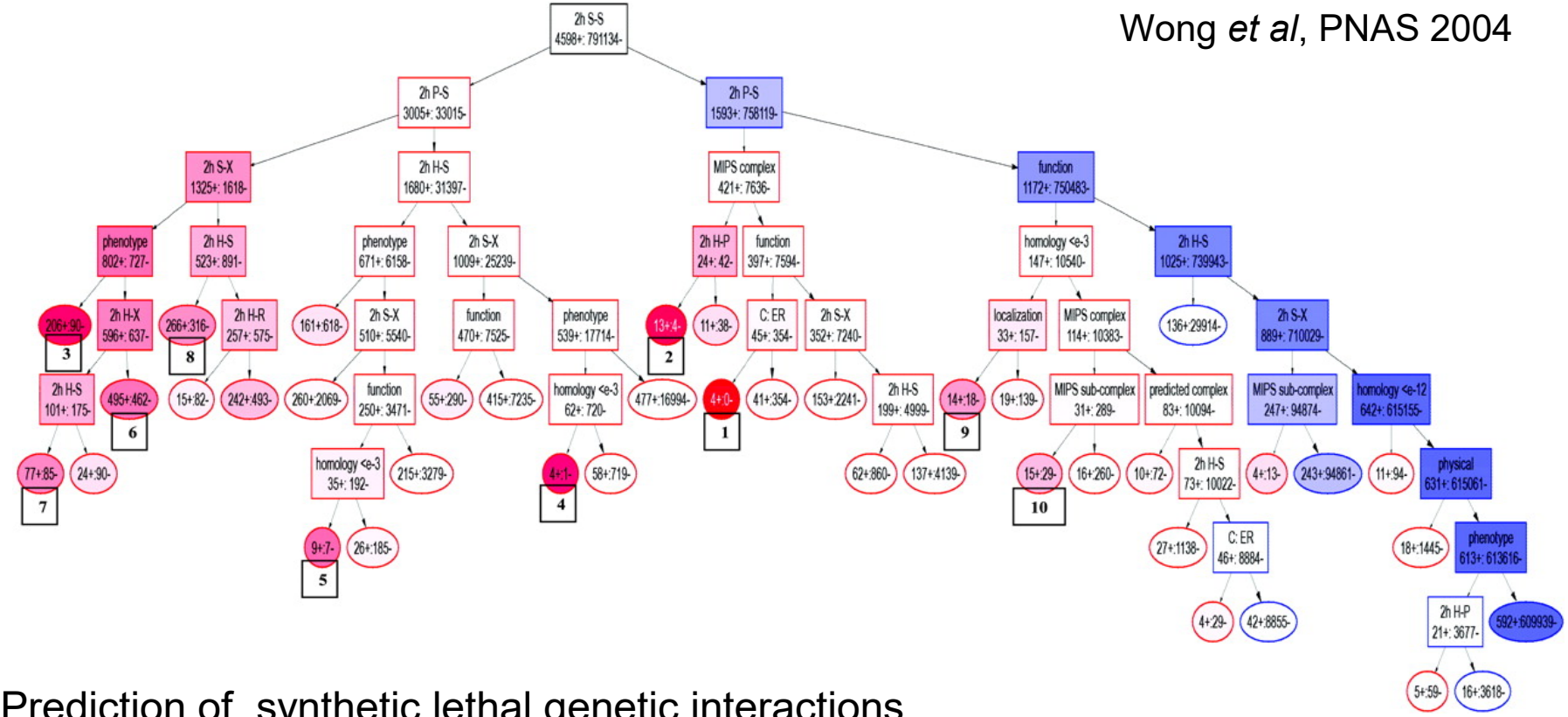
standard error (1-se)

10-fold cross-validation: mean +/- std. error

BioSB

Decision tree: application

Wong *et al*, PNAS 2004



- Prediction of synthetic lethal genetic interactions
- Integrate multiple types of data: localization, mRNA expression, physical interaction, protein function, and characteristics of network topology

BioSB

Advantages/disadvantages

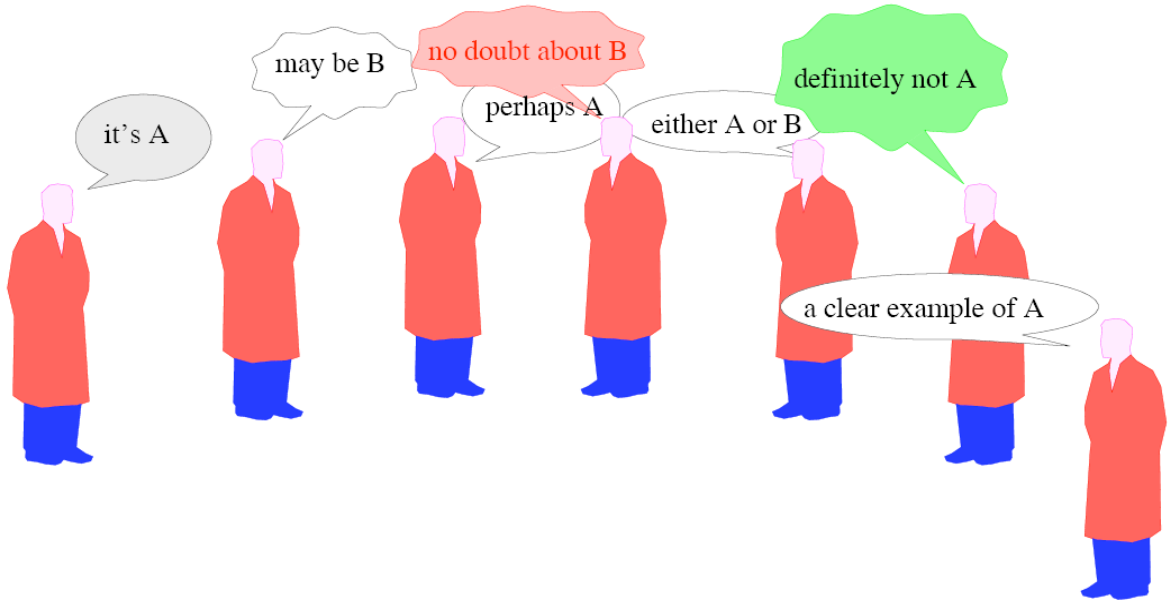
- simple and flexible classifier
- combination of discrete and continuous features
- feature selection (Day 3)
- interpretability

- hard splits
- splits are axis-aligned
- sensitive to small variations in data (high variance, Day 5)

BioSB

Classifier combination

- Idea: combine different classifiers and have them vote
- Design choices:
  - Identical or different?
    - Base classifiers, feature spaces, training sets, initialisations, etc.
  - Combination by a fixed rule or by another classifier?



BioSB

Example: random forests

- General overview: Day 4
- Specific example: random forest – an ensemble of decision trees

- Choices to be made:
  - Base classifiers: identical – decision trees
  - Feature spaces: for each node in each tree sample randomly *m* features
    - *m* << total number of features
  - Training sets: sampling with replacement (bootstrapping)
    - About two-third of the cases are used for training each tree
- Combination: majority vote

BioSB

Characteristics

- Out-of-bag error (oob) estimate:
  - Each tree can be tested on about one-third of the cases – the out-of-bag samples

- Variable importance:
  - For each tree: predict the class for oob cases and count the number of votes cast for the correct class
  - For each tree: randomly permute the values of variable *n* in the oob cases and count the number of votes cast for the correct class
  - Importance: rank (from high to low) based on average difference of these two scores

BioSB

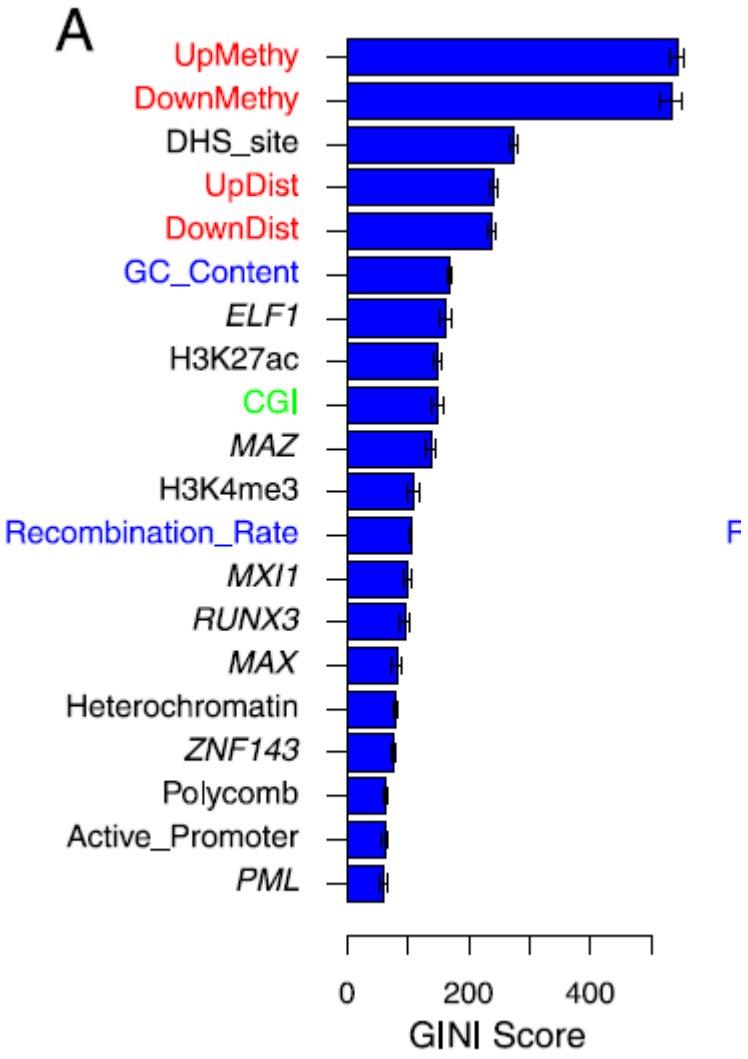
Some intuition

- Breiman *et al.*, Machine Learning (2001) paper
- Accuracy depends on two factors:
  - Correlation between any two trees in the forest. Decreasing correlation increases the forest accuracy: diversity
  - Accuracy of each individual tree (strength) in the forest. Increasing strength of individual trees increases the forest accuracy
- Trade-off:
  - Reducing *m* reduces correlation and strength
  - Increasing *m* increases correlation and strength
- Solution: somewhere in between is an *optimal* range of *m* - usually quite wide. Using the oob error rate a value of *m* in the range can be found

BioSB

Random forests: example

- Prediction of genome-wide DNA methylation
- Features:
  - Neighbors
  - Genomic position
  - DNA sequence properties
  - Cis-regulatory elements
- Random forest: feature selection



Zhang *et al.*, Genome Biology (2015)

Recapitulation

- Decision trees: simple and flexible classifier
  - Incorporates feature selection
  - Interpretable
  - Hard, axis-aligned splits
  - Pruning is essential to avoid overfitting
- Random forest: example of ensemble method
  - Ensemble of decision trees
  - Variation between members introduced via randomness
  - When number of features is large and percentage of truly informative features is small (gene expression-based diagnostics): performance tends to decline significantly

BioSB

Exercises 2.12-2.13

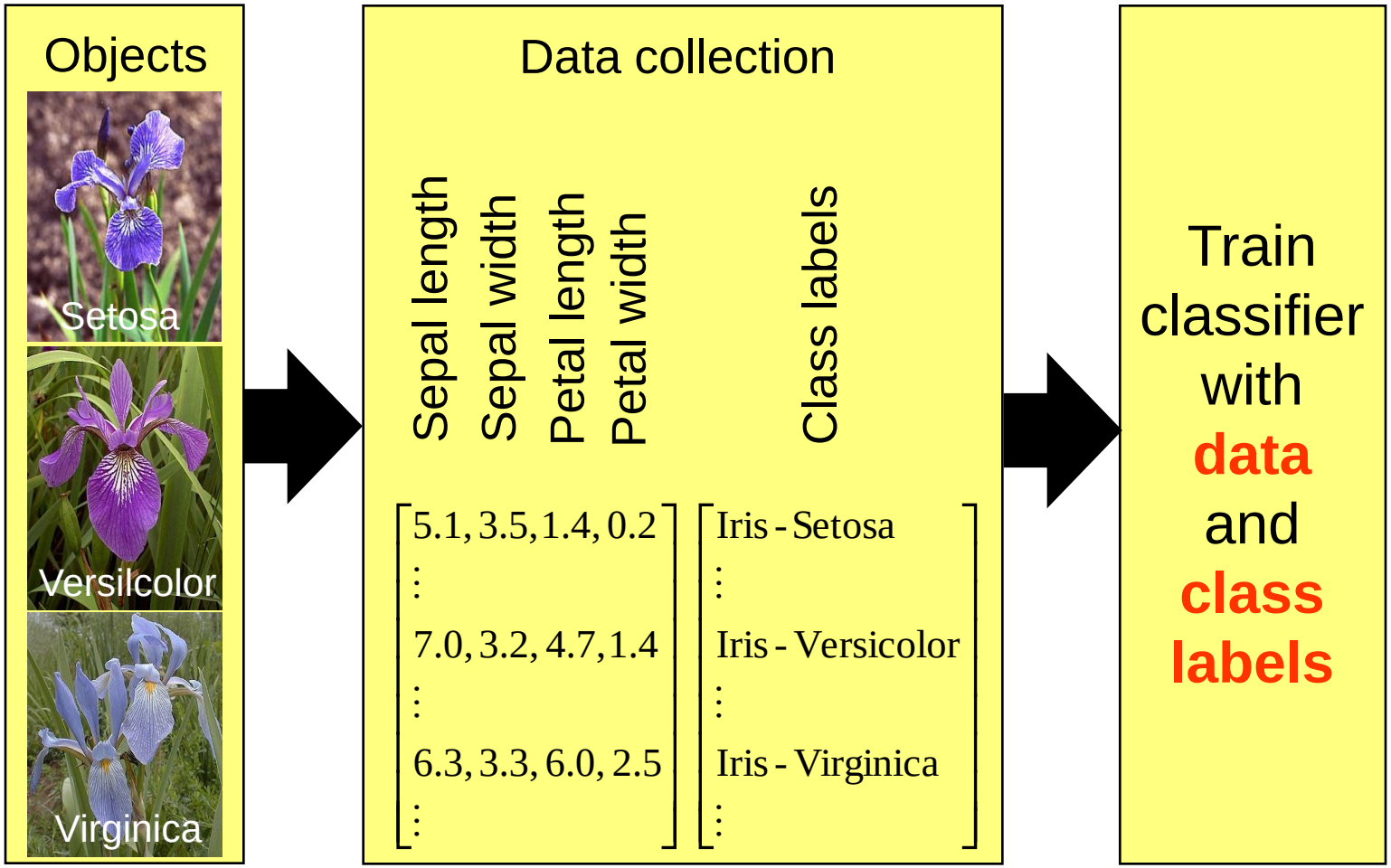
Clustering

- Supervised vs. unsupervised learning

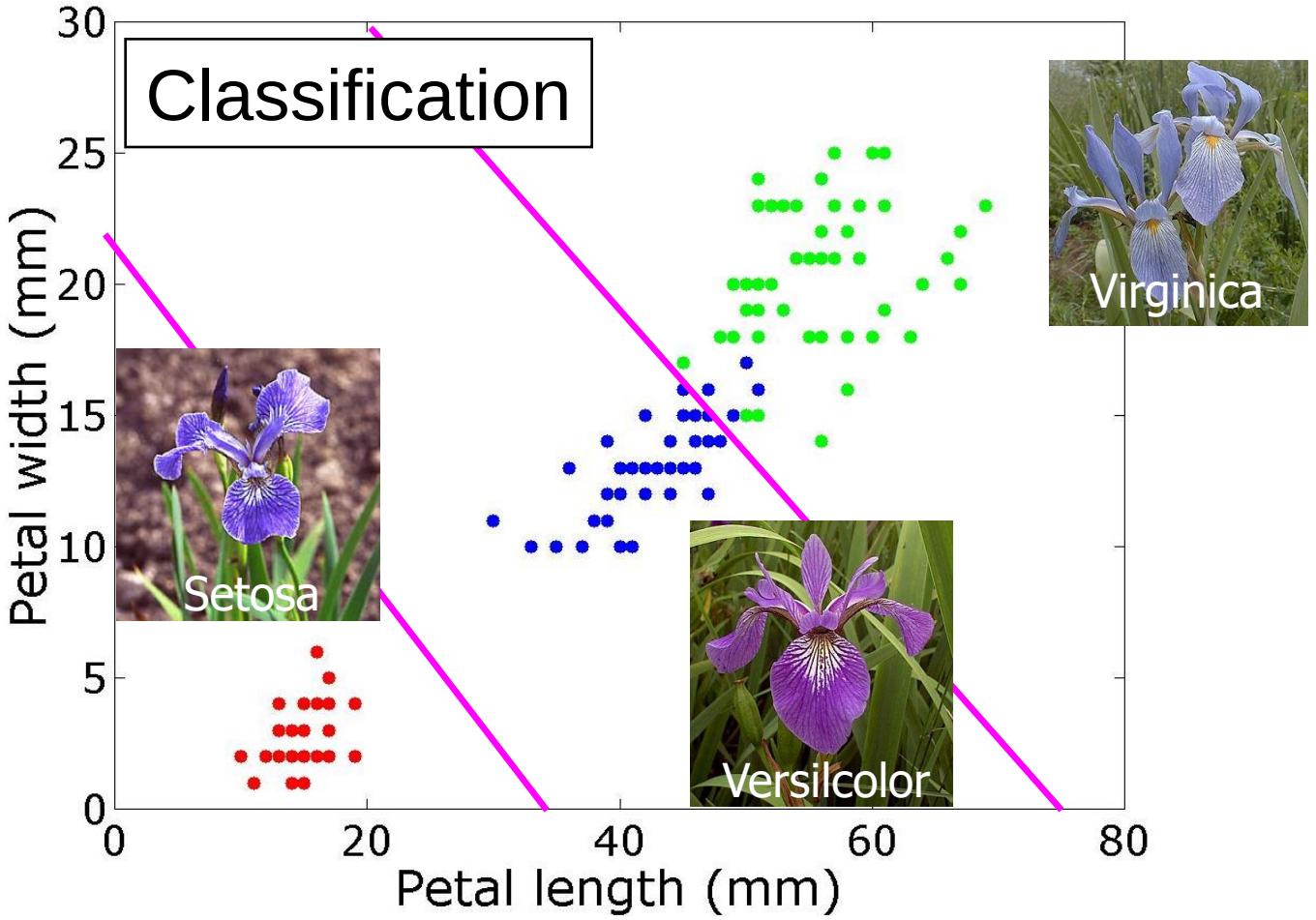
- Hierarchical clustering
- Sum-of-squares clustering (k-means)
- Cluster validation
- Mixtures-of-Gaussians clustering (EM algorithm)

BioSB

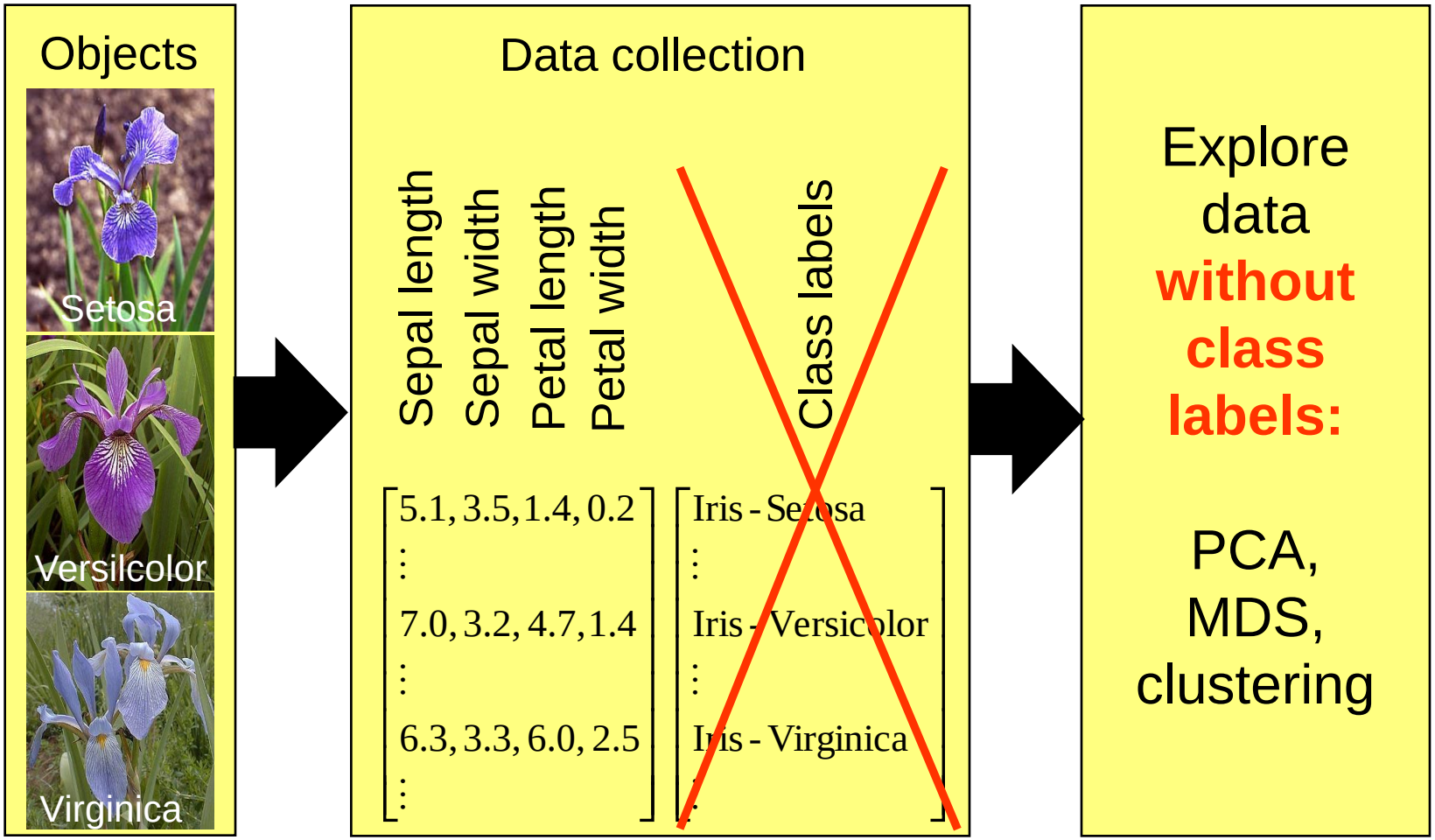
Supervised learning



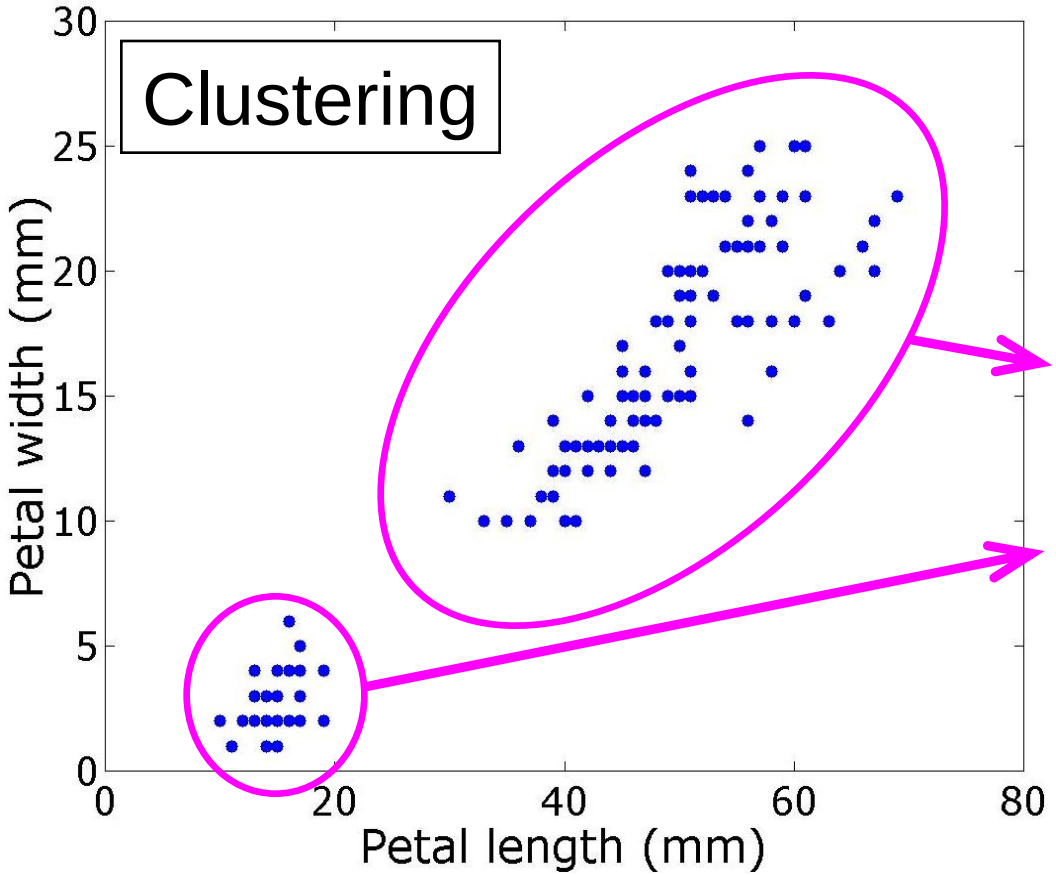
Supervised learning (2)



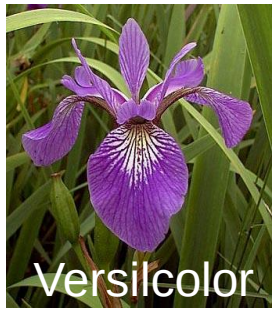
Unsupervised learning



Unsupervised learning (2)

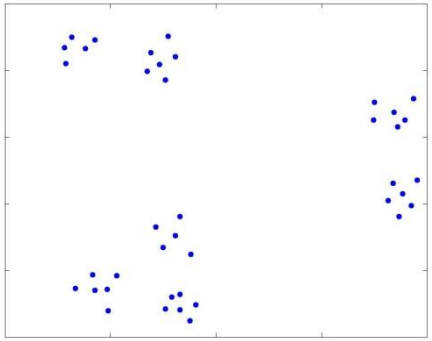


?

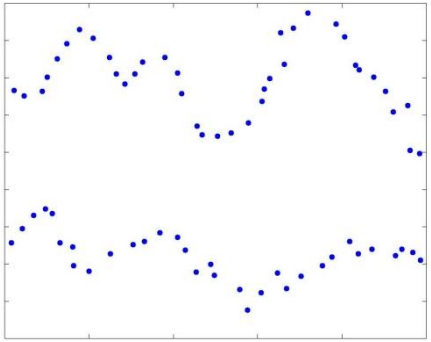


BioSB

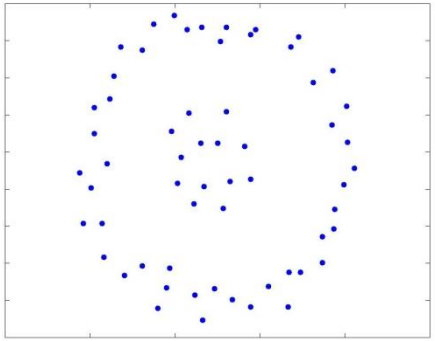
What is a cluster?



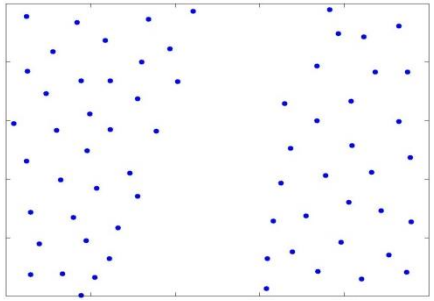
Shape: compact, convex
Separation: large



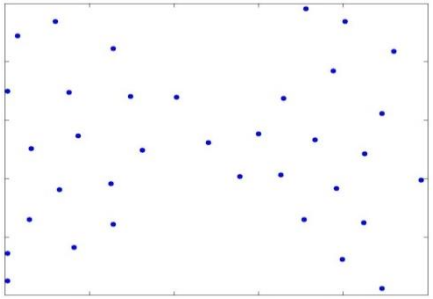
Shape: strings
Separation: large?



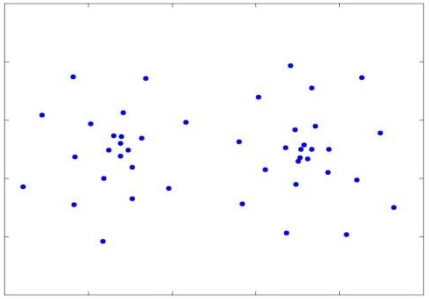
Shape: convex and circular
Separation: large?



Shape: ?
Separation: large?



Shape: loose, convex
Separation: small



Shape: loose, convex
Separation: small

BioSB

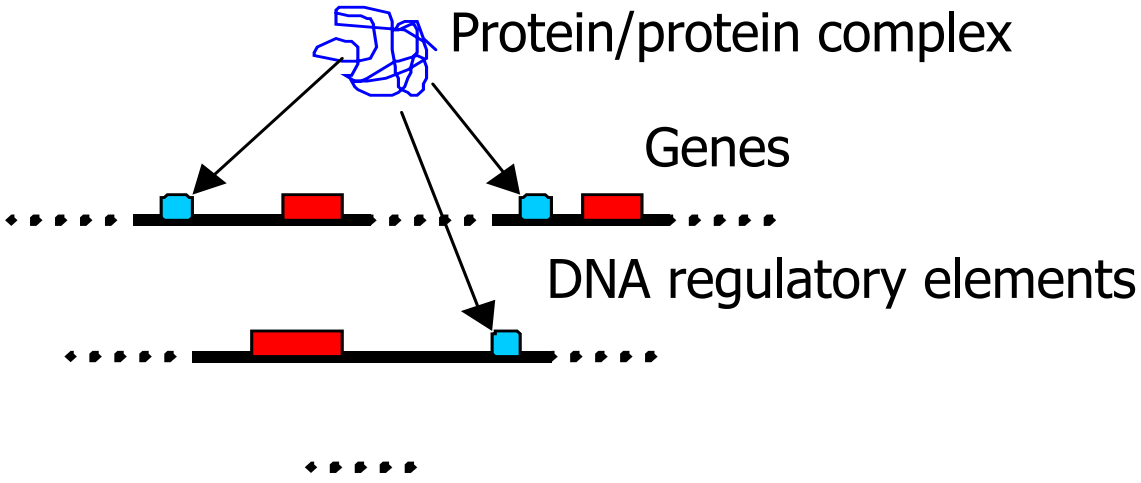
What is a cluster? (2)

- Clustering: finding natural groups in data...
  - which themselves are far apart
  - in which objects are close together
- Define what is “far apart” and “close together”:
  - Need a distance measure or dissimilarity measure
  - This measure should capture what we think is important for the grouping
  - The choice for a certain distance measure is often the most important choice in clustering!
- There is no such thing as *the objective clustering*

BioSB

What is a cluster in bioinformatics?

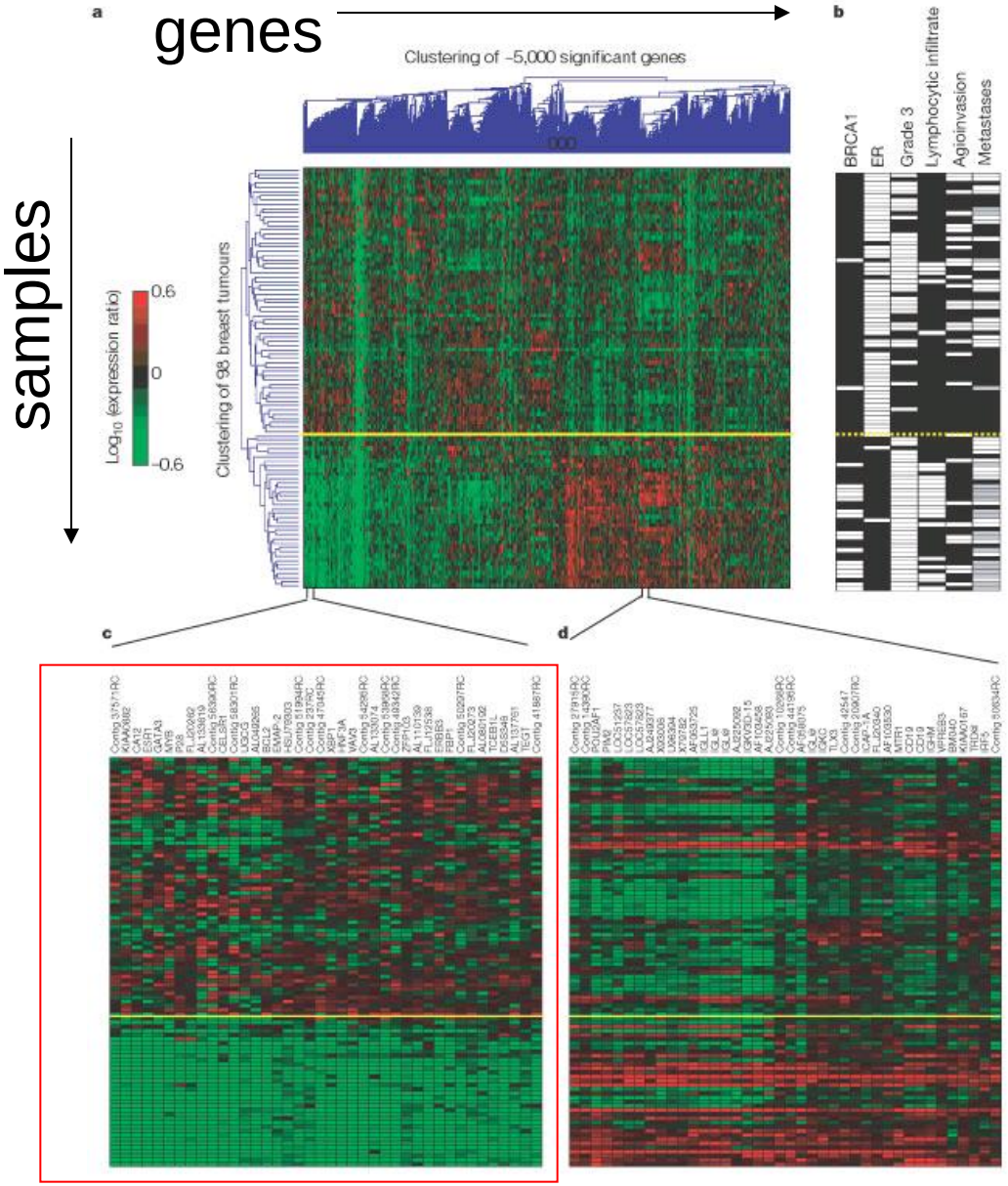
- Clustering gene expression data:
- Genes: similar ~ co-expression ~ co-regulation ~
  same pathway / same function



- Samples: similar ~ same type of tissue
- Used for discovery of new subclasses (*subtypes*) in tumors

BioSB

Example: genes (and samples)



negative

positive

histopathological data

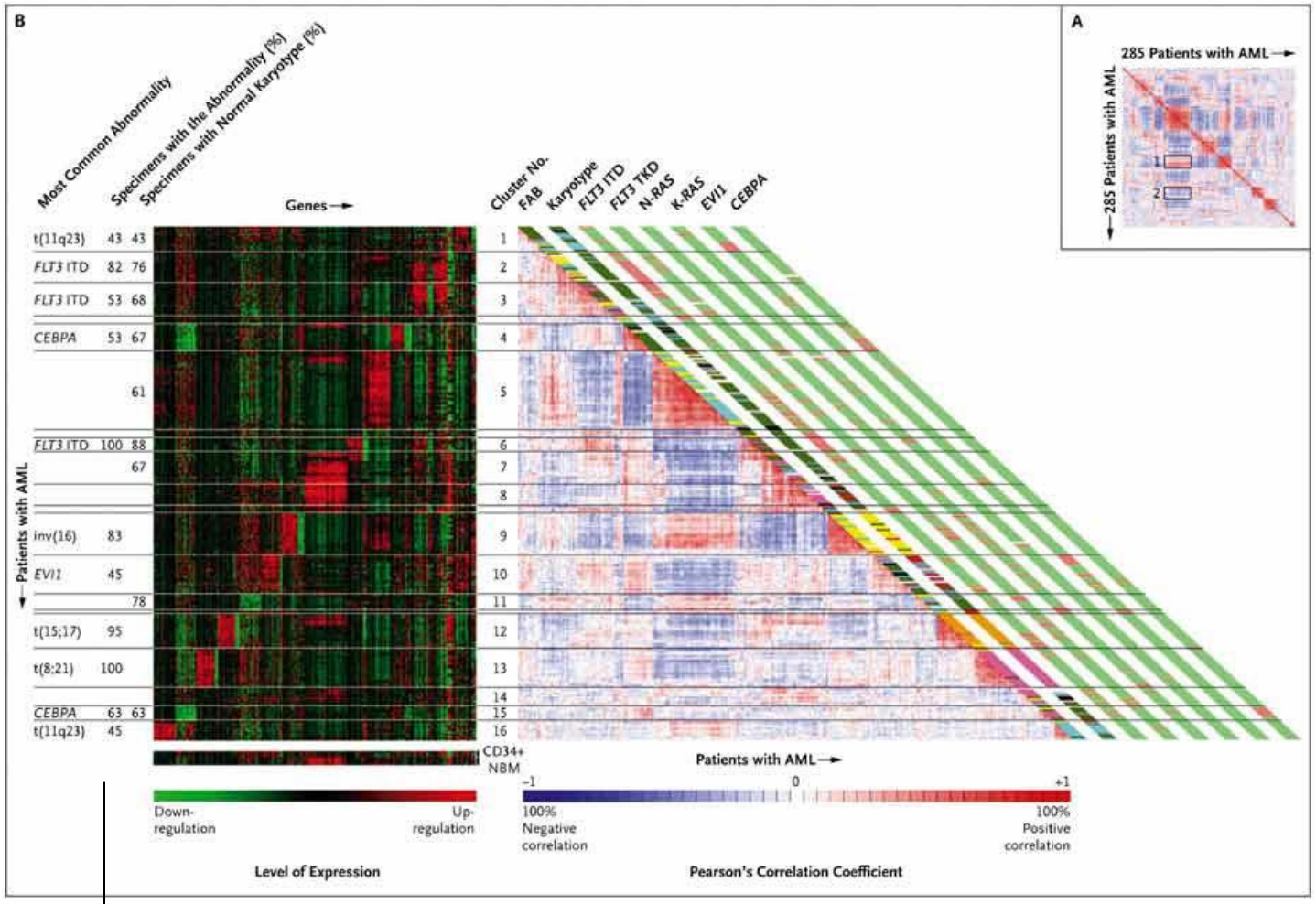
ER gene *(ESR1)* and genes co-regulated with ER, some of which are known ER target genes

Van 't Veer et al, Nature 415: 530-536 (2002)

BioSB

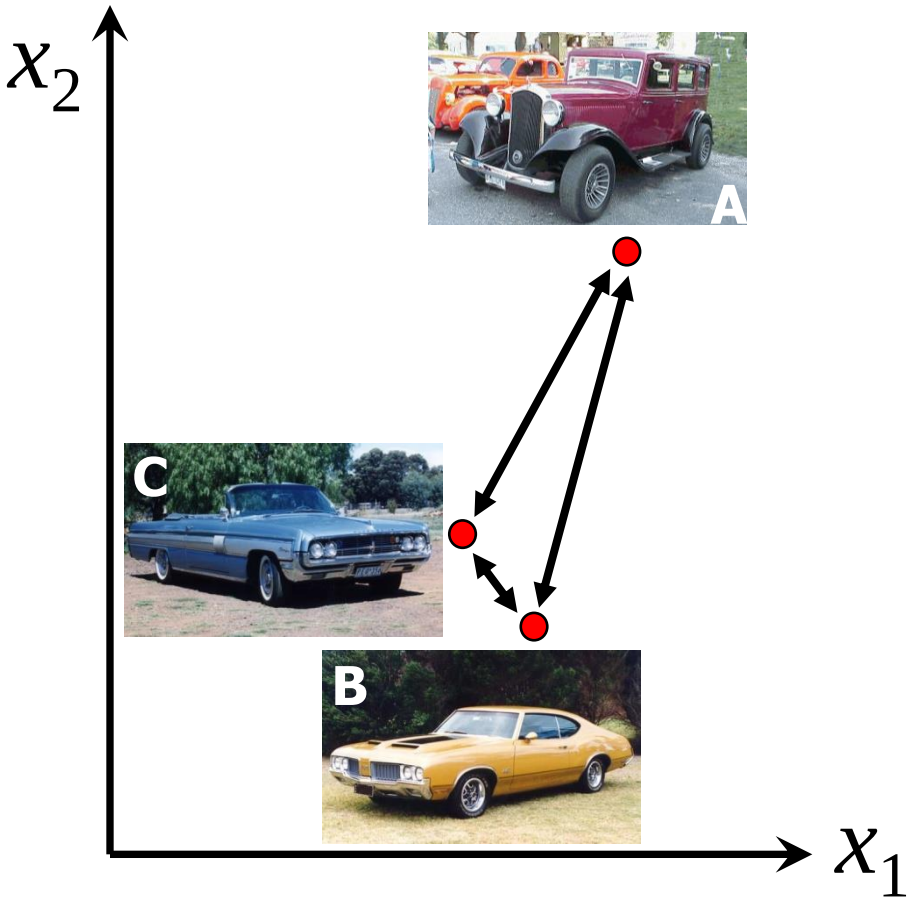
Example: samples

Valk et al, N Engl J Med. 2004 Apr 15;350(16):1617-28.



Identified 16 groups of patients with acute myeloid leukemia

Dissimilarity measures



$$\boldsymbol{D} = \begin{bmatrix} 0 & d(\mathbf{A},\mathbf{B}) & d(\mathbf{A},\mathbf{C}) \\ & 0 & d(\mathbf{B},\mathbf{C}) \\ & & 0 \end{bmatrix}$$

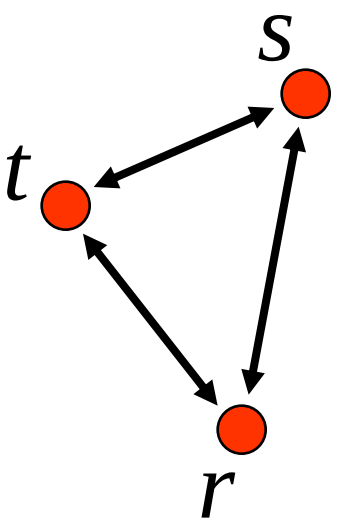
$$= \begin{bmatrix} 0 & 10 & 11 \\ & 0 & 2 \\ & & 0 \end{bmatrix}$$

BioSB

Dissimilarity measures (2)

- Let $d(r,s)$ be the dissimilarity between objects *r* and *s*
- Formally, dissimilarity measures should satisfy

$$d(r,s) \geq 0, \forall\, r,s$$
$$d(r,r) = 0, \forall\, r$$
$$d(r,s) = d(s,r), \forall\, r,s$$



- If in addition, the triangle inequality holds, the measure is a *metric*

$$d(r,t) + d(t,s) \geq d(r,s), \forall\, r,s,t$$

- Most often used: Euclidean distance (metric)

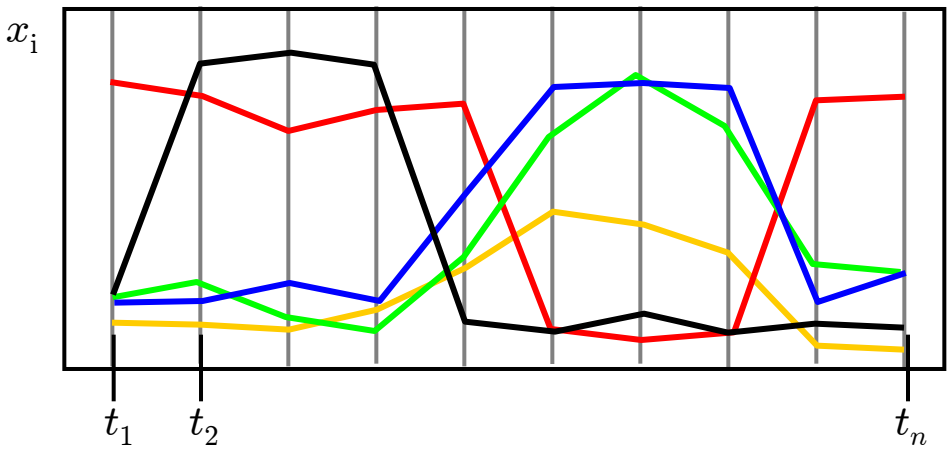
BioSB

Dissimilarity measures (3)

- Example: time series data

(squared) Euclidean distance

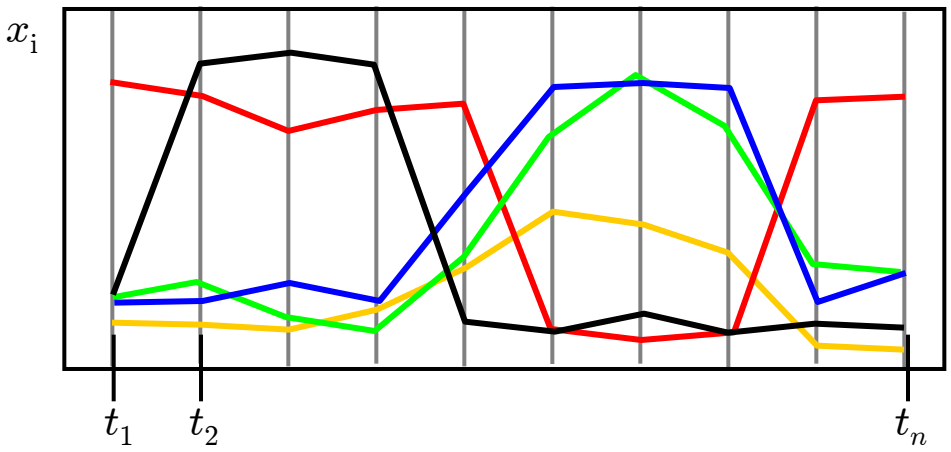
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{n} (x_{i,t} - x_{j,t})^2$$



BioSB

Dissimilarity measures (3)

- Example:
  time series data



Euclidean distance
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{n} (x_{i,t} - x_{j,t})^2$$

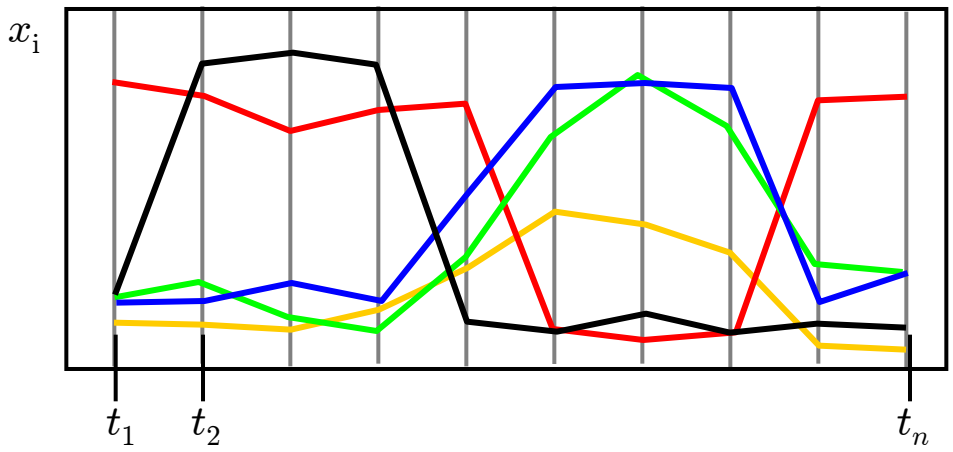
d(● , ●) < d(● , ●)
d(● , ●) << d(● , ●)
d(● , ●) << d(● , ●)

Dissimilarity measures (3)

- Example: time series data

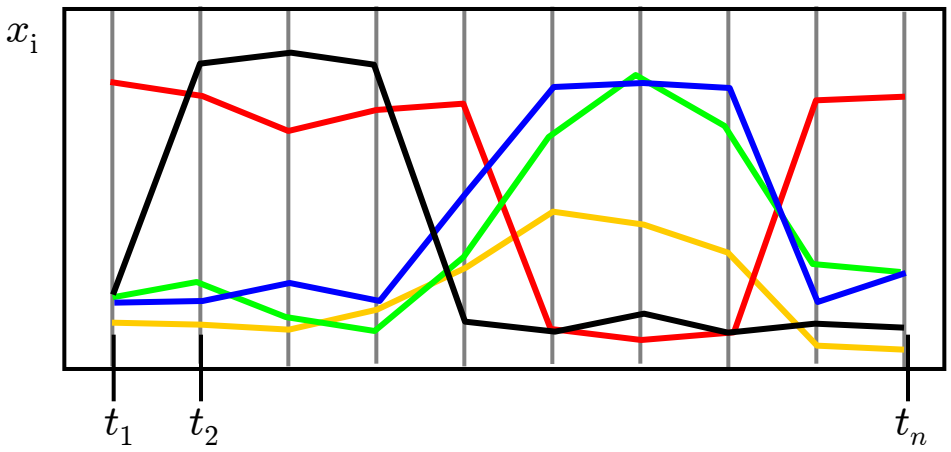
Euclidean distance
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{n} (x_{i,t} - x_{j,t})^2$$



Dissimilarity measures (3)

- Example: time series data



Euclidean distance
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{n} (x_{i,t} - x_{j,t})^2$$

d(● , ●) < d(● , ●)
d(● , ●) << d(● , ●)
d(● , ●) << d(● , ●)

Pearson correlation
ignore amplitude

$$\rho_{ij} = \sum_{t=1}^{n} (x_{i,t} - \mu_i)(x_{j,t} - \mu_j) \Big/ \sigma_i \sigma_j$$

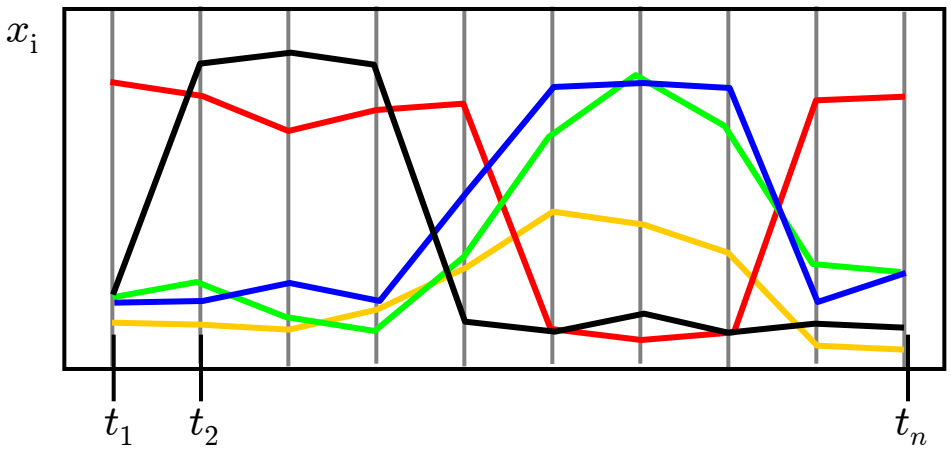
$1 - \rho_{ij}$

d(● , ●) ≈ d(● , ●)
d(● , ●) << d(● , ●)
d(● , ●) << d(● , ●)

BioSB

Dissimilarity measures (3)

- Example: time series data



Euclidean distance
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{n} (x_{i,t} - x_{j,t})^2$$

d(● , ●) < d(● , ●)
d(● , ●) << d(● , ●)
d(● , ●) << d(● , ●)

Pearson correlation
ignore amplitude

$$\rho_{ij} = \sum_{t=1}^{n} (x_{i,t} - \mu_i)(x_{j,t} - \mu_j) \Big/ \sigma_i \sigma_j$$

$$1 - \rho_{ij}$$

d(● , ●) ≈ d(● , ●)
d(● , ●) << d(● , ●)
d(● , ●) << d(● , ●)

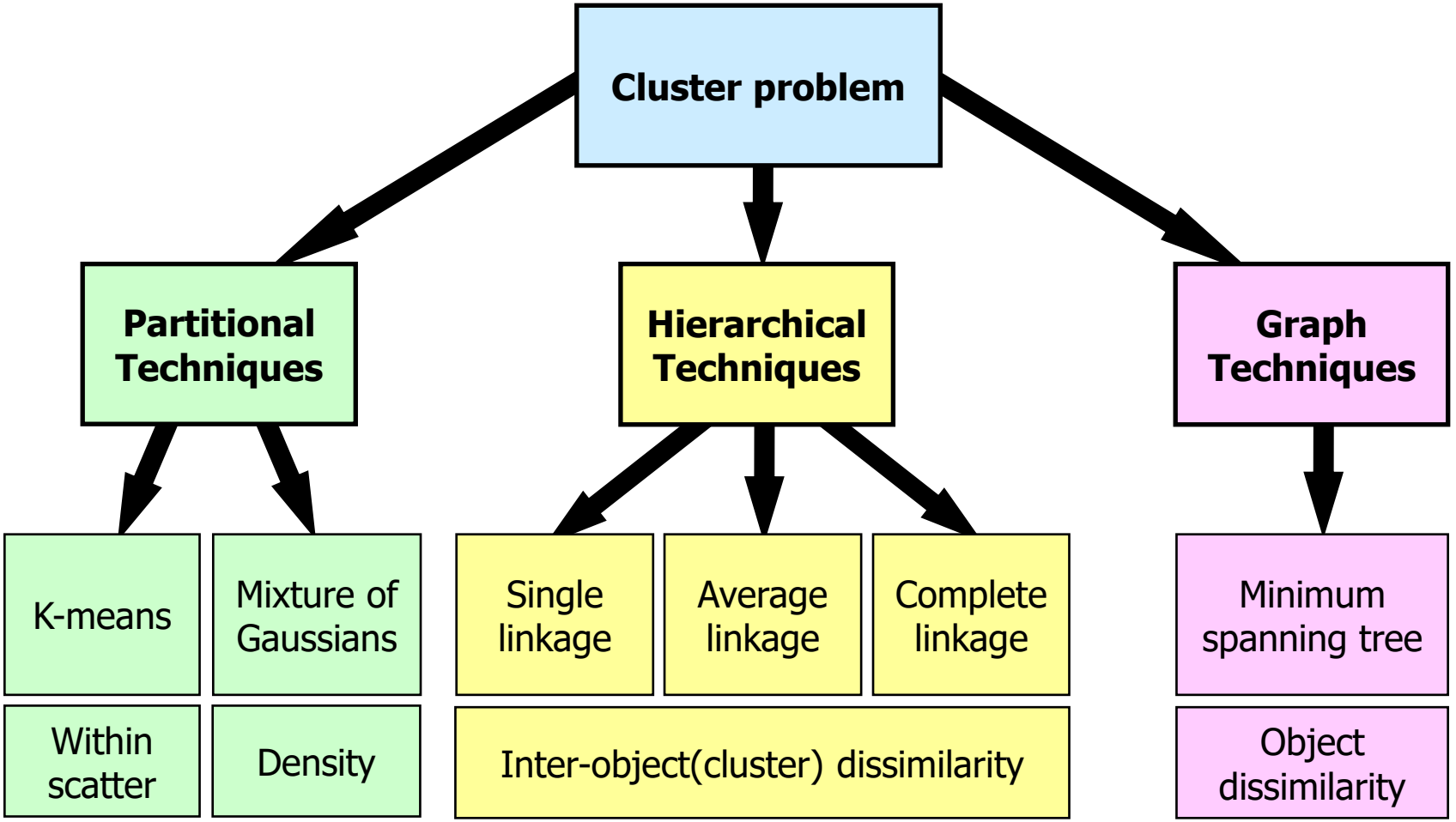
Absolute correlation
ignore amplitude & sign

$$1 - |\rho_{ij}|$$

d(● , ●) ≈ d(● , ●)
d(● , ●) ≈ d(● , ●)
d(● , ●) << d(● , ●)

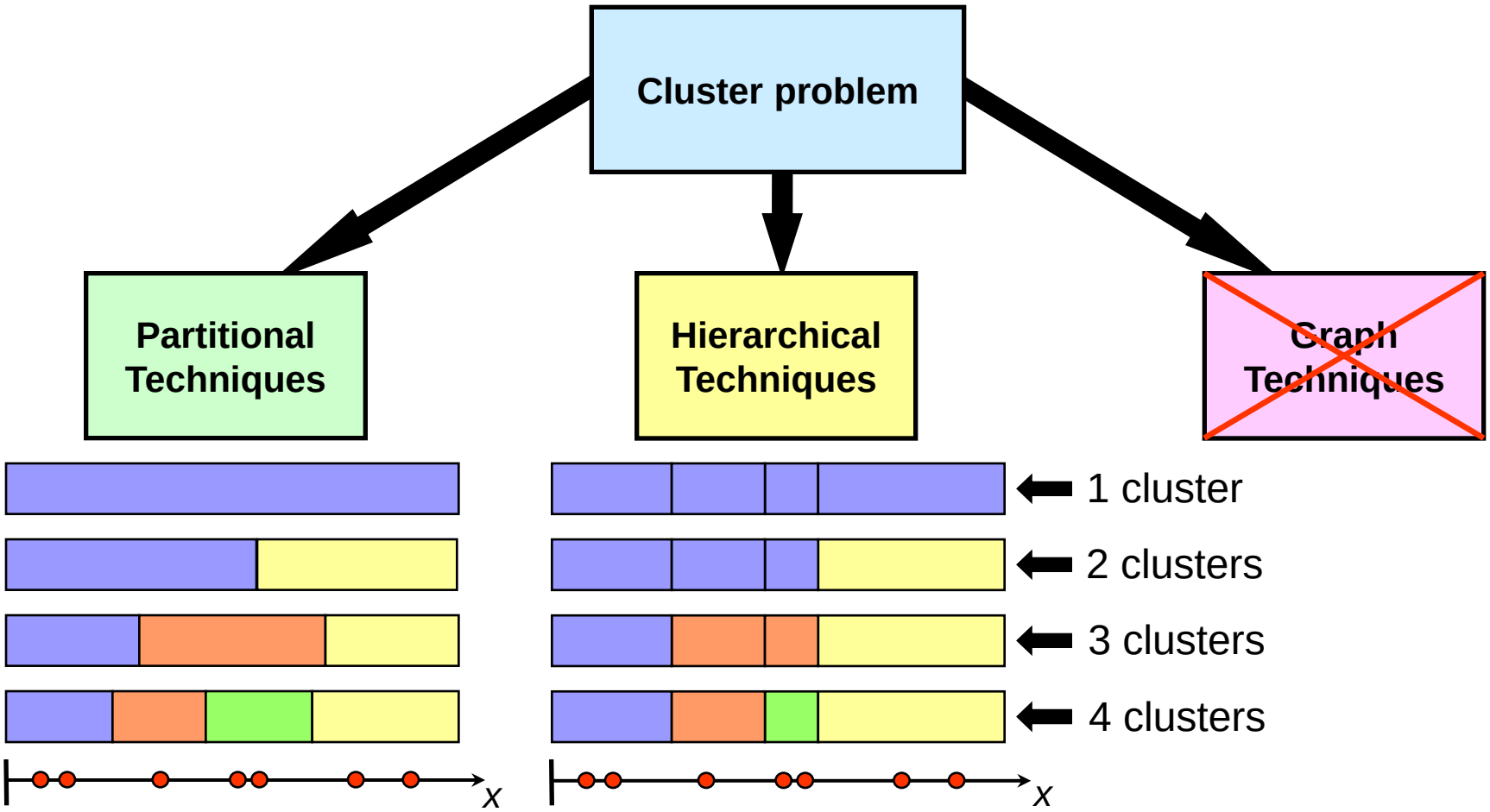
BioSB

Clustering techniques



BioSB

Clustering techniques (2)



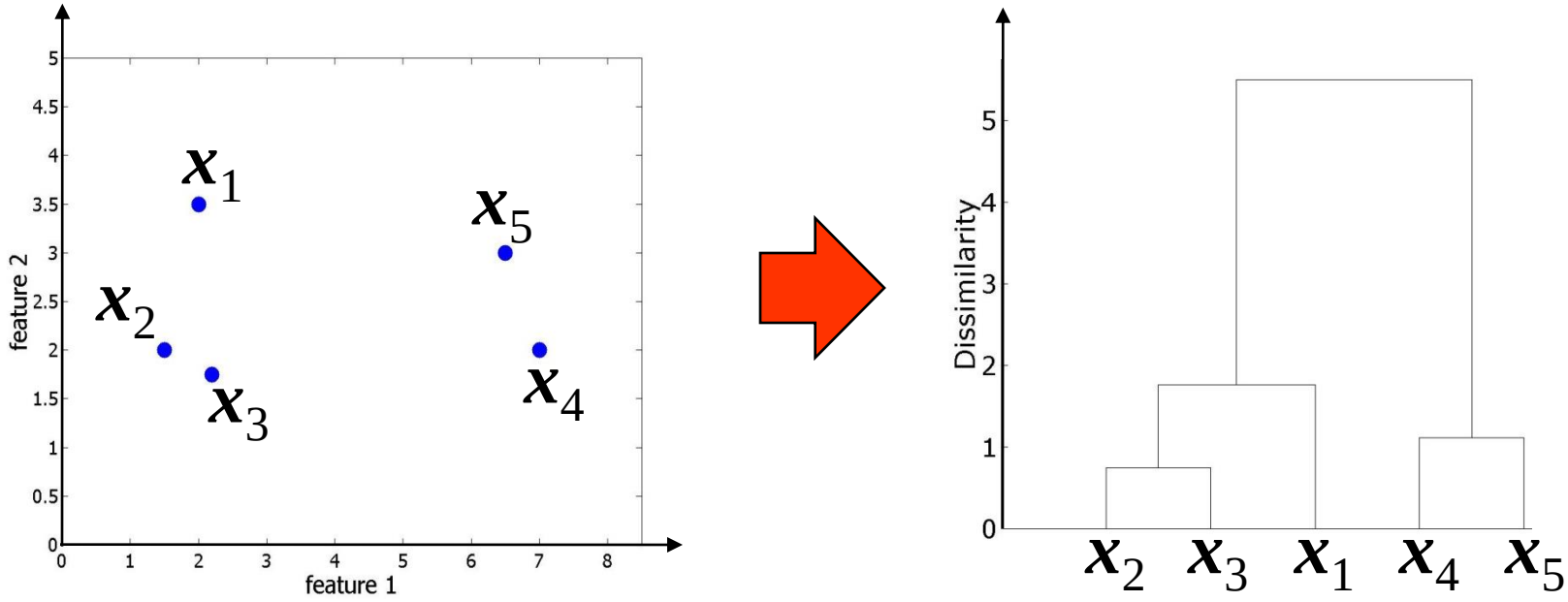
Hierarchical clustering

Input:

- dataset, ***X***: [*n* x *p*], or directly:
- dissimilarity matrix, ***D***: [*n* x *n*]
- linkage type

Output:

- dendrogram



Hierarchical clustering (2)

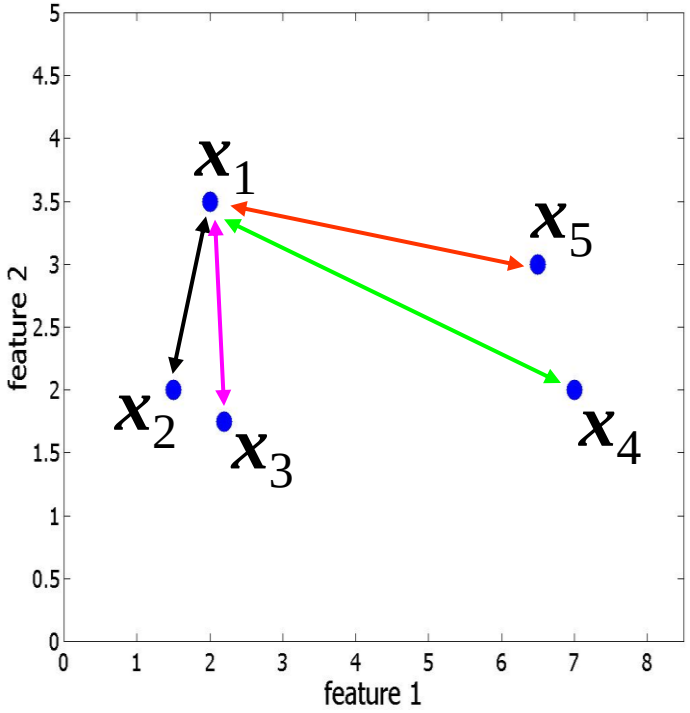
- **Algorithm** (agglomerative clustering)
  - Start: all objects of ***X*** in a separate cluster
  - Clustering: combine the 2 clusters with

    the shortest distance in dissimilarity matrix, ***D***
  - Distance between clusters is based on linkage type:
    - single, complete, average, …
  - Repeat until only 1 cluster is left

BioSB

Hierarchical clustering (3)

Dataset

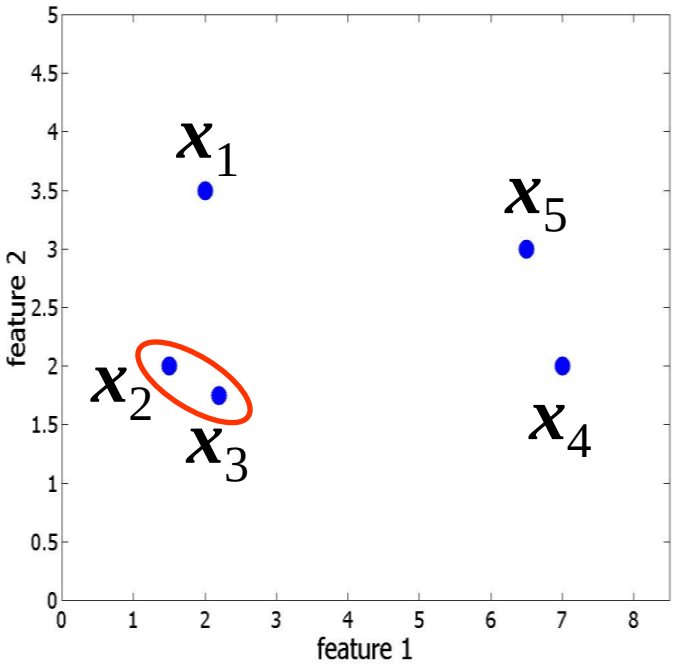


Euclidean distance matrix, ***D***

	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

Hierarchical clustering (4)

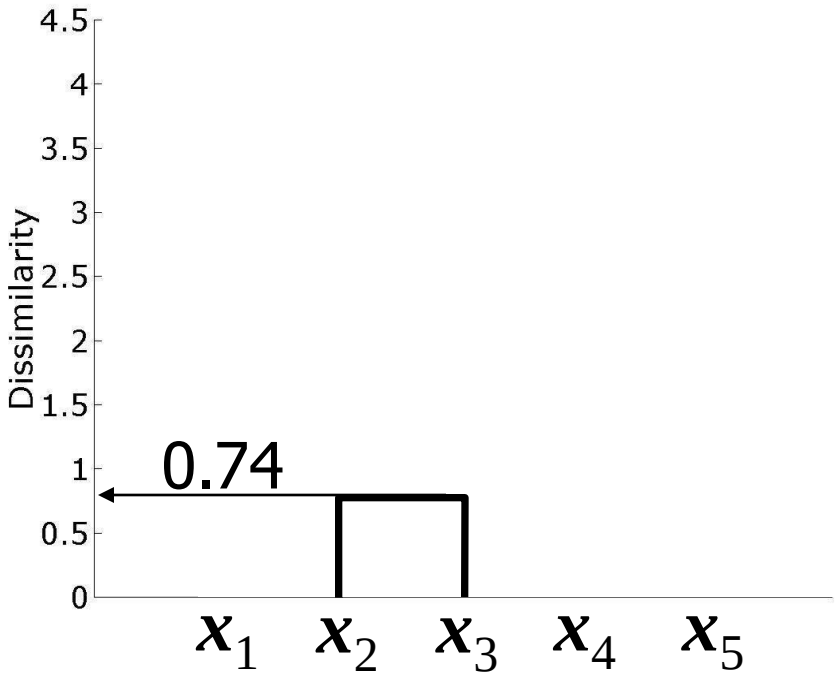
- **Step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\} = d(2,3)$



	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

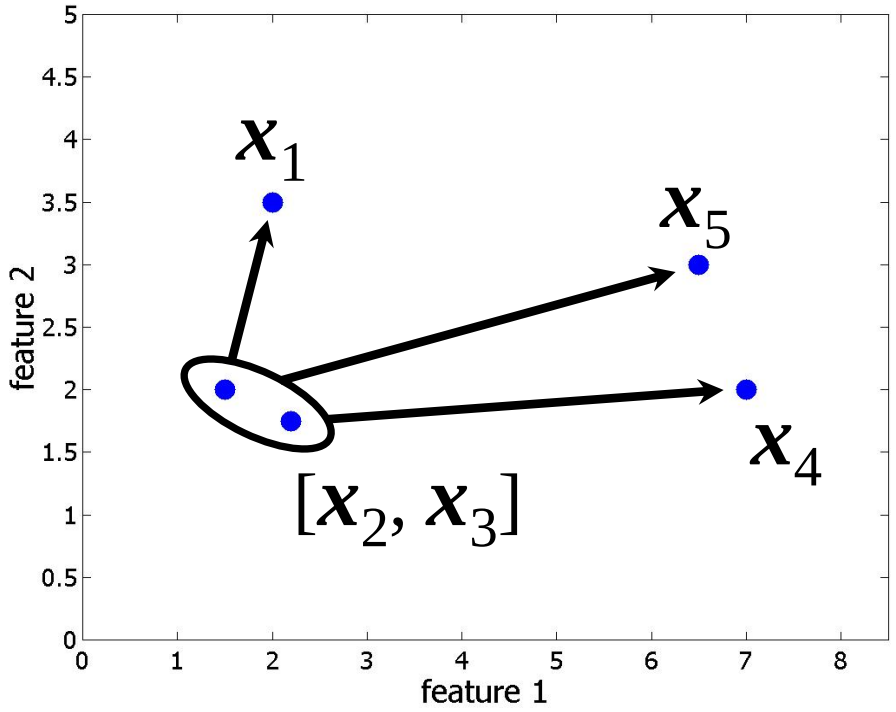
Hierarchical clustering (5)

- **Step 2:**
  Merge x_2 and x_3 into a single object, $[x_2, x_3]$;



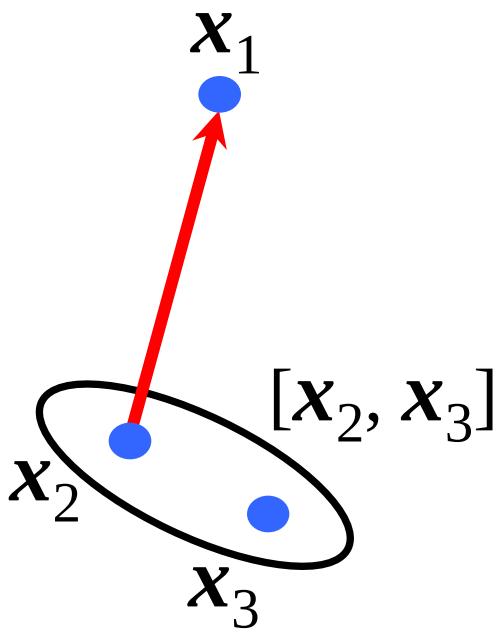
Hierarchical clustering (6)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  what is the distance between $[x_2, x_3]$ and the rest?



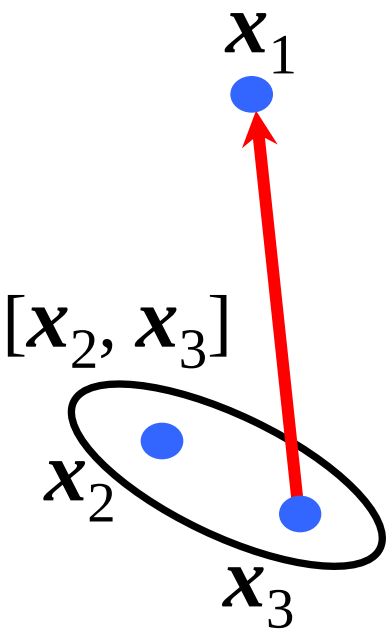
Hierarchical clustering (7)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  single linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \min(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



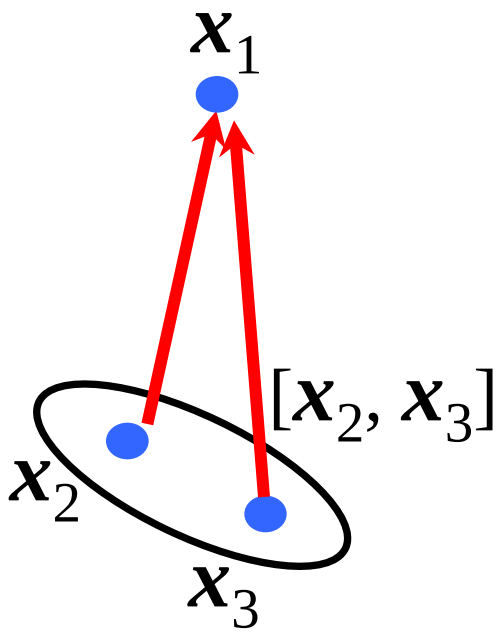
Hierarchical clustering (8)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  complete linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \max(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



Hierarchical clustering (9)

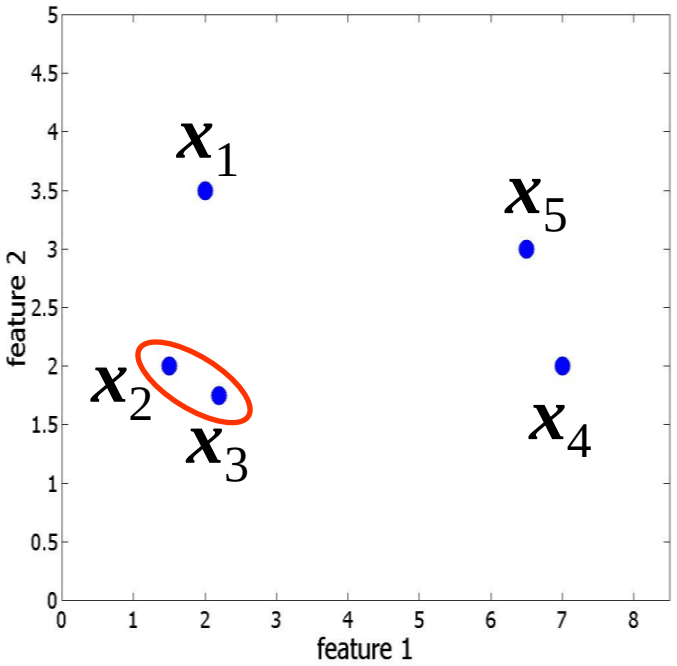
- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  average linkage: $d([x_2,x_3],x_1) = \text{mean}(d(x_1,x_2),d(x_1,x_3))$



BioSB

Hierarchical clustering (10a)

- **Step 3:**
  Recompute D – **single linkage:**



	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

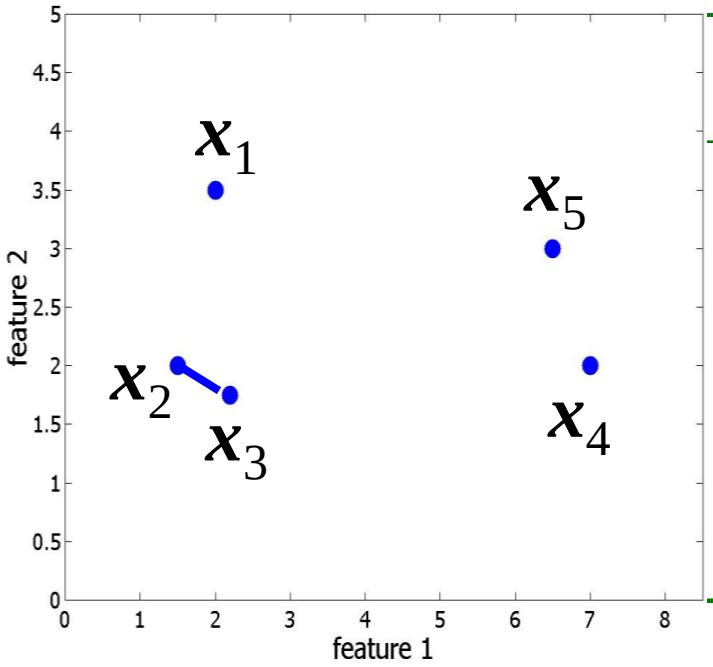
Hierarchical clustering (10b)

- **Step 3:**
  Recompute D – **single linkage:**

	x_1	$[x_2,x_3]$	x_4	x_5
x_1	0.00	1.58	5.22	4.53
$[x_2,x_3]$		0.00	4.81	4.48
x_4			0.00	1.12
x_5				0.00

Hierarchical clustering (11)

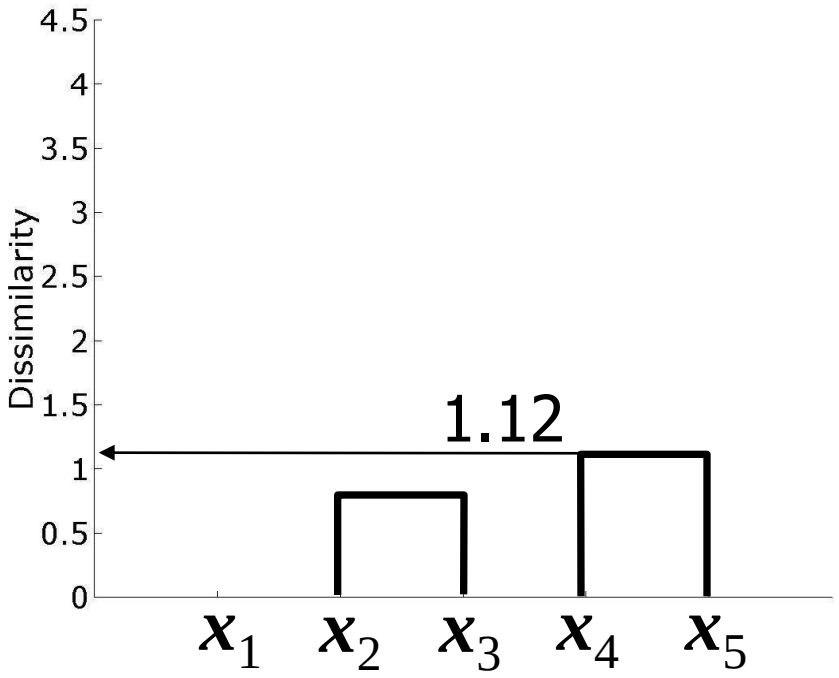
- **Repeat, step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\} = d(4,5)$



	x_1	$[x_2,x_3]$	x_4	x_5
x_1	0.00	1.58	5.22	4.53
$[x_2,x_3]$	1.58	0.00	4.81	4.48
x_4	5.22	4.81	0.00	1.12
x_5	4.53	4.48	1.12	0.00

Hierarchical clustering (12)

- **Repeat, step 2:**
  Merge x_4 and x_5 into a single object, $[x_4,x_5]$;



BioSB

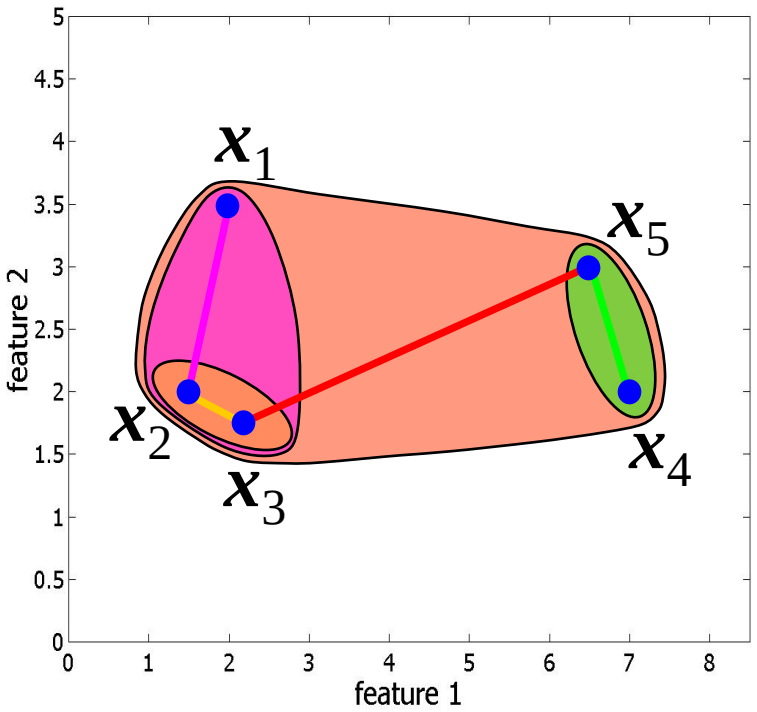
Hierarchical clustering (13)

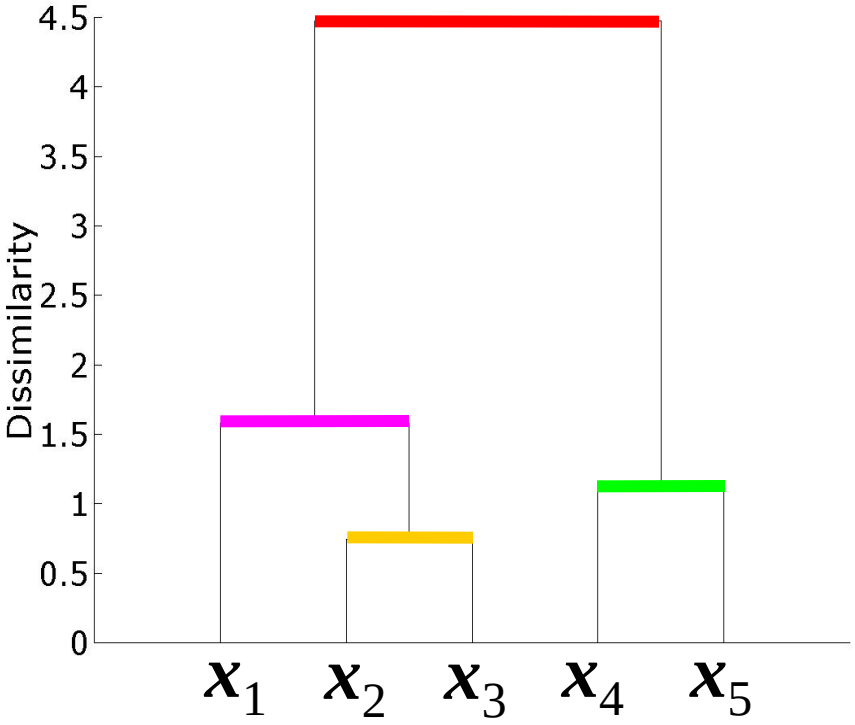
- **Repeat, step 3:**
  Recompute $\boldsymbol{D}$ (single linkage):

	x_1	$[x_2,x_3]$	$[x_4,x_5]$
x_1	0.00	1.58	4.53
$[x_2,x_3]$		0.00	4.48
$[x_4,x_5]$			0.00

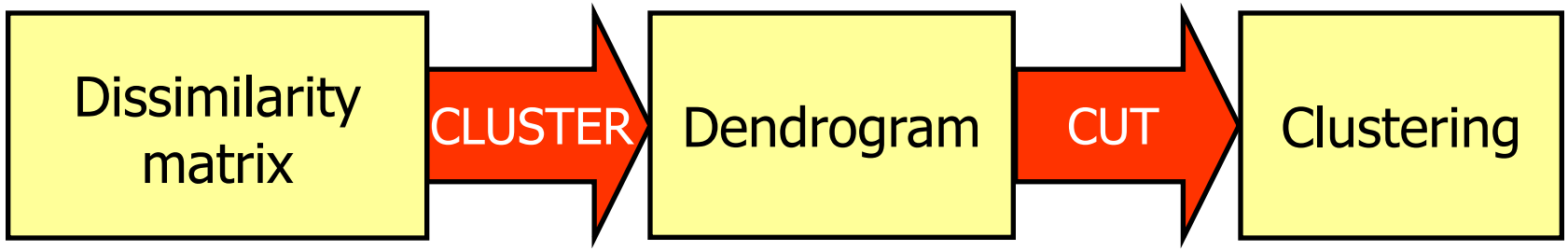
Hierarchical clustering (14)

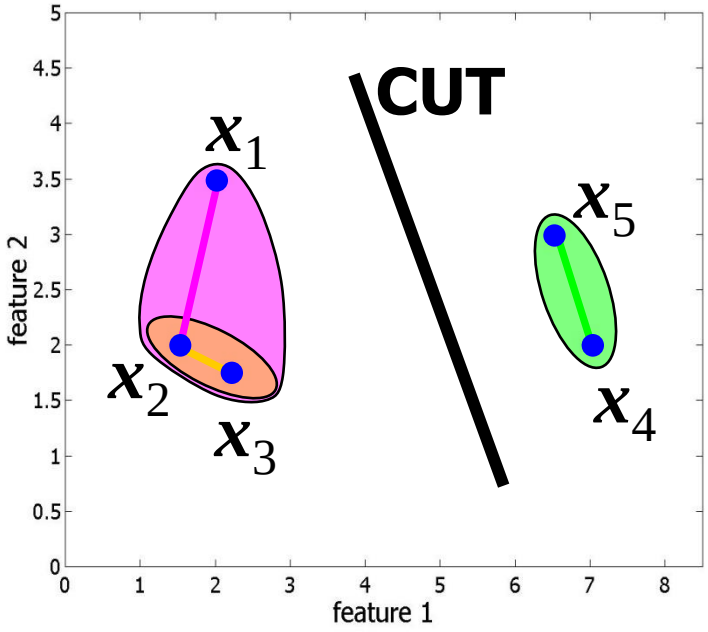
- Repeat steps 1-3 until a single cluster remains

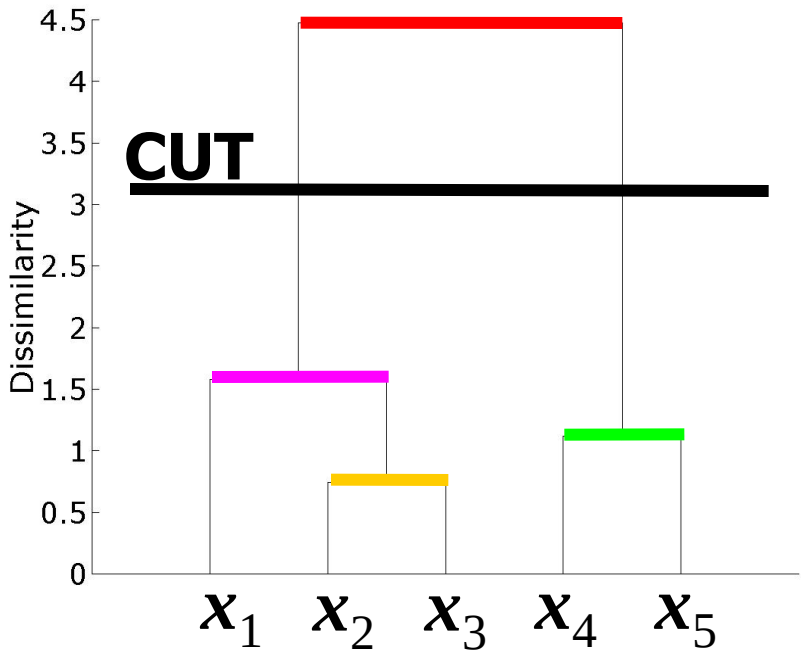




Hierarchical clustering (15)



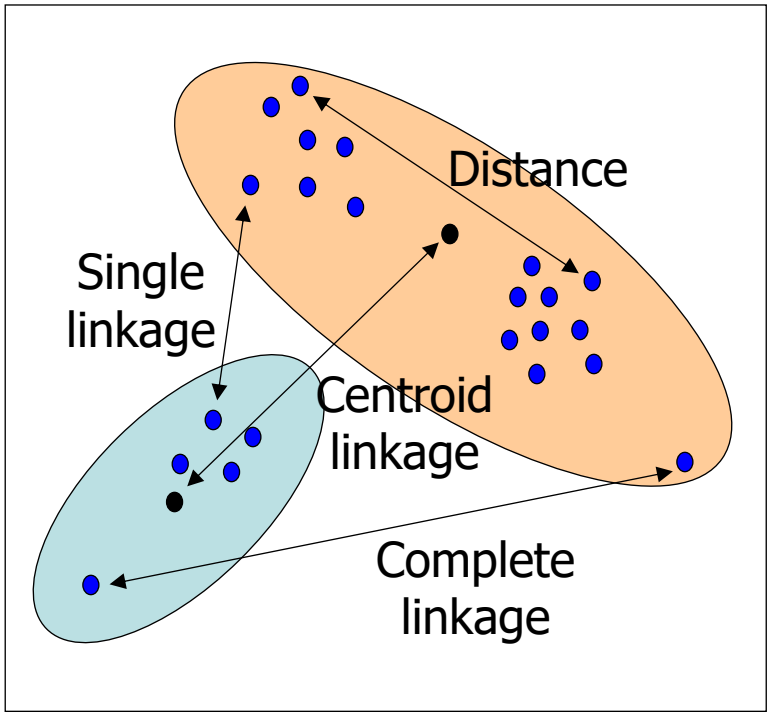




Hierarchical clustering (16)

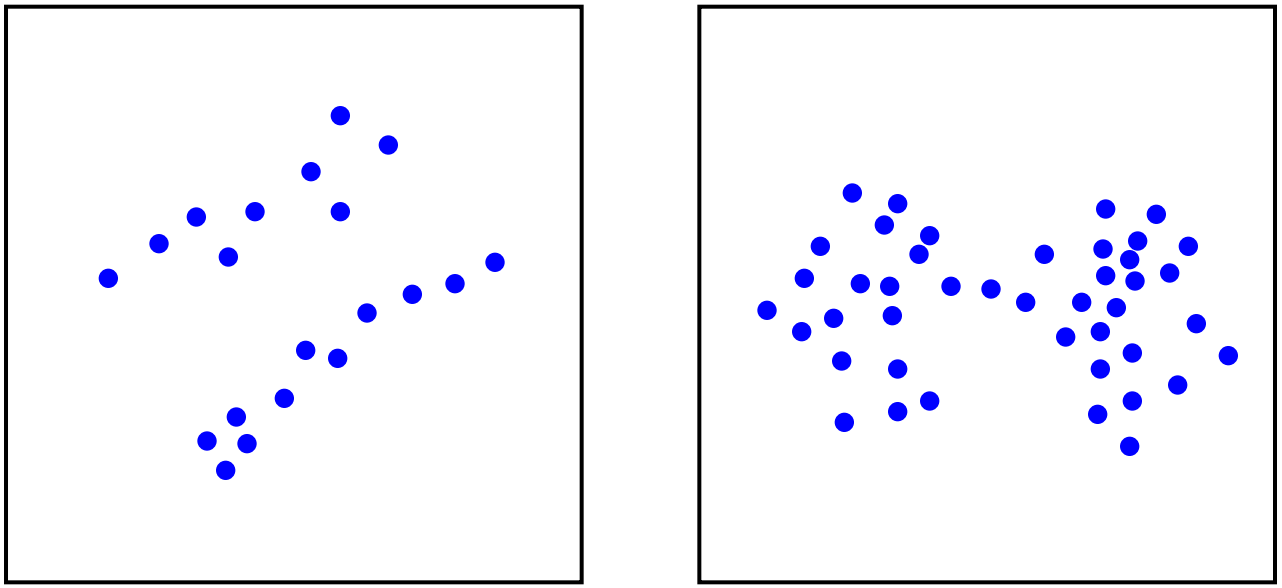
- Hierarchical clustering: repeatedly group closest clusters

- Important choices:
  - *Distance measure*
    between objects:
    Euclidean, correlation,
    Hamming, Minkowski, ...
  - *Linkage*
    between clusters:
    single, average (centroid), complete



BioSB

Linkage and cluster shape

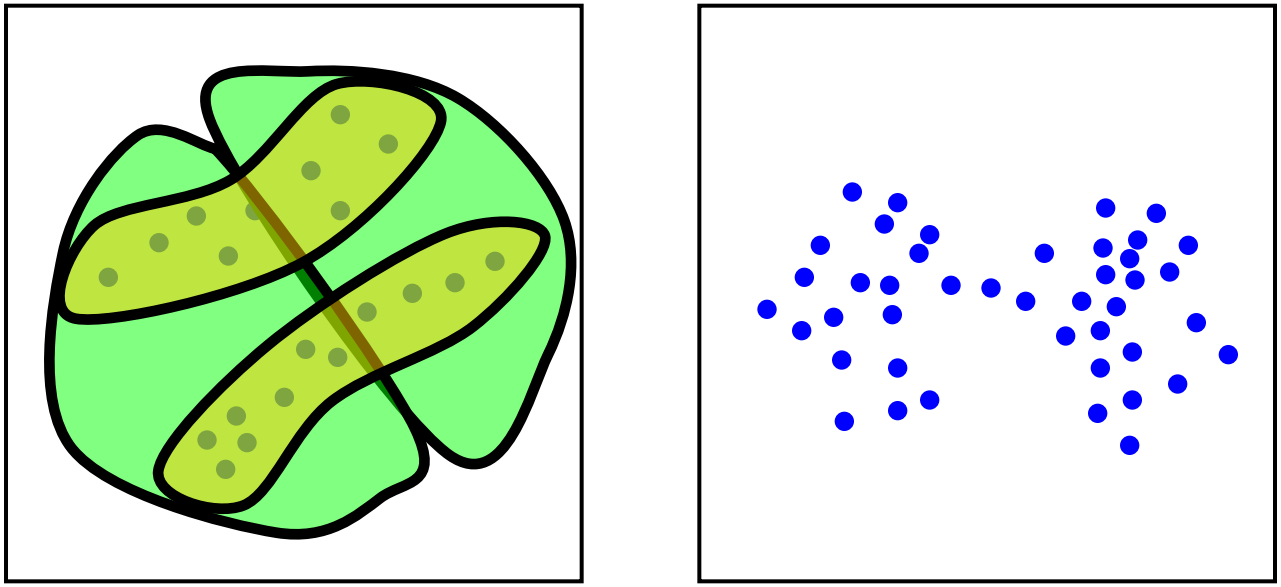


Complete linkage

Single linkage

BioSB

Linkage and cluster shape (2)

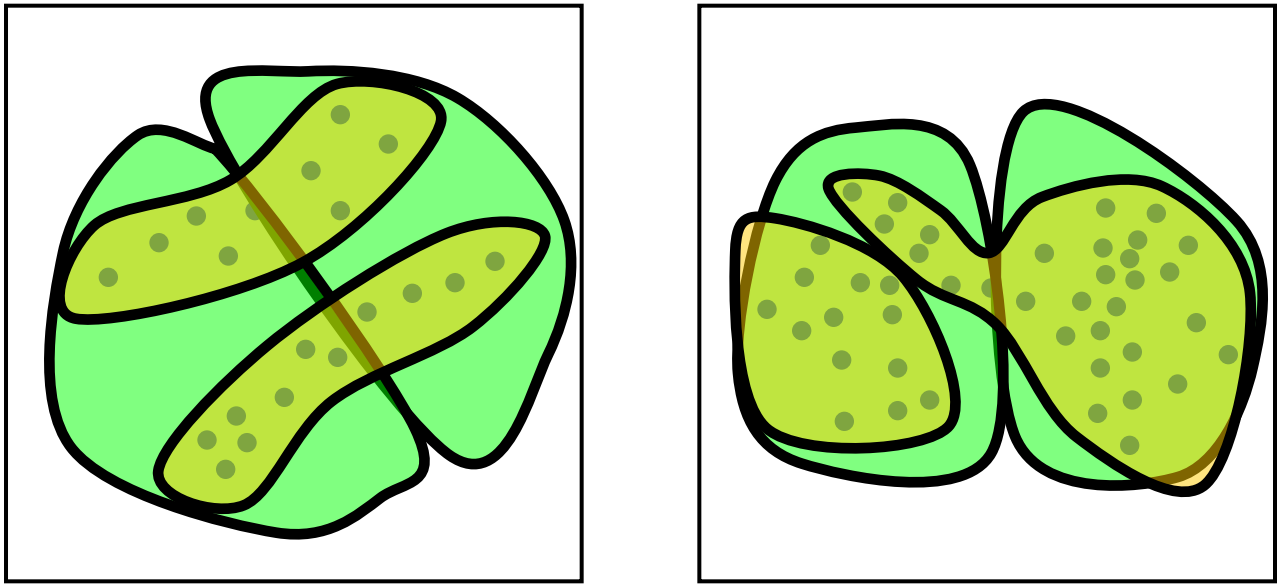


Complete linkage

Single linkage

BioSB

Linkage and cluster shape (3)



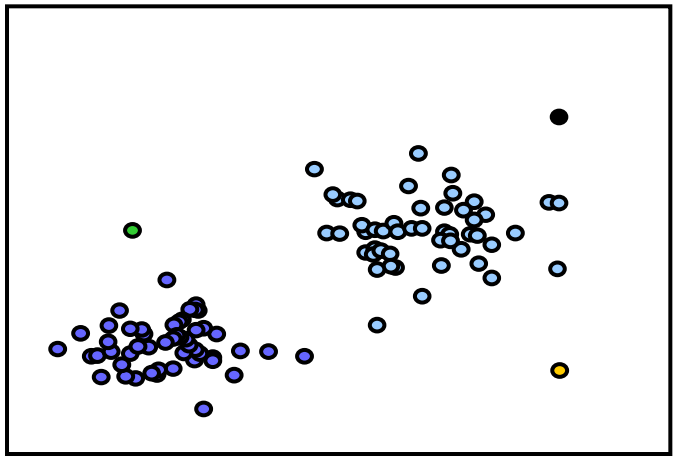
Complete linkage

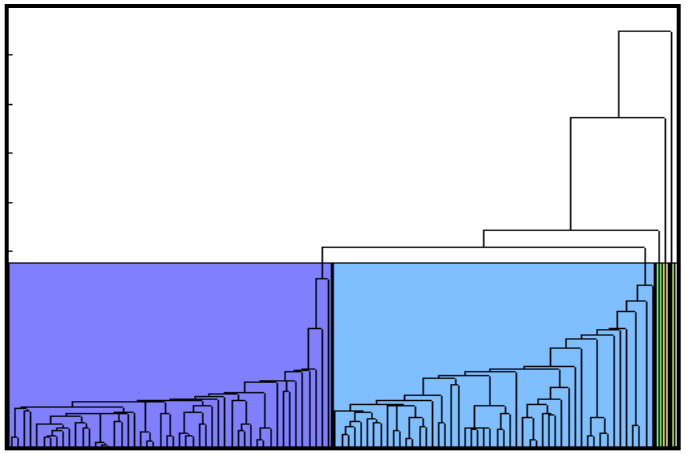
Single linkage

BioSB

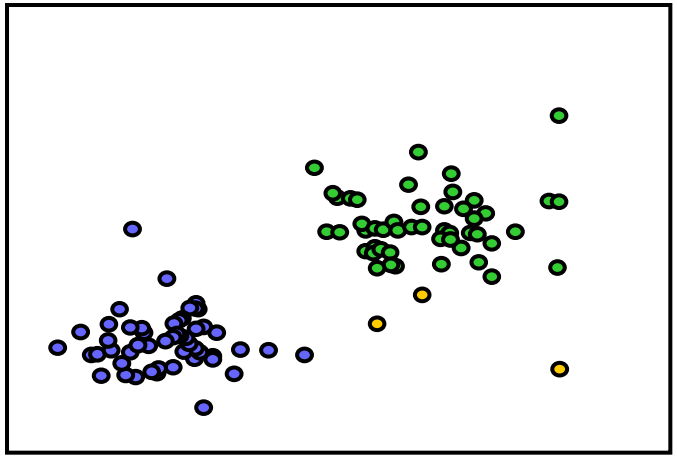
Linkage and outliers

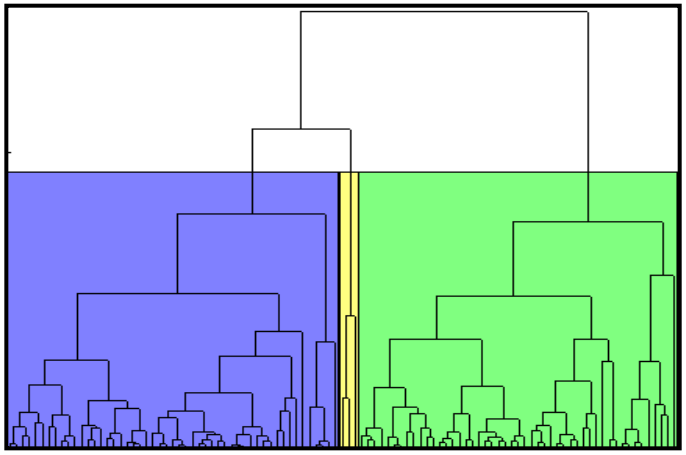
Single linkage





Complete linkage

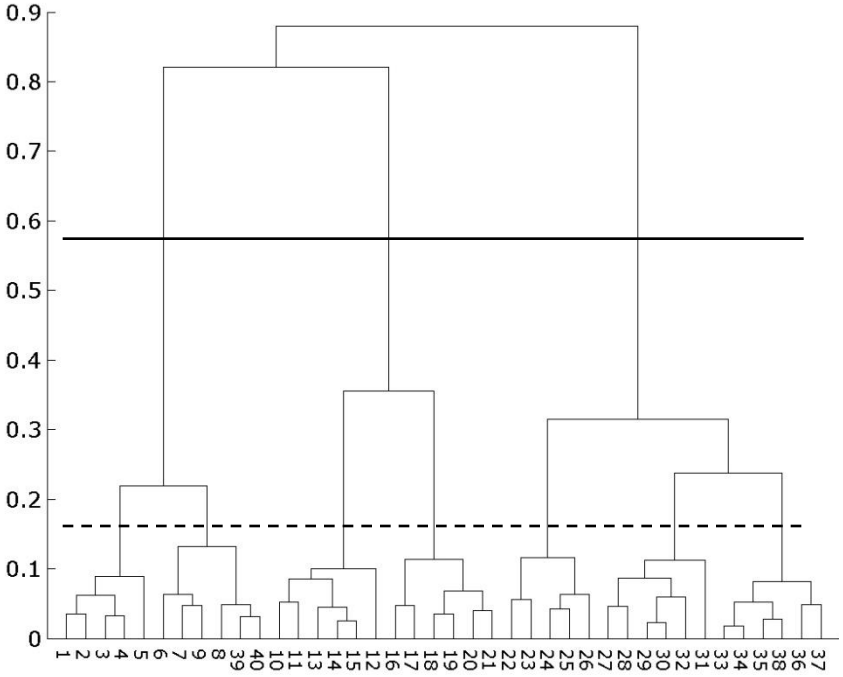


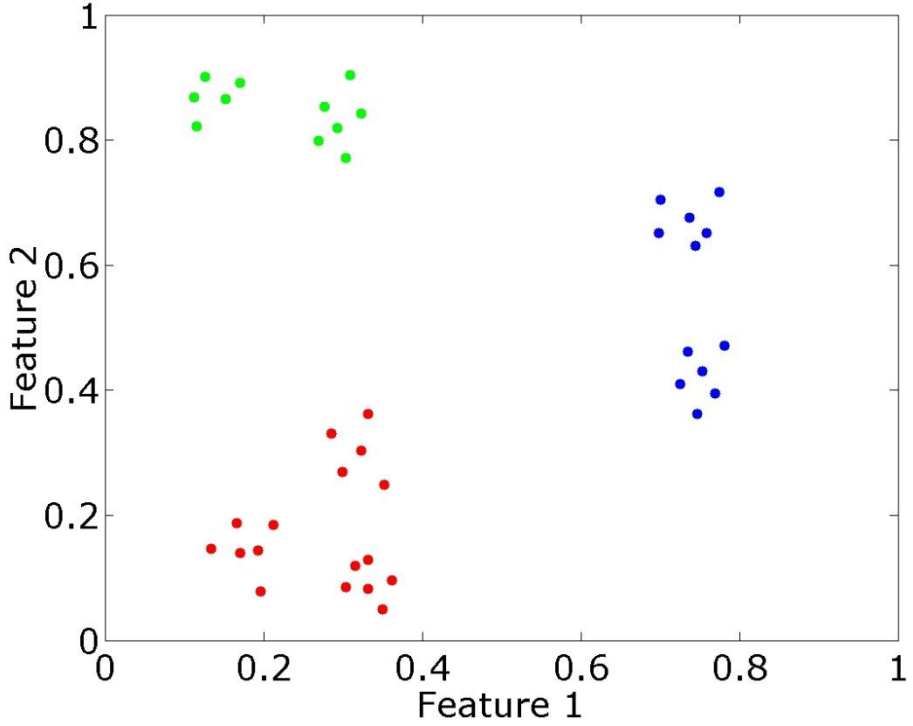


BioSB

Hierarchical clustering examples

Euclidean, complete linkage

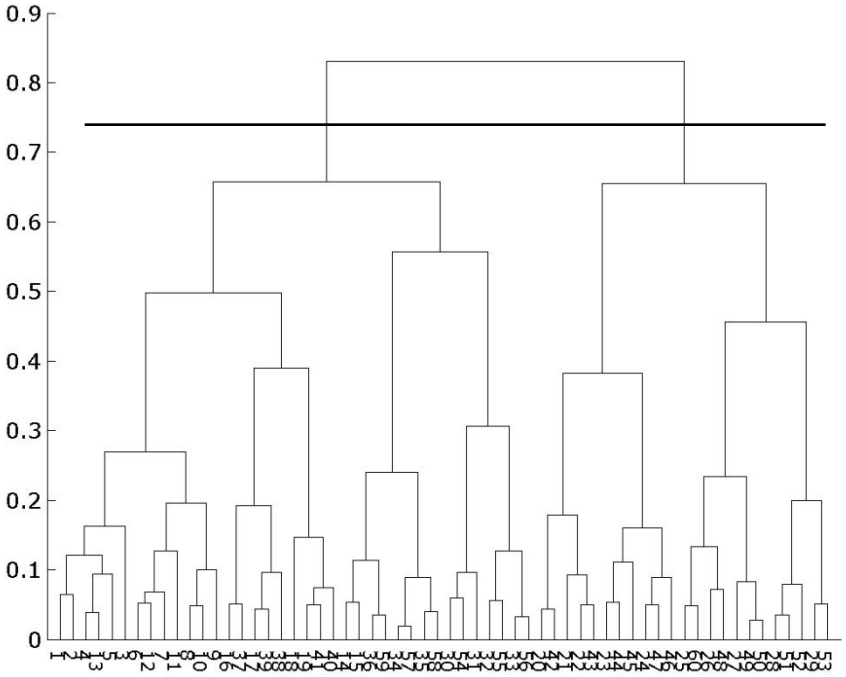


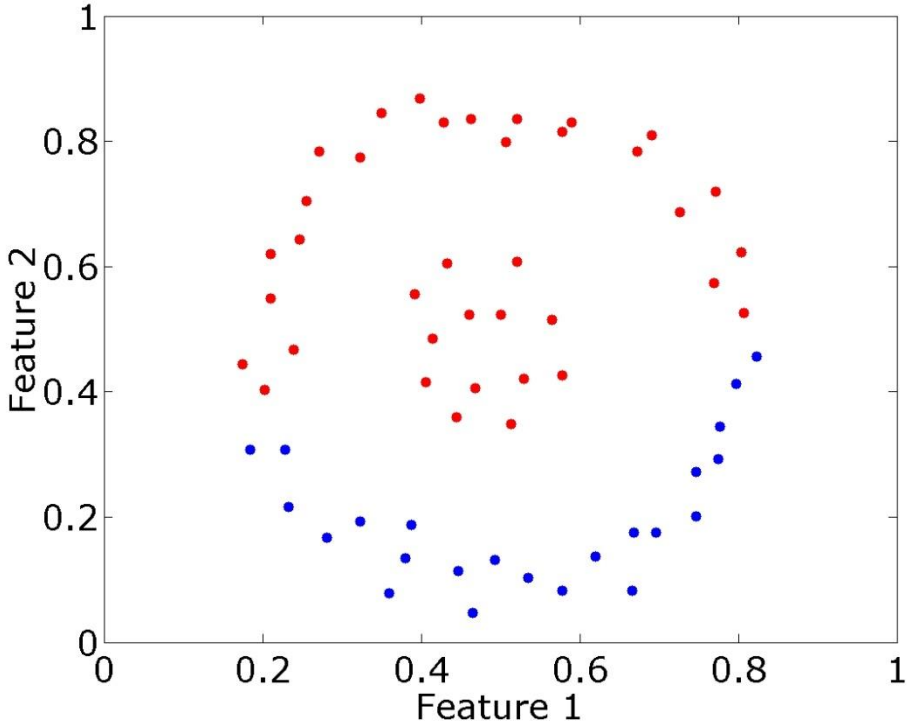


BioSB

Hierarchical clustering examples (2)

Euclidean, complete linkage

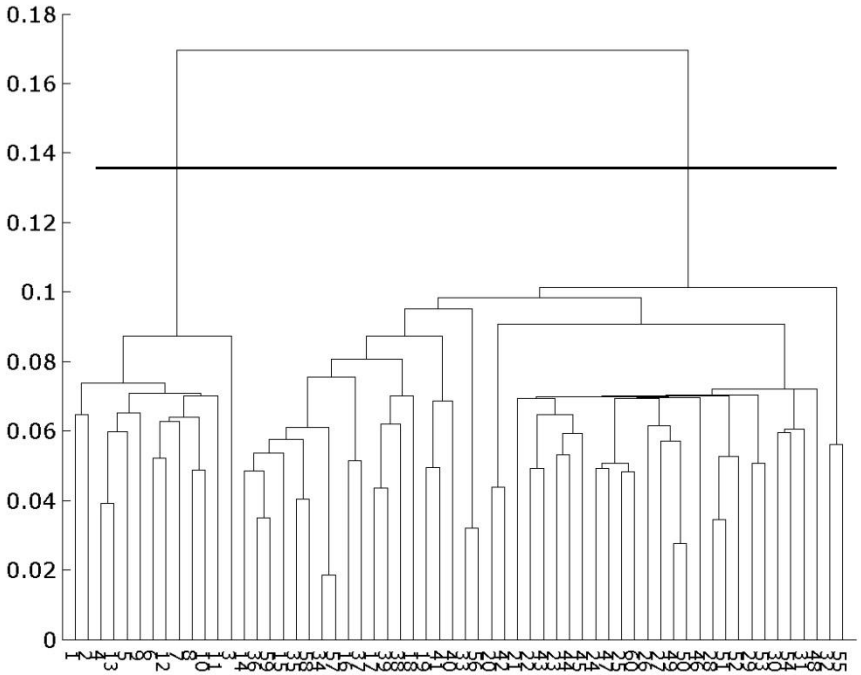


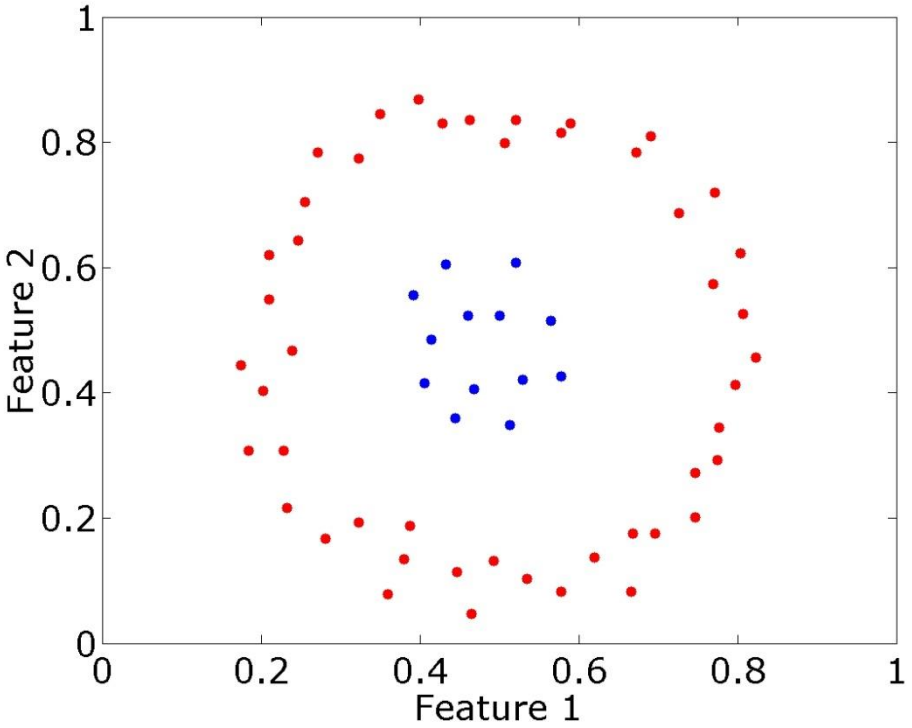


BioSB

Hierarchical clustering examples (3)

Euclidean, single linkage





Hierarchical clustering (17)

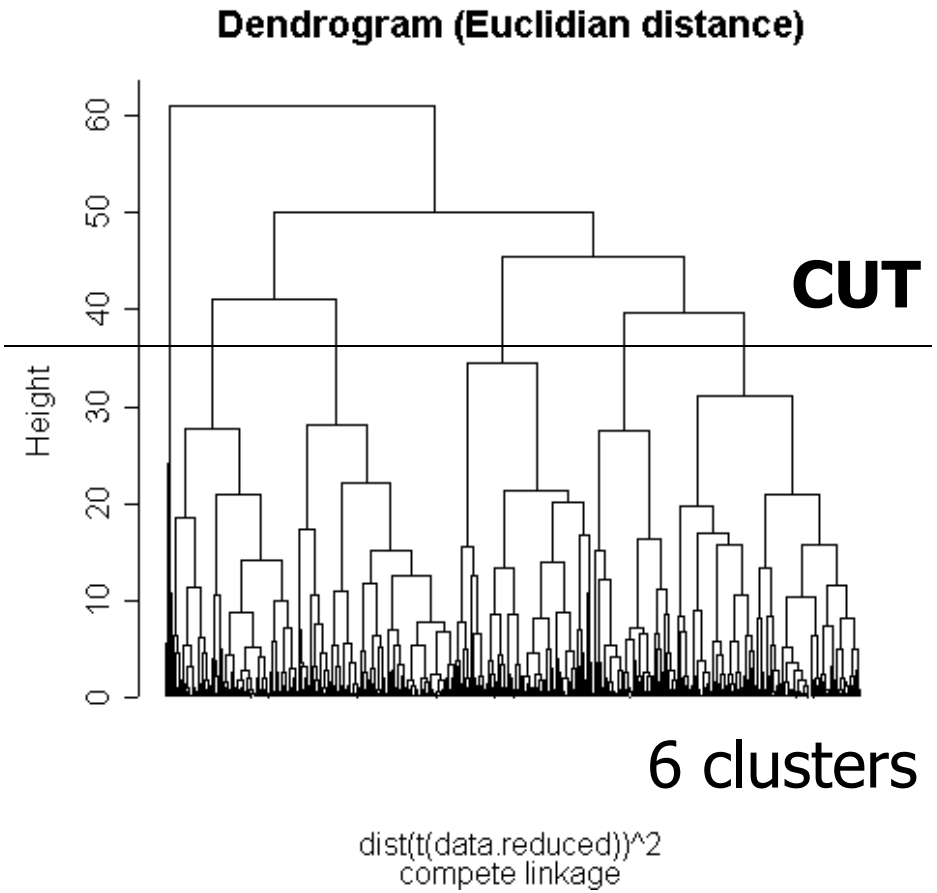
- Advantages:
  - dendrogram gives overview of all possible clusterings
  - linkage type allows to find clusters of varying shapes (convex and non-convex)
  - different dissimilarity measures can be used

- Disadvantages:
  - computationally intensive:
    $O(n^2)$ in complexity and memory
  - clusterings limited to “hierarchical nestings”

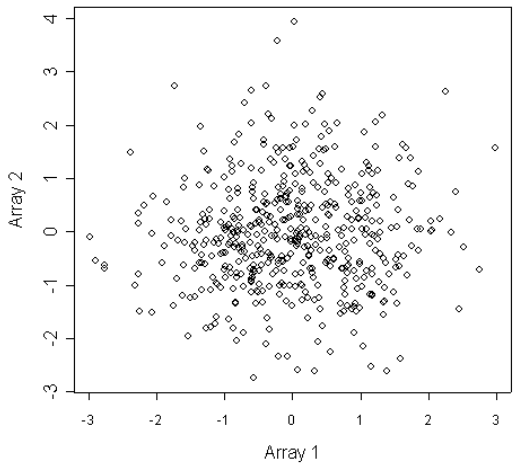
BioSB

Hierarchical clustering: warning

- Cluster 500 genes, 5 arrays:



Data were random …



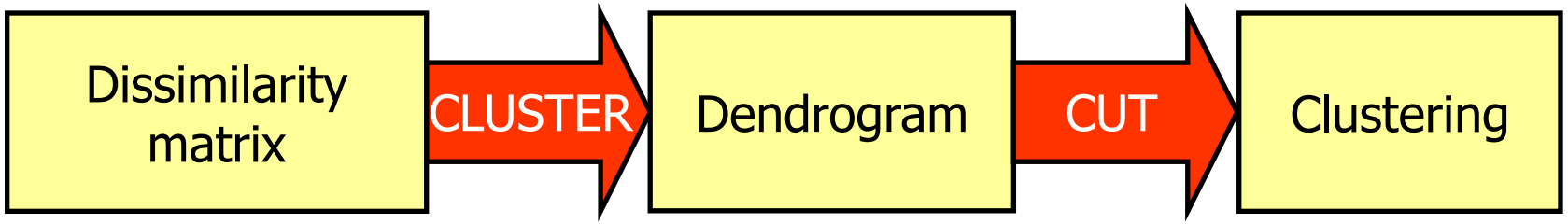
Validation is needed

BioSB

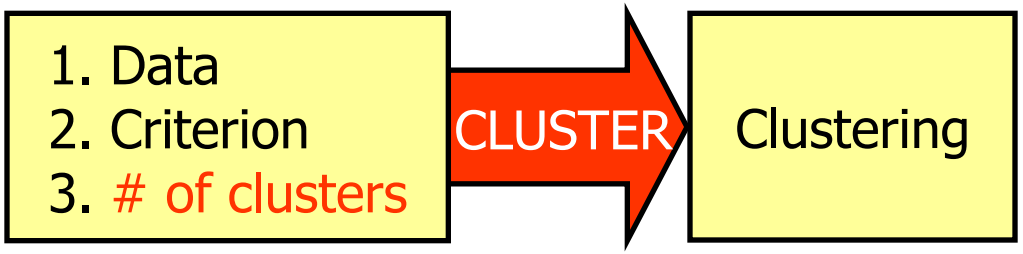
Exercise 2.14-2.19

Sum-of-squares clustering

- Hierarchical:



- Sum-of-squares:



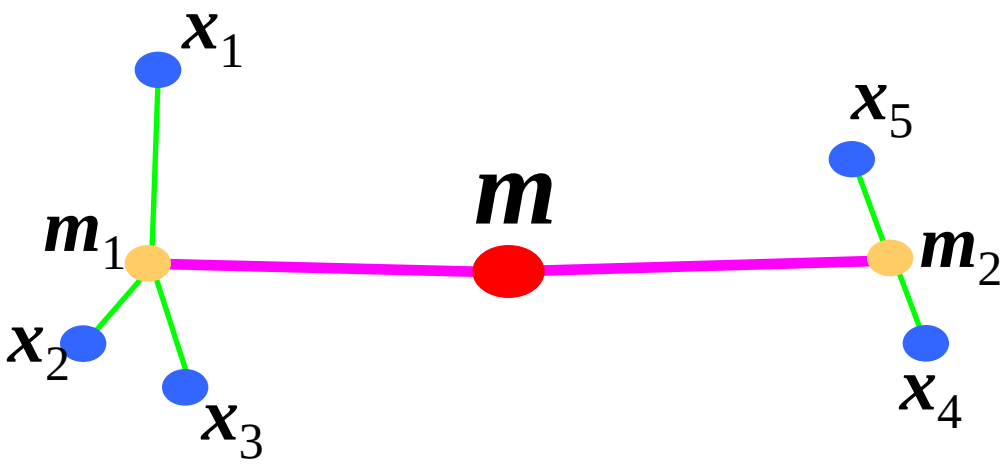
BioSB

Sum-of-squares clustering (2)

- Within and between scatter:

$$\boldsymbol{S}_w = \sum_{i=1}^{C} \frac{n_i}{n} \Sigma_i \quad (n_1 = 3, n_2 = 2, n = 5, C = 2)$$

$$\boldsymbol{S}_B = \sum_{i=1}^{C} \frac{n_i}{n} (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T, \ \boldsymbol{m} = \sum_{i=1}^{C} \frac{n_i}{n} \boldsymbol{m}_i$$



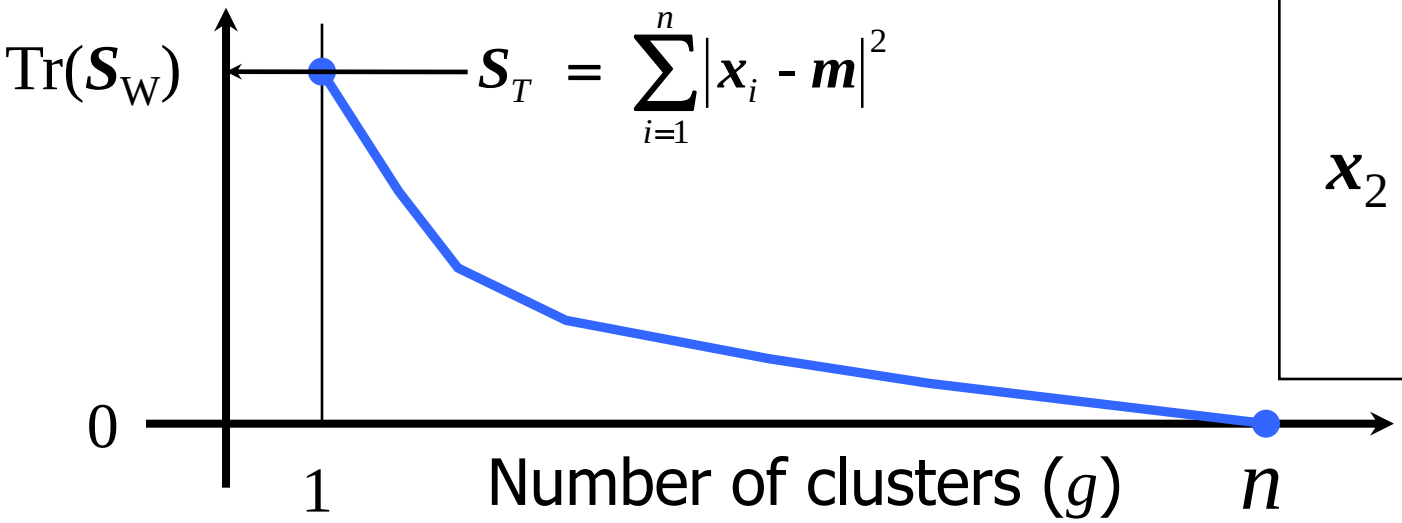
BioSB

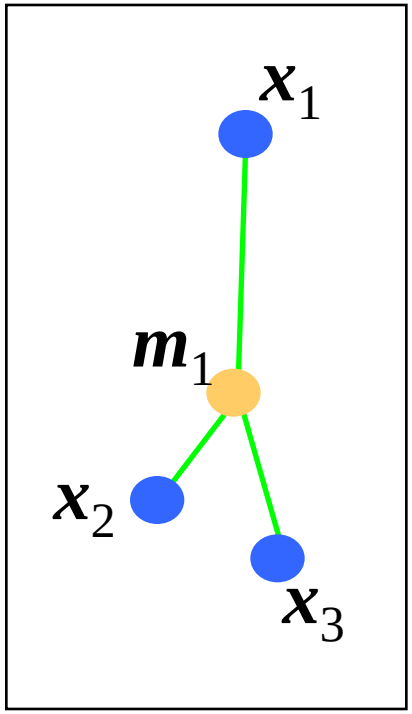
K-means

- Minimize: $\mathrm{Tr}(\mathbf{S}_W) = \frac{1}{n}\sum_{j=1}^{g}\mathbf{S}_j$

$$\boldsymbol{S}_j = \sum_{i=1}^{n_j}\left|\boldsymbol{x}_i - \boldsymbol{m}_j\right|^2$$

(sum of per cluster variances)





K-means (2)

- Iterative procedure to search for $\min(\mathrm{Tr}(\boldsymbol{S}_\mathrm{W}))$:
  1. choose number of clusters (g)
  2. position prototypes (m_j, $j=1,\ldots,g$) randomly
  3. assign samples to closest prototype
  4. compute mean of samples assigned to same prototype: new prototype position

  Repeat steps 3 and 4 as long as prototypes move

BioSB

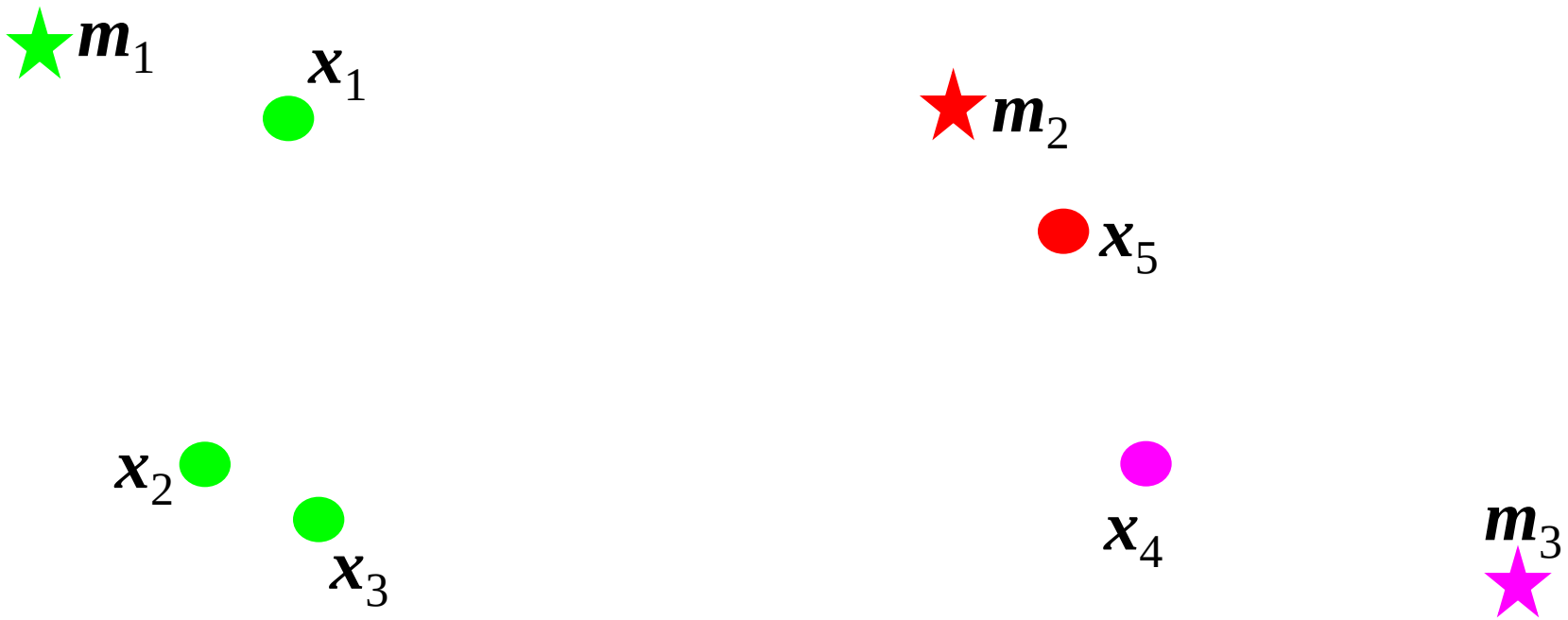
K-means (3)

- **Step 1:** Choose number of clusters/prototypes
- **Step 2:** Position prototypes randomly

BioSB

K-means (4)

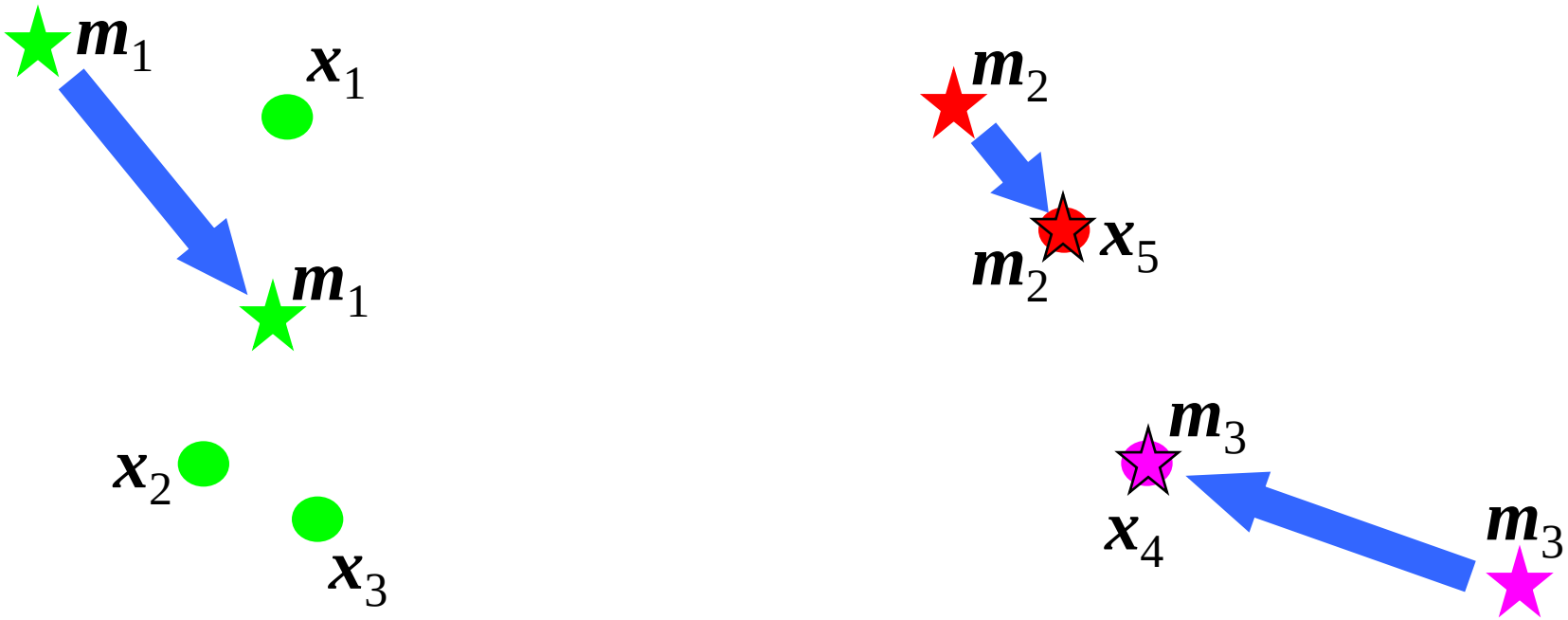
- **Step 3:** Assign samples to closest prototype



BioSB

K-means (5)

- **Step 4:** Compute mean of samples assigned to same prototype: new prototype positions



BioSB

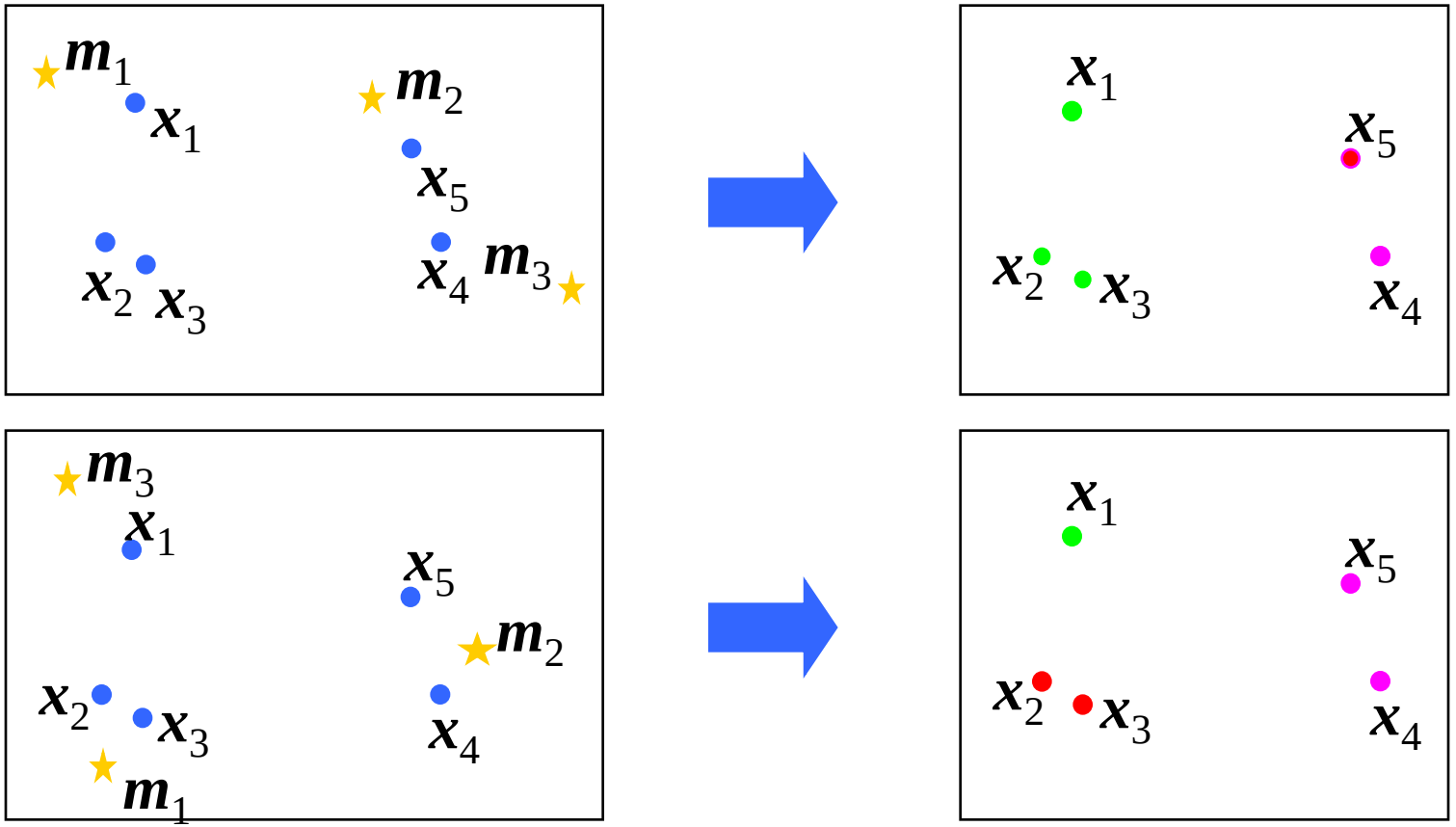
K-means (6)

- **Repeat** as long as prototype positions change**:**
  - **Step 3:** Assign samples
  - **Step 4:** Recompute prototype positions

BioSB

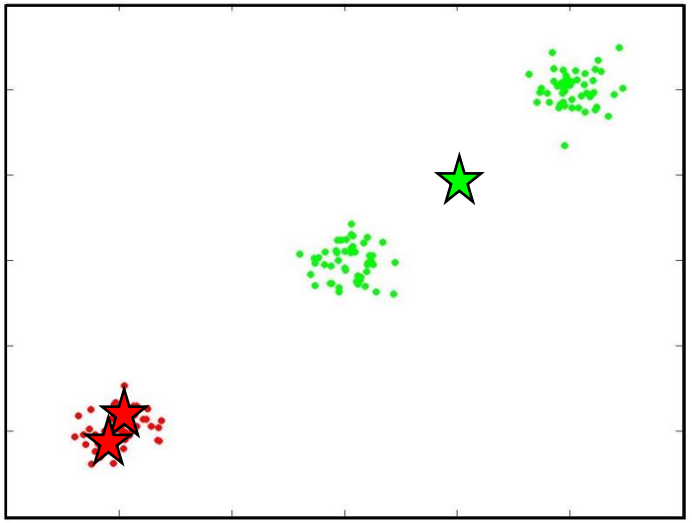
K-means problems

- Clustering depends on initialization



K-means problems (2)

- Algorithm can get stuck in local minima

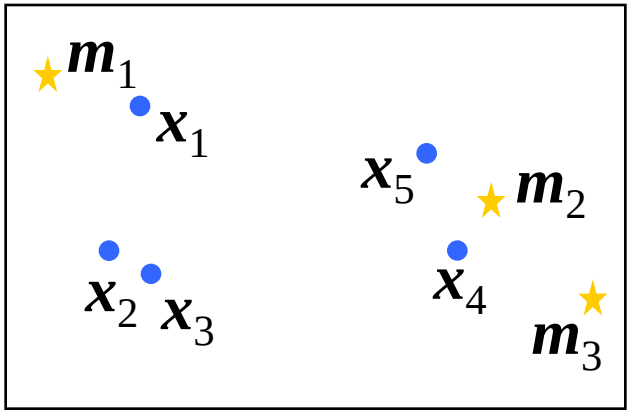


- Solution:
  - start from I different random initialisations
  - keep the best clustering (lowest $\mathrm{Tr}(\boldsymbol{S}_\mathrm{W})$)
  - For high-dimensional data, many restarts can be necessary (e.g. I = 100)

BioSB

K-means problems (3)

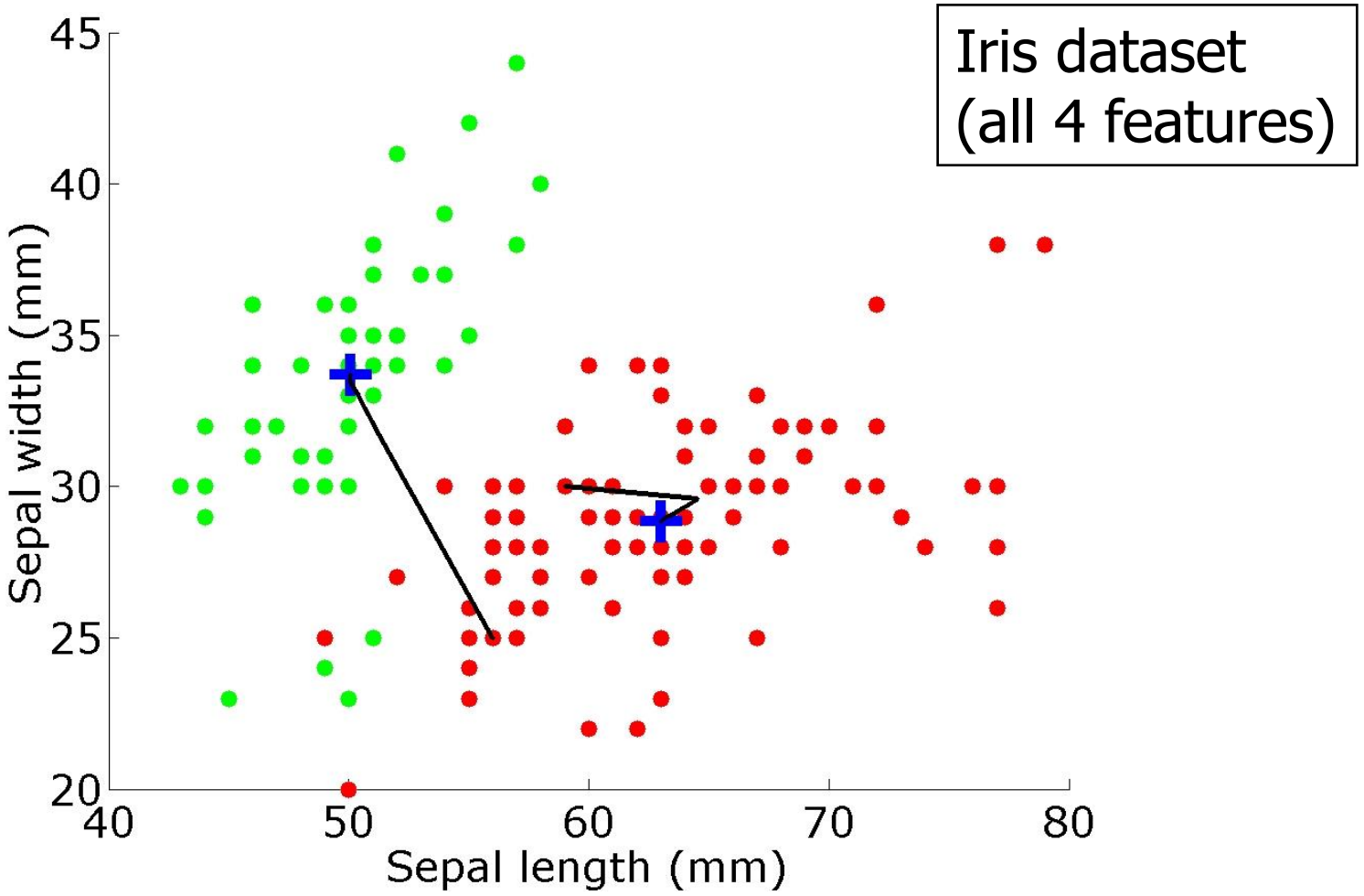
- Clusters can loose all samples



- Possible solution:
  - remove cluster and continue with $g - 1$ means
  - alternatively, split largest cluster into two
    or add a random cluster to continue with g means

BioSB

K-means example



BioSB

Advantages/disadvantages: *K*-means

- Disadvantages:
  - Finds only convex clusters (“round shapes”)
  - Sensitive to initialization
  - Can get stuck in local minima

- Advantages:
  - Very simple
  - Fast

BioSB

Recapitulation

- Clustering is way to detect *natural* groups in data
- What is natural is partly subjective
- We looked at:
  - *Hierarchical clustering*
  - *Sum of squares* (k-means) clustering
- Hierarchical clustering:
  - *dendrogram* shows a complete hierarchy of possible clusterings
  - computionally intensive
- *K*-means
  - fast
  - sensitive to *initialization* and *local minima*

BioSB

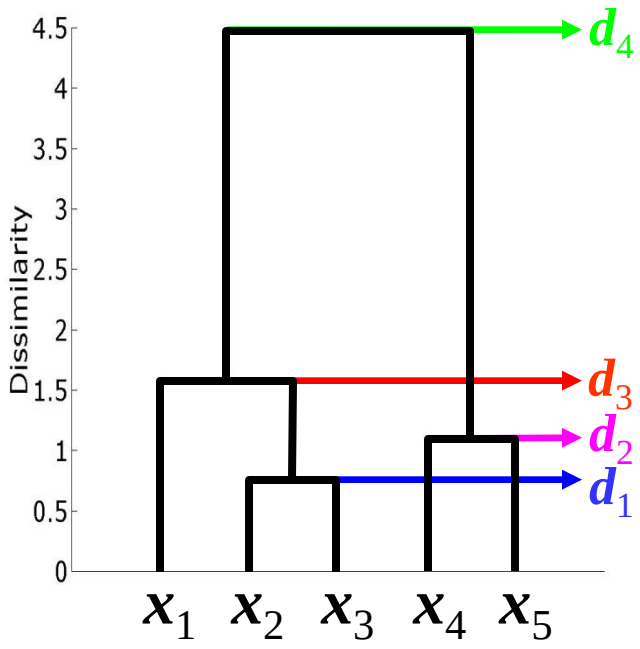
Cluster validation

- Cluster validation:
  - Checking whether grouping is really present
  - Choosing the optimal number of clusters
- A difficult problem – the ground truth is not known (since we do not know the object labels)!
- Methods:
  - Distortion measures:
    - Does clustering approximate structure in data?
  - Validity measures:
    - Davies-Bouldin index
  - Fusion graph
  - Gap statistic

BioSB

Distortion measures

- How well does a dendrogram capture structure in data?



d^*	x_1	x_2	x_3	x_4	x_5
x_1	0	d_3	d_3	d_4	d_4
x_2		0	d_1	d_4	d_4
x_3			0	d_4	d_4
x_4				0	d_2
x_5					0

Distortion measures (2)

- Measure of distortion: Pearson correlation of $\boldsymbol{d}$ and $\boldsymbol{d}^*$

$$\rho(\boldsymbol{d},\boldsymbol{d}^*) = \frac{\mathrm{cov}(\boldsymbol{d},\boldsymbol{d}^*)}{\sqrt{\mathrm{var}(\boldsymbol{d})\mathrm{var}(\boldsymbol{d}^*)}} \in [-1,1]$$

$\boldsymbol{d}$

	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

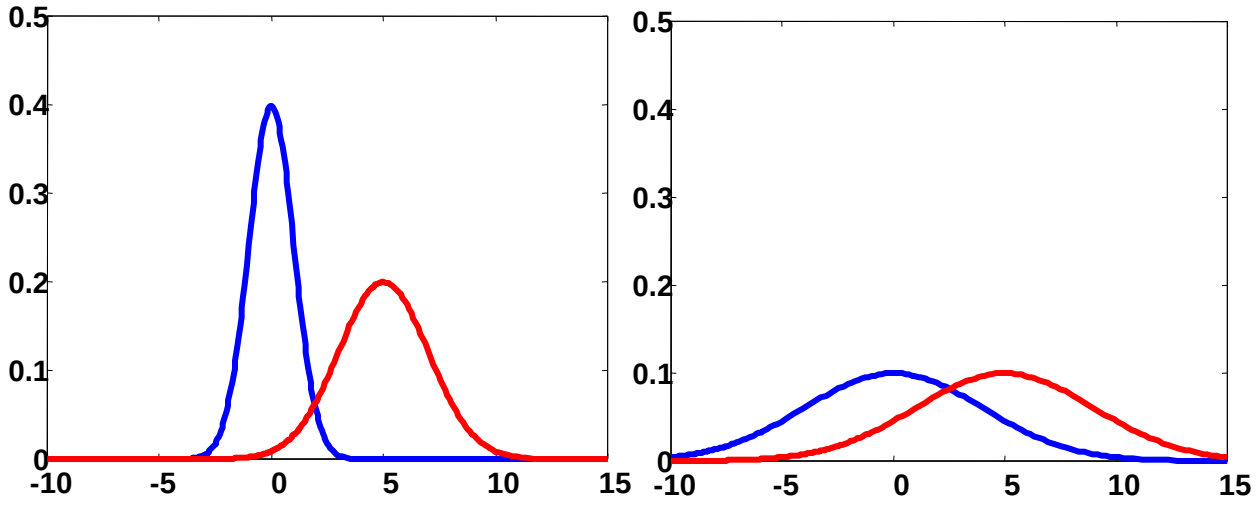
$\boldsymbol{d}^*$

	x_1	x_2	x_3	x_4	x_5
x_1	0	d_3	d_3	d_4	d_4
x_2		0	d_1	d_4	d_4
x_3			0	d_4	d_4
x_4				0	d_2
x_5					0

BioSB

Validity measures

- Many are based on within and between group scatter
- The larger the between group scatter and the smaller the within group scatter, the better
- Example: Davies-Bouldin



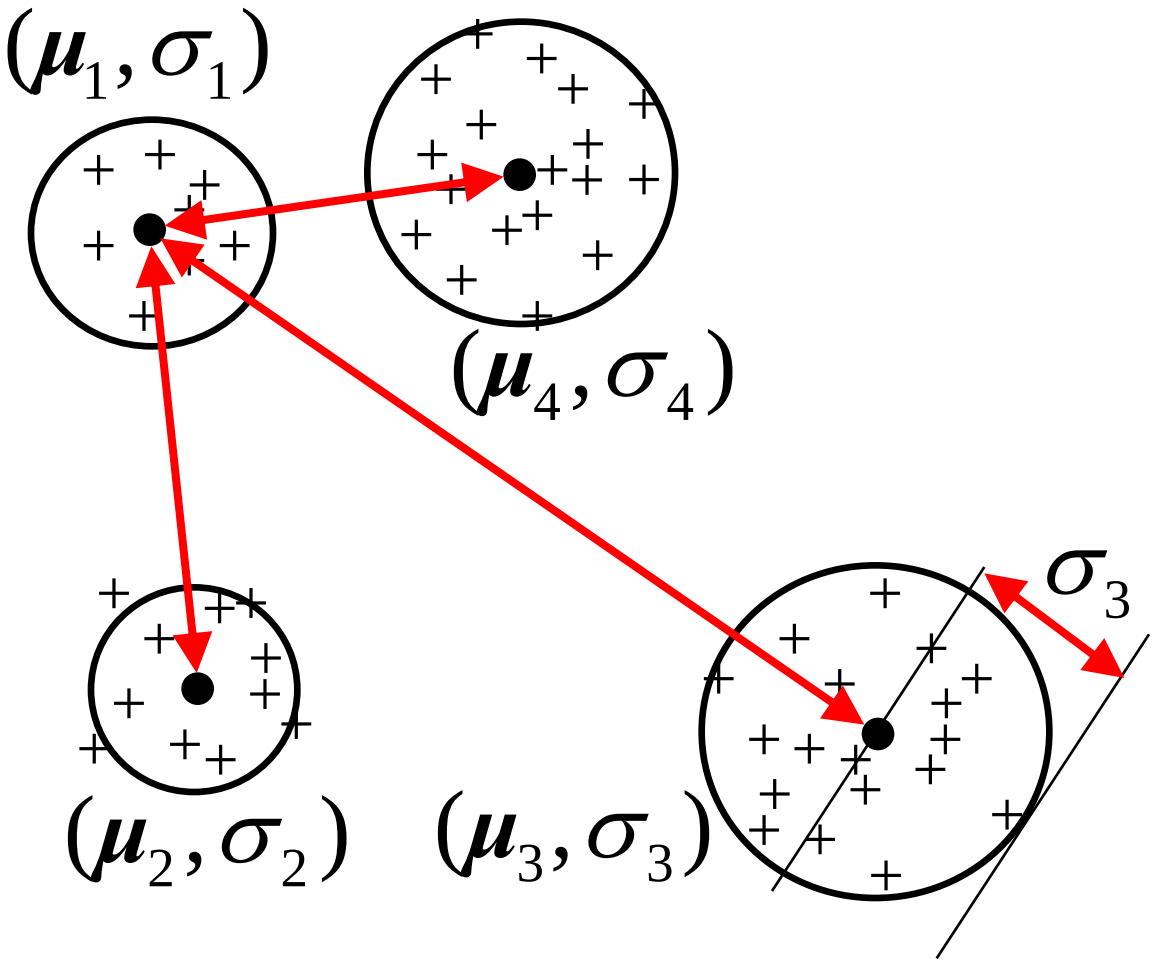
Davies-Bouldin index

- Assumption: clusters are spherical
- For a good clustering, it should hold that:
  - objects are compactly organized within a cluster
  - clusters are far apart

- D.L. Davies and D.W. Bouldin, IEEE Transactions on Pattern Analysis and Machine Intelligence 1, pp. 224-227, 1979

BioSB

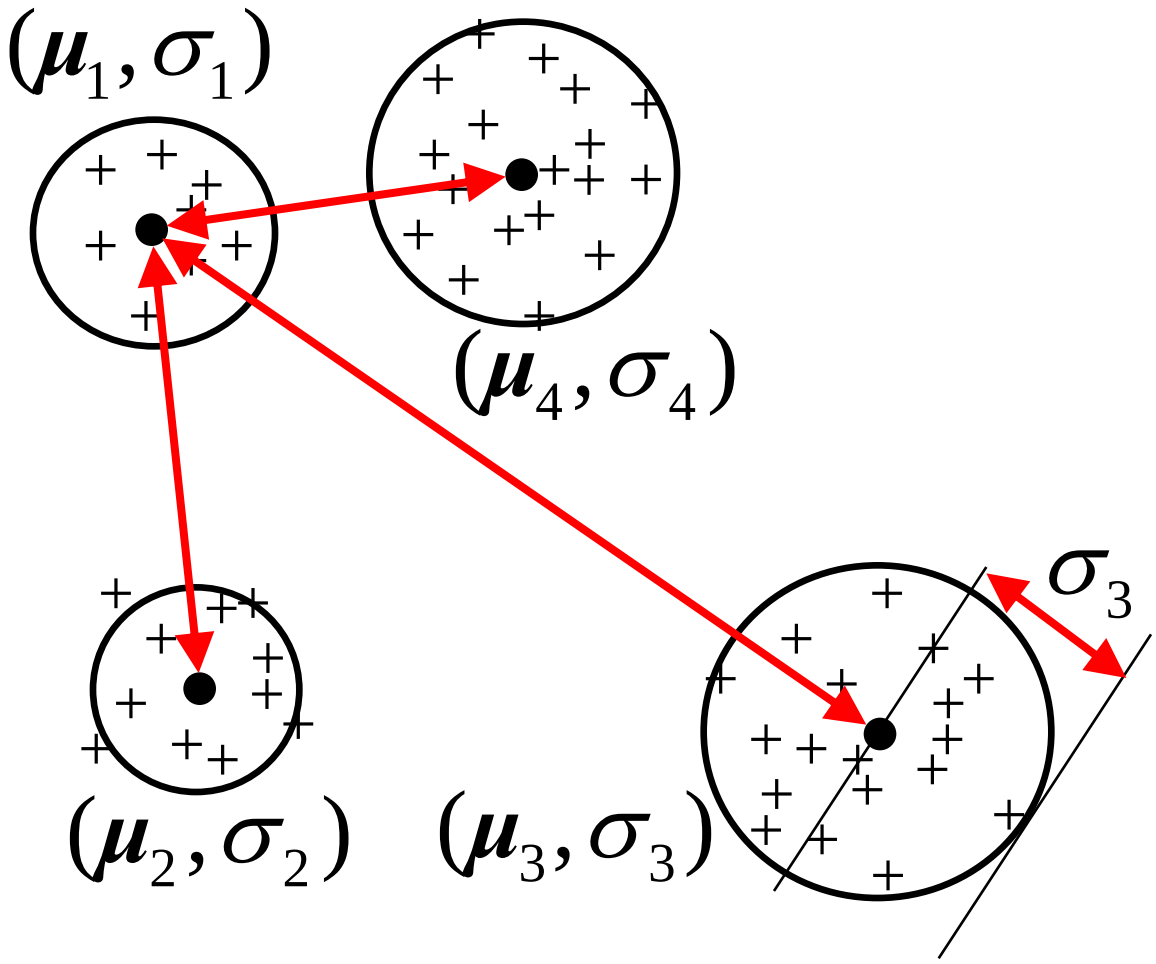
Davies-Bouldin index (2)



$$\sigma_j = \sqrt{\frac{1}{n_j} \sum_{x_i \in C_j} \left\| x_i - \mu_j \right\|^2}$$

$$\mu_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

Davies-Bouldin index (3)



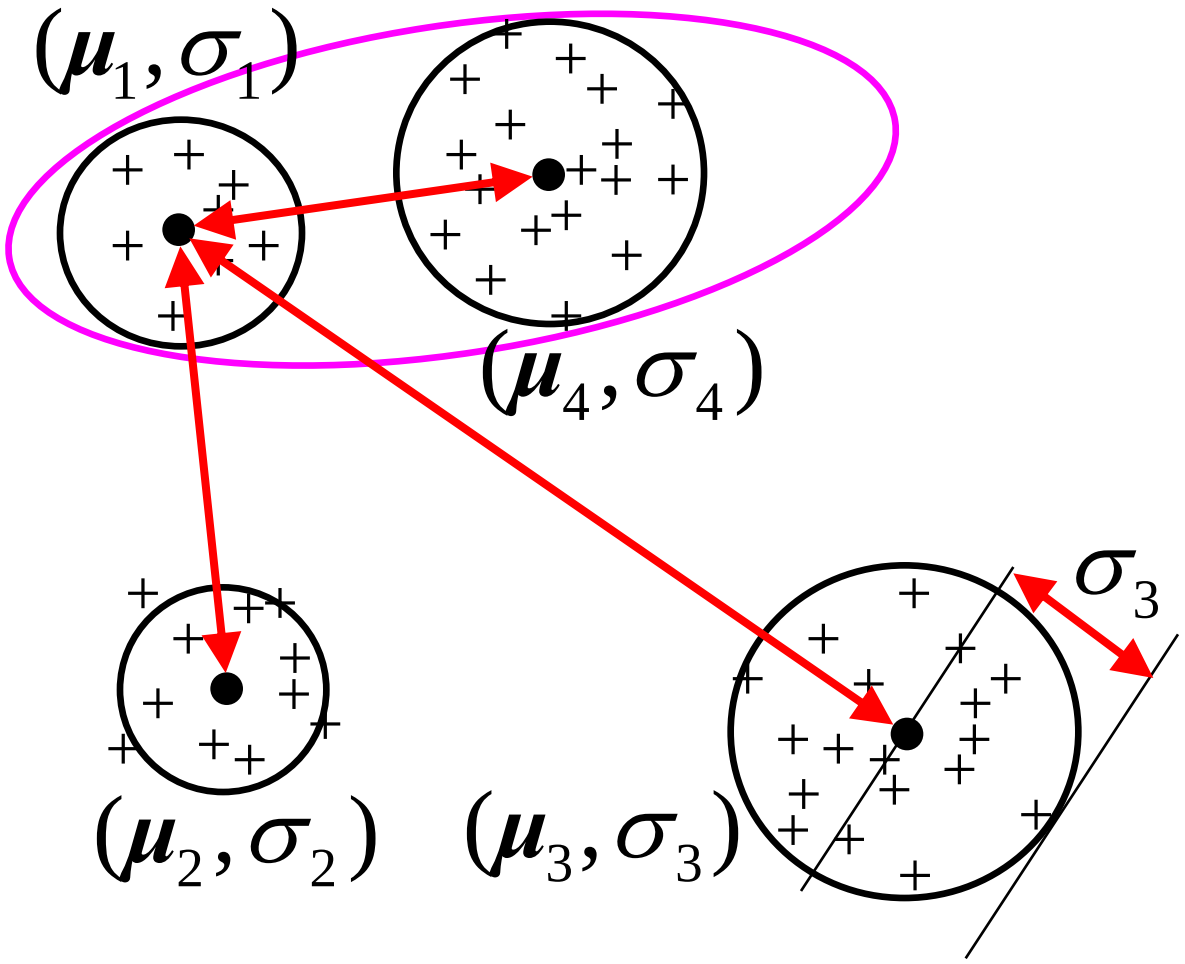
$$R_{jk} = \frac{\sigma_j + \sigma_k}{\left\| \boldsymbol{\mu}_j - \boldsymbol{\mu}_k \right\|}$$

$$\sigma_j = \sqrt{\frac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2}$$

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i$$

BioSB

Davies-Bouldin index (4)



$$R_{jk} = \frac{\sigma_j + \sigma_k}{\left\| \boldsymbol{\mu}_j - \boldsymbol{\mu}_k \right\|}$$

$$R_j = \max_{k=1,..g;k\neq j} R_{jk}$$

Davies-Bouldin index (5)

$$R_{jk} = \frac{\sigma_j + \sigma_k}{\left\| \boldsymbol{\mu}_j - \boldsymbol{\mu}_k \right\|}$$

Paired cluster criterion

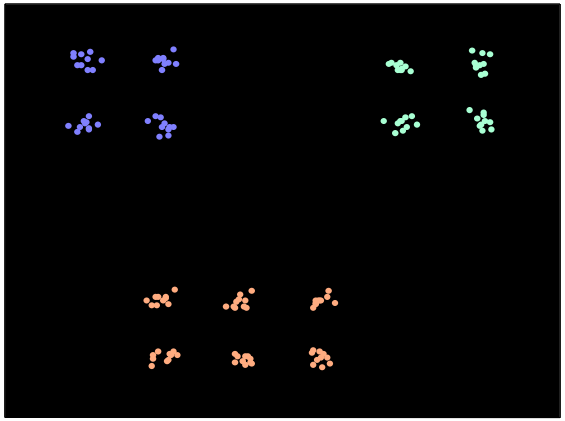
$$R_j = \max_{k=1,..g;k\neq j} R_{jk}$$

Worst-case value per cluster

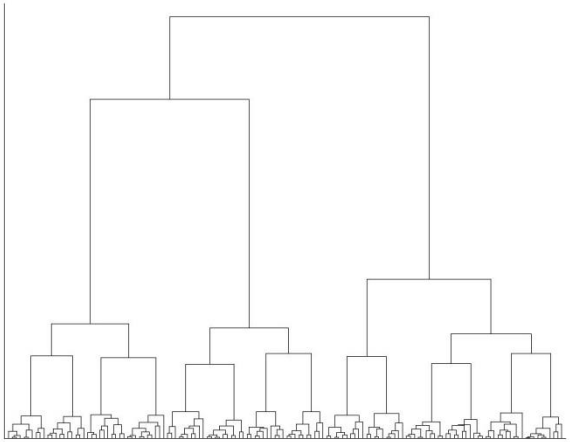
$$I_{DB} = \frac{1}{g} \sum_{j=1}^{g} R_j$$

Average worst-case

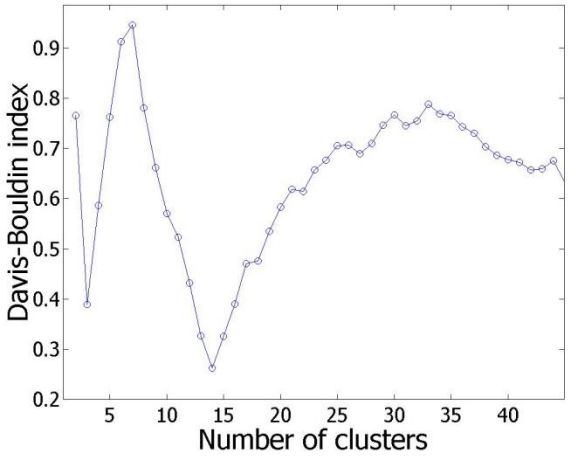
Davies-Bouldin index (5)



Dataset



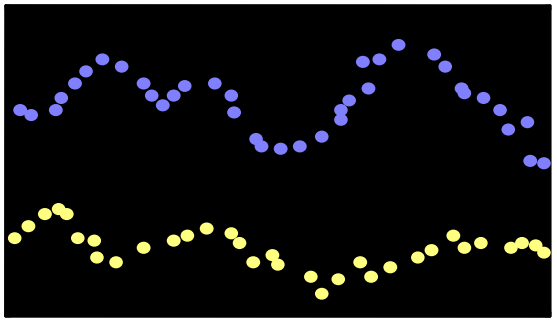
Complete link



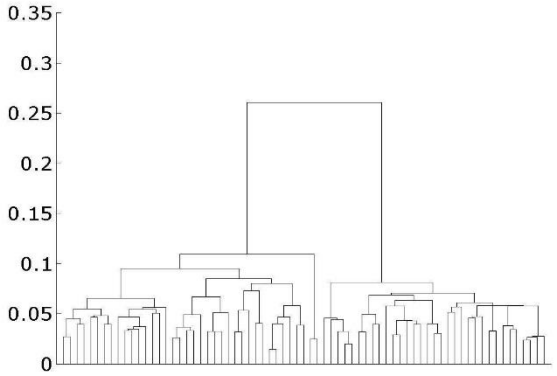
Davies-Bouldin:
3 or 14 clusters

BioSB

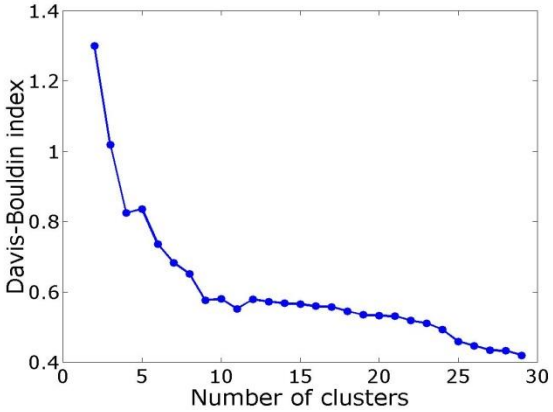
Davies-Bouldin index (7)



Single link

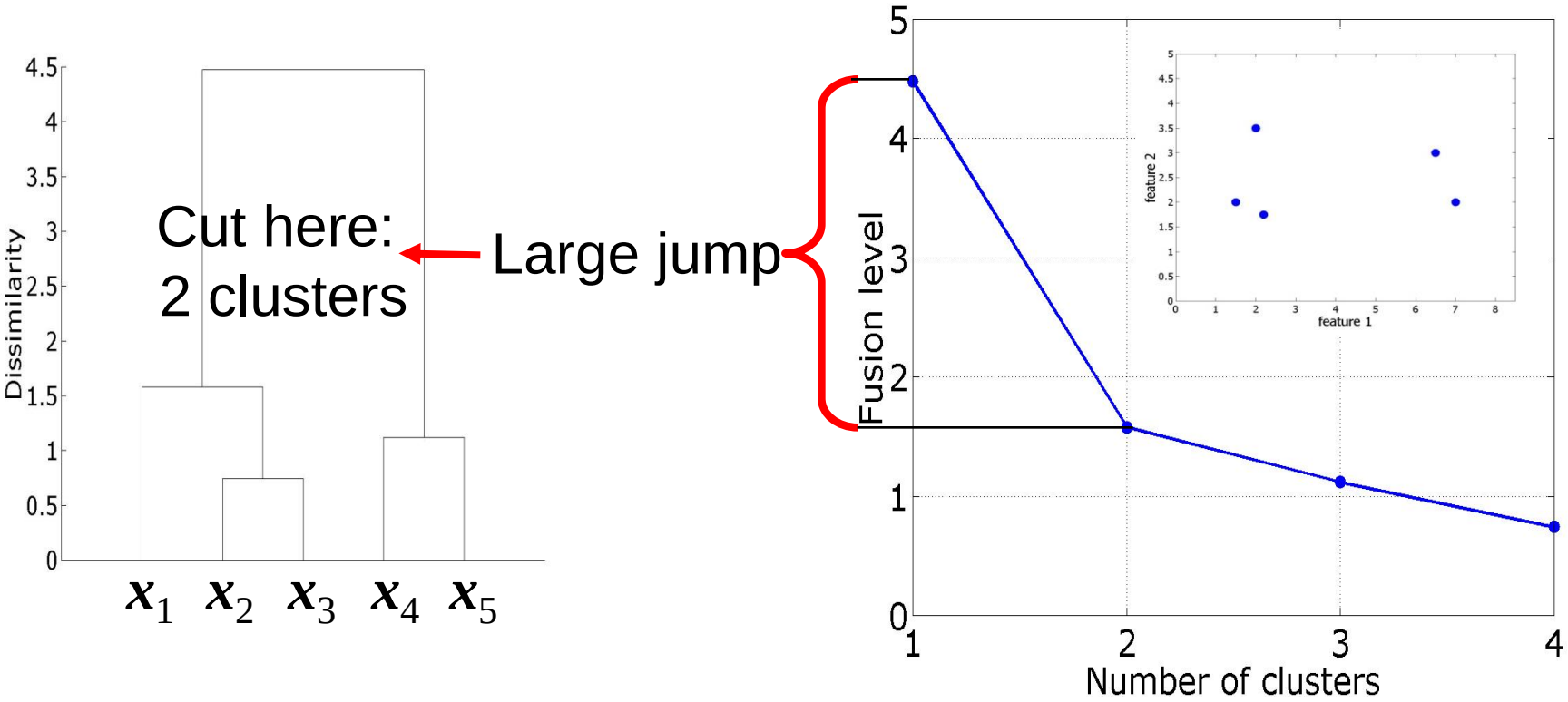


Davies-Bouldin:



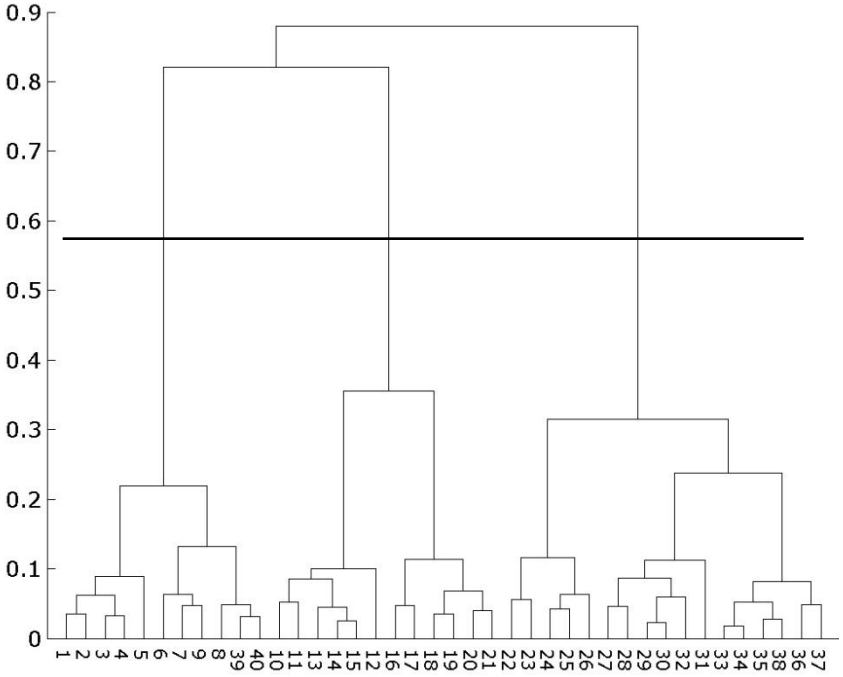
Fusion graph

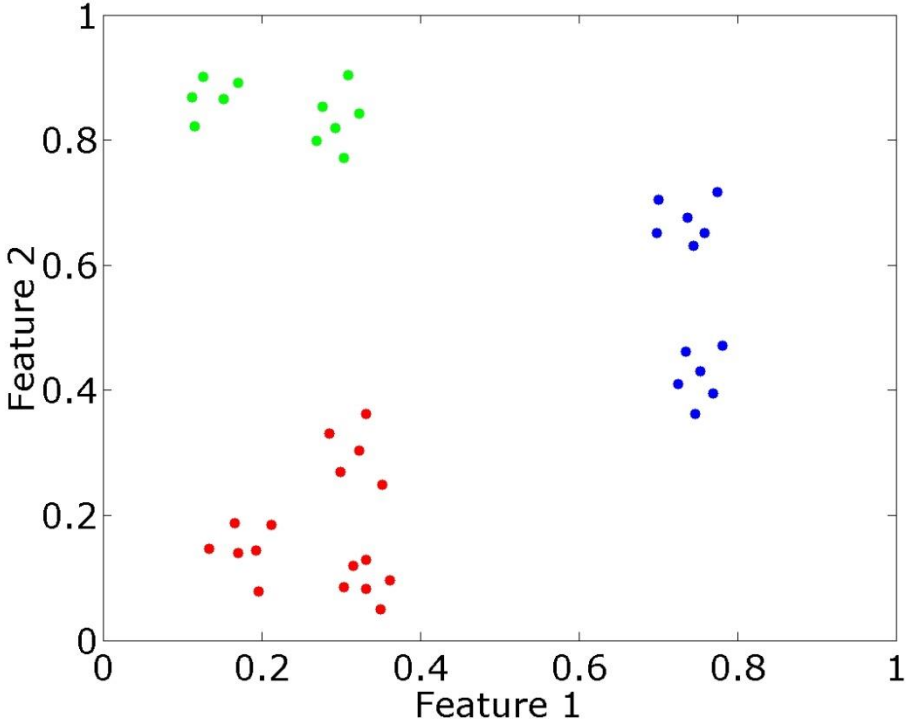
- Heuristic approach: fusion level



Fusion graph (2)

(Euclidean; complete linkage)

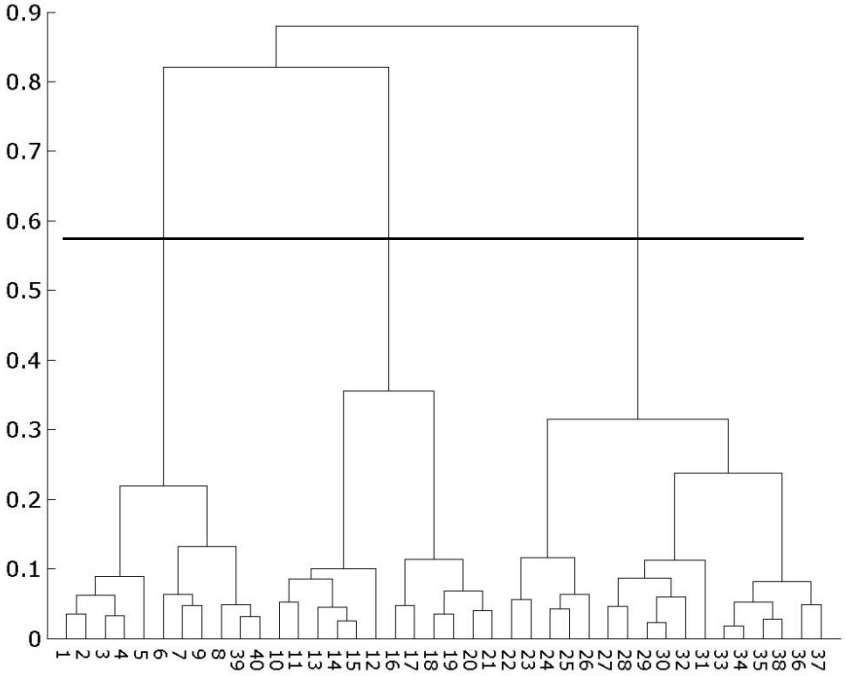


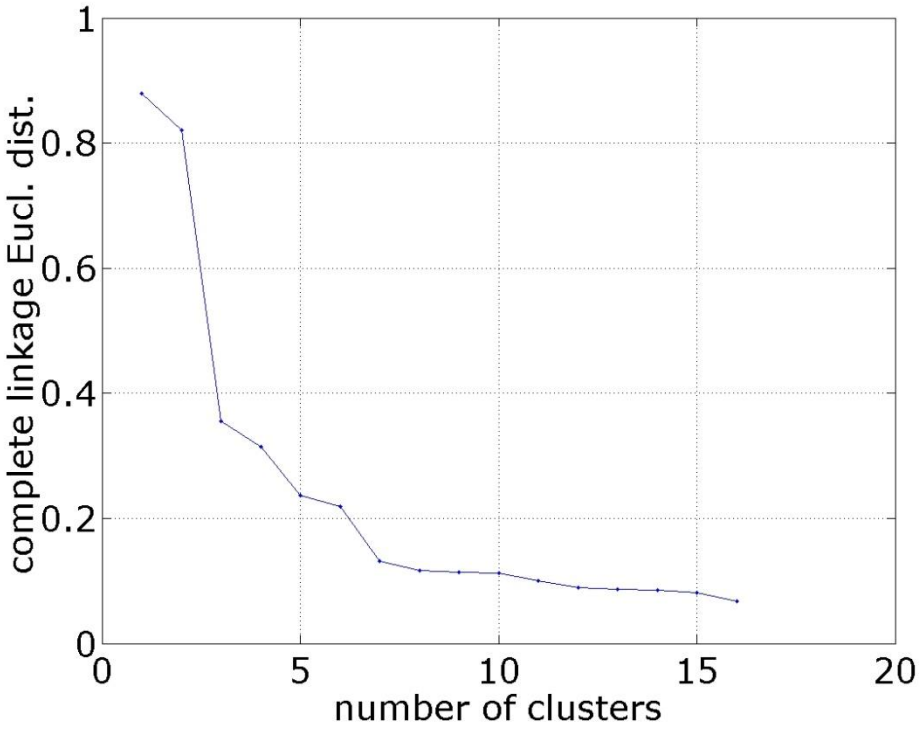


BioSB

Fusion graph (3)

(Euclidean; complete linkage)

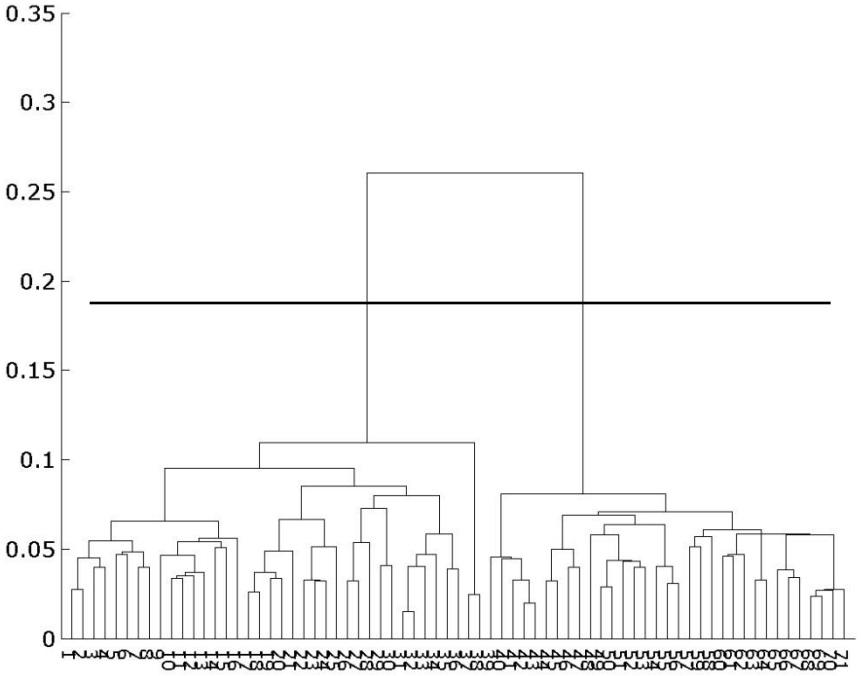


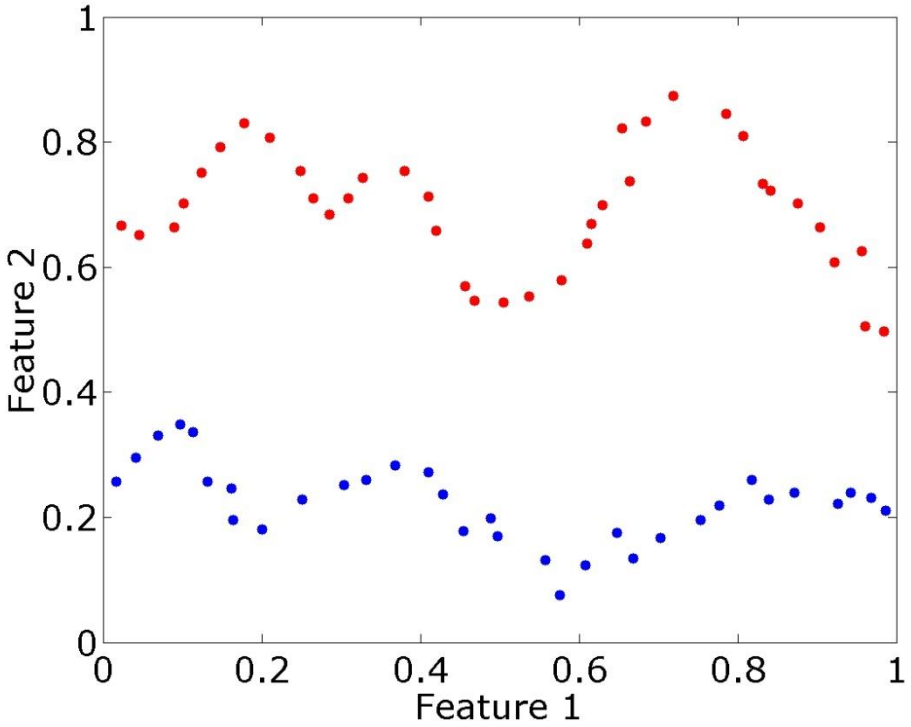


BioSB

Fusion graph (4)

(Euclidean; single linkage)

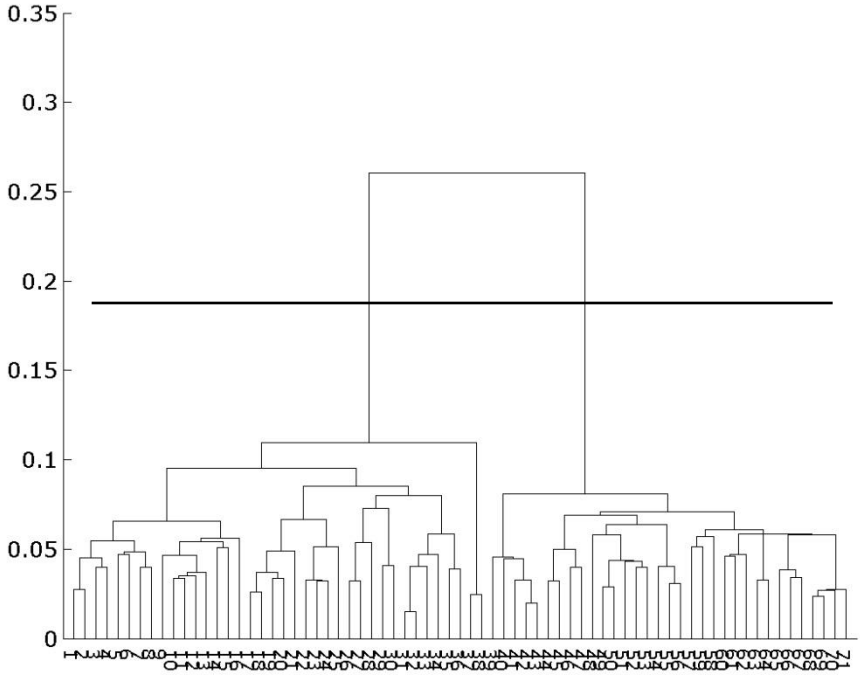


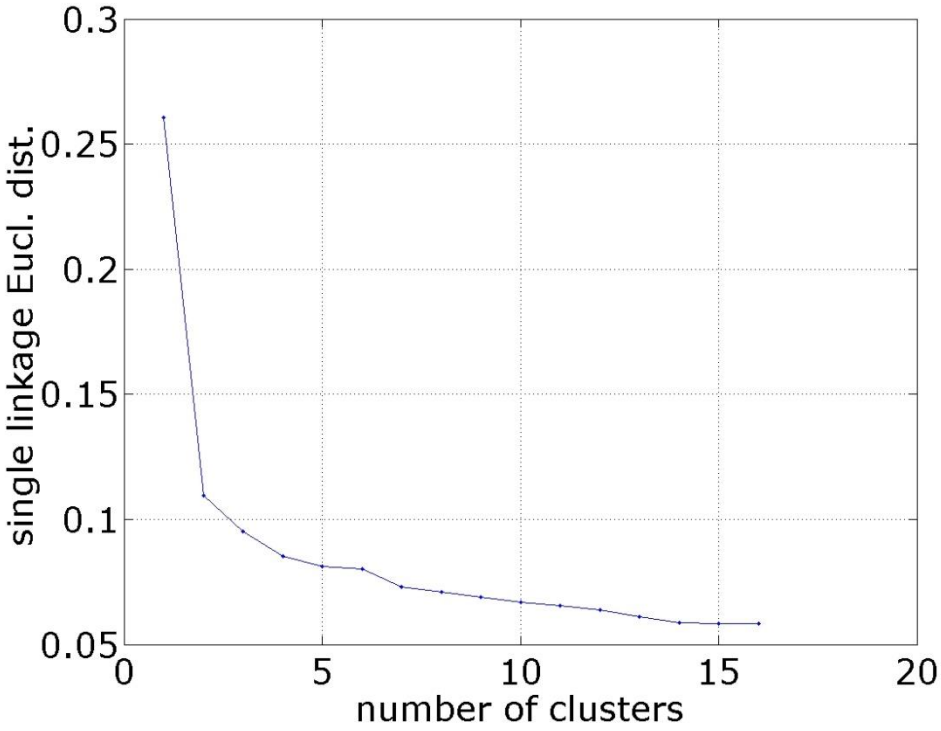


BioSB

Fusion graph (5)

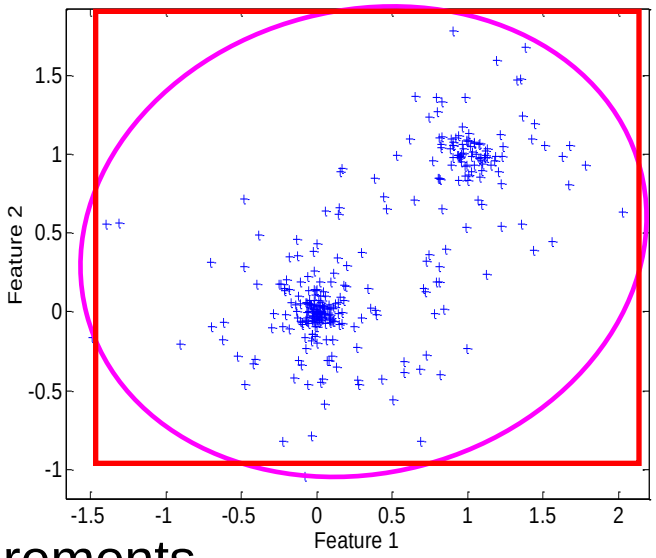
(Euclidean; single linkage)





What is a large jump?

- Compare the fusion graph of the dataset with a *null hypothesis*, i.e. a dataset where the clustering structure has been destroyed
- Different approaches:
  - Generate random data within bounding box or convex hull of data;
  - Preferable to shuffle data, i.e. not generate new data, but perturb relationships between measurements
  - For example, randomly match feature values, i.e. permute values within columns



BioSB

The gap statistic

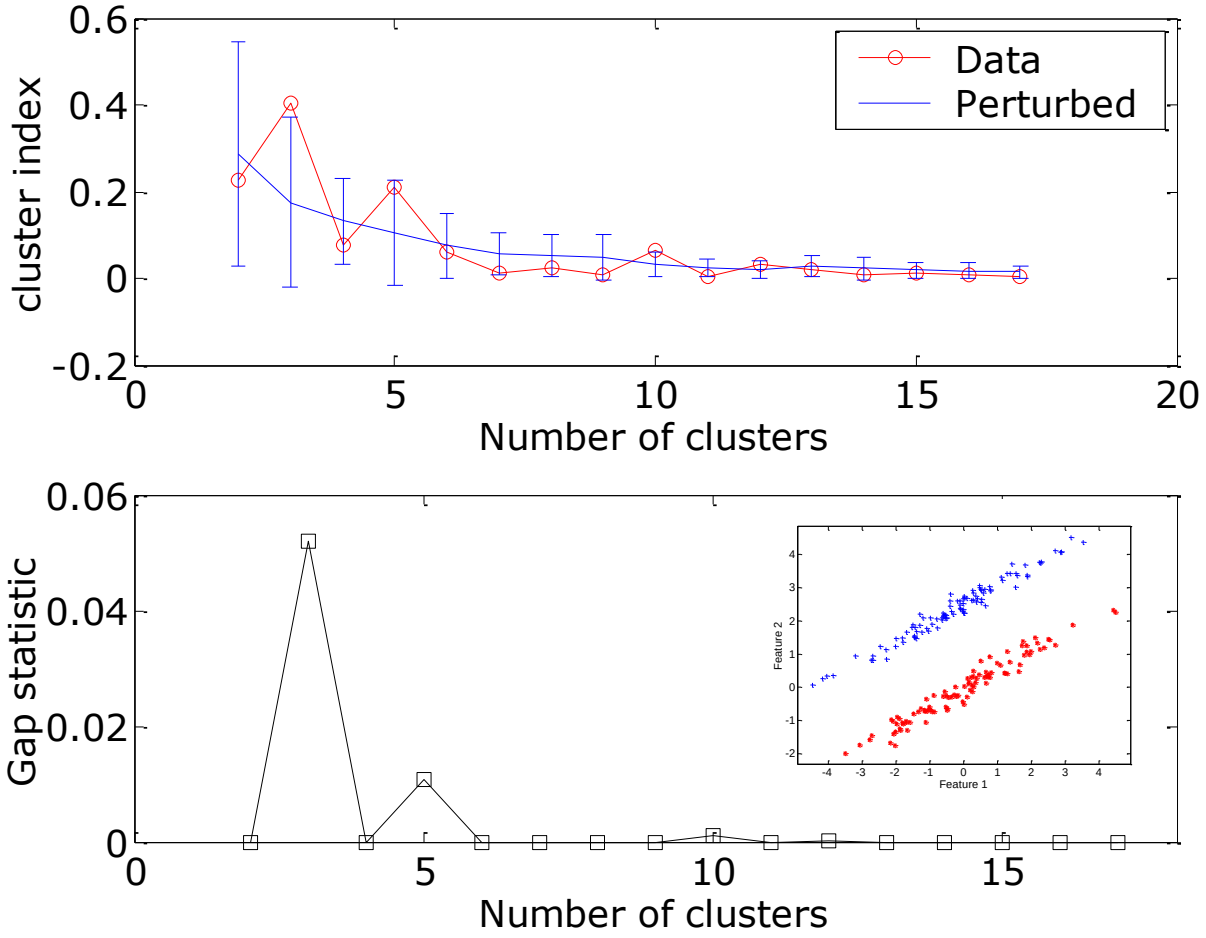
1. Generate dendrogram and extract fusion graph, f_j
2. Repeat r times
   1. Perturb columns
   2. Generate dendrogram and fusion graph, $f^*_{j,r}$
3. Compute average μ^*_j and standard deviation σ^*_j of these perturbed graphs
4. Compute the difference between the data fusion graph and the average perturbed fusion graph (*gap statistic*):

$$g_j^{gap} = \max\left\{ f_j - \mu_j^*, 0 \right\}, j = 1, 2, ..., g$$

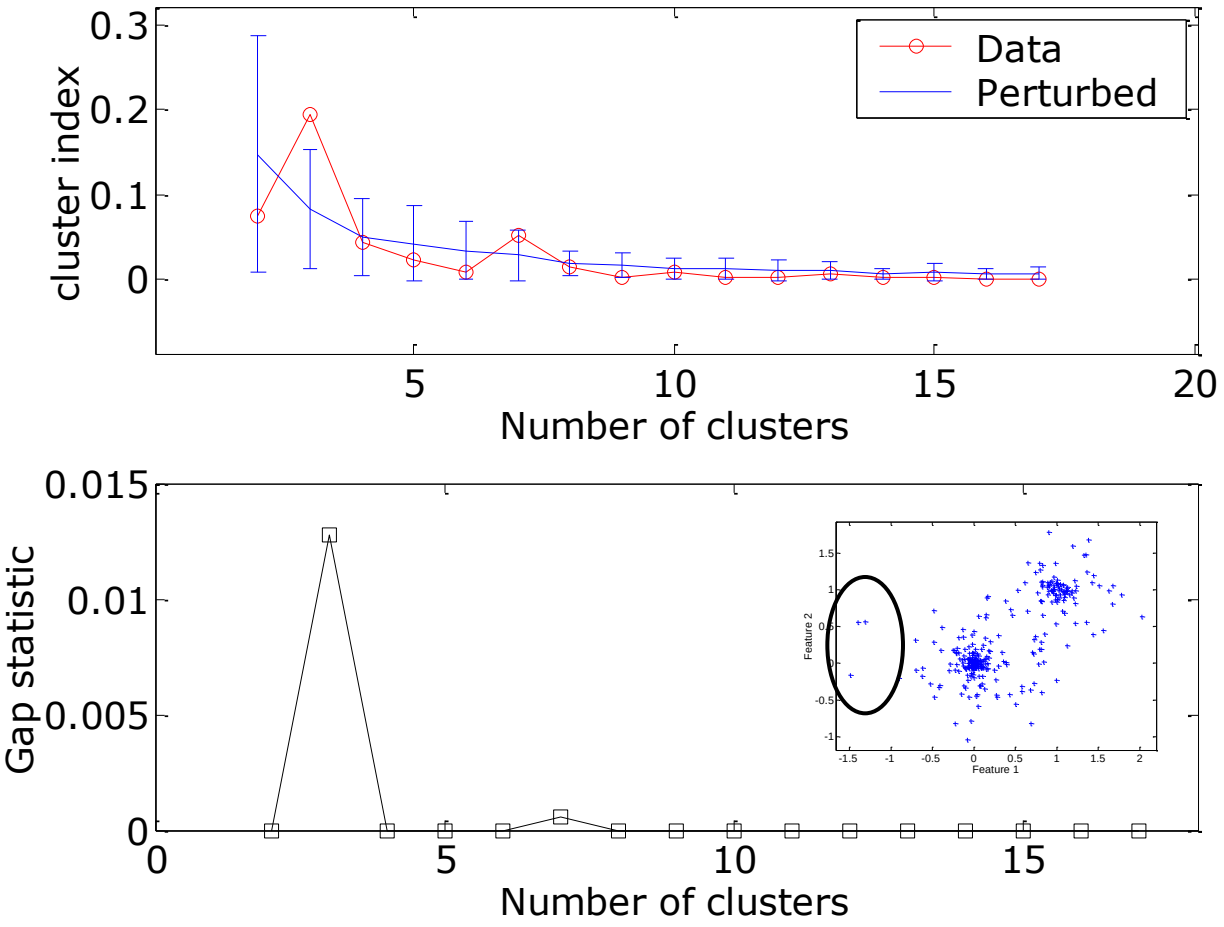
5. Look for large values of gap statistic $g_j^{gap} = f_j$

BioSB

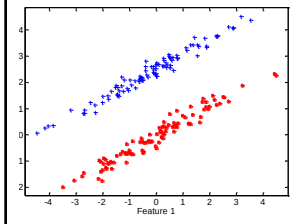
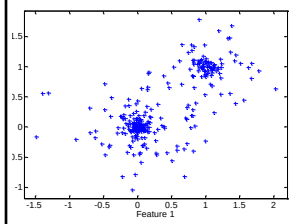
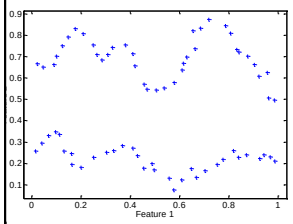
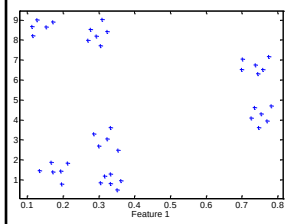
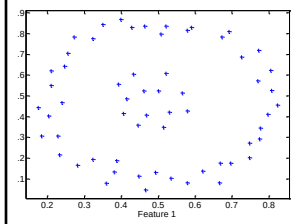
Gap fusion graph (single linkage)



Gap fusion graph (single linkage) (2)



DBI vs. fusion graphs

					
DBI (s)	?	3/4	?	4	4+
DBI (c)	8+	2	5+	4	8+
Gap fusion graph (s)	3	3	2	3	2
Gap fusion graph (c)	2 (?)	2	4	3	3

BioSB

Recapitulation

- *Cluster validation* is used for:
  - Assessing clustering
  - Deciding on the number of clusters

- Methods:
  - *Distortion* measures (dendrogram)
  - *Davies-Bouldin index*
  - *Fusion graph* and *gap statistic*

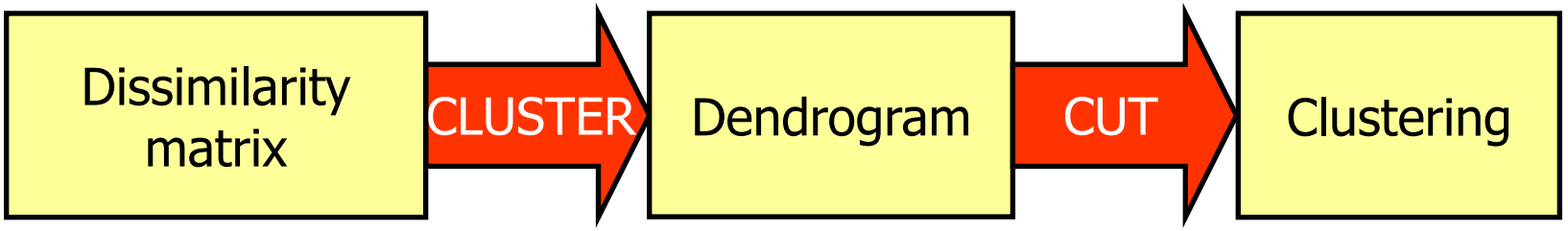
- When applying cluster validation, one also needs to define what a good cluster is – like in clustering itself.
  There's no free lunch...

BioSB

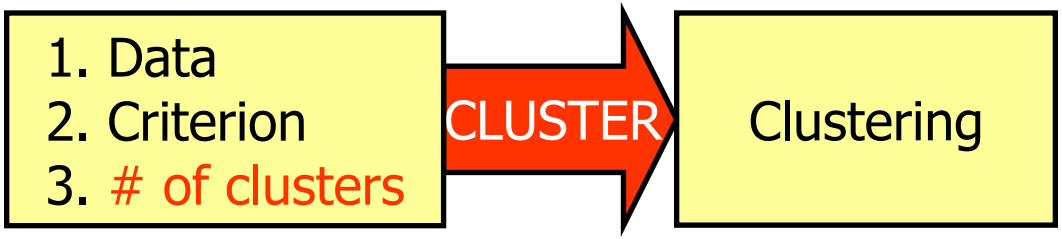
Exercise 2.20-2.28

Clustering overview

1. Hierarchical:



2. K-means:

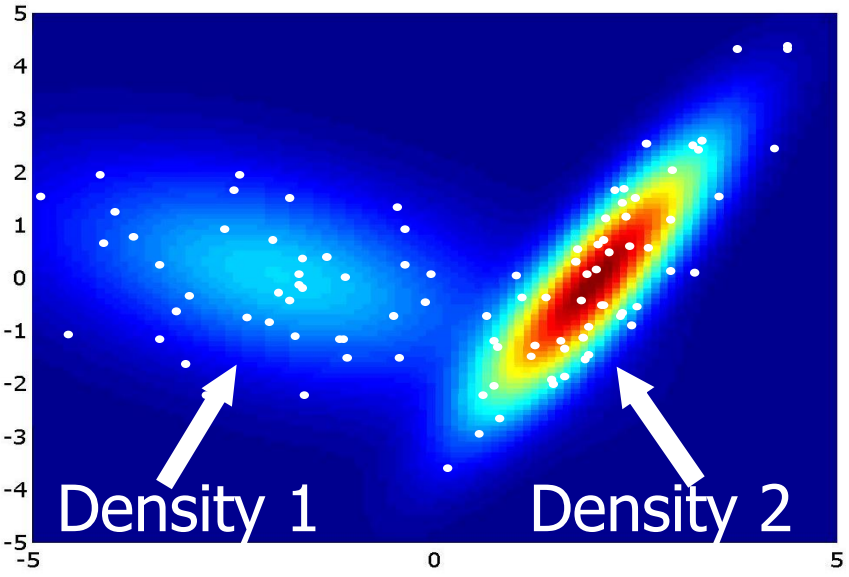


3. Density-based:

BioSB

Density-based clustering

- Each cluster is described by a probability density function
- Total dataset described by a *mixture* of density functions
- Clustering = maximizing the mixture fit
- Clusters are based on *a posteriori probabilities*

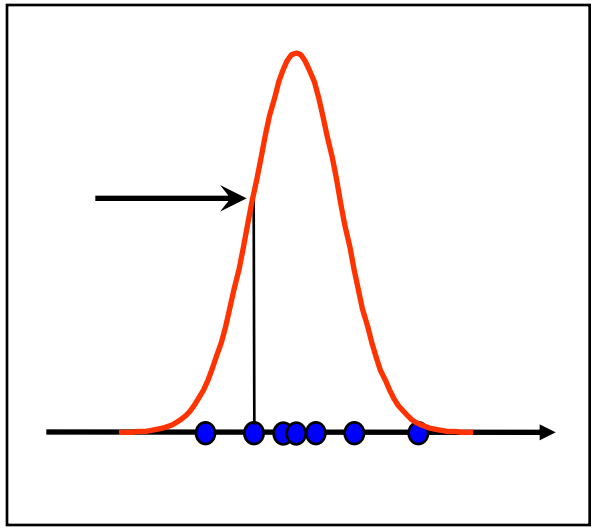


BioSB

Density-based clustering (2)

- Given:
  - n independent objects: $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$
  - probability density function model:

$$p(\boldsymbol{x} \mid \theta) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$



- Estimate parameters $\theta = \{\mu, \Sigma\}$ such that model *fits* data
- Use *likelihood* as criterion: probability of observing the data set, given the model (as on Day 1, for kernel width h in Parzen density estimation)

BioSB

Estimation: maximum likelihood

- General method to estimate parameters θ of probability distribution from data $D = \{x_1, ..., x_n\}$. How?
- Maximize joint probability of the data

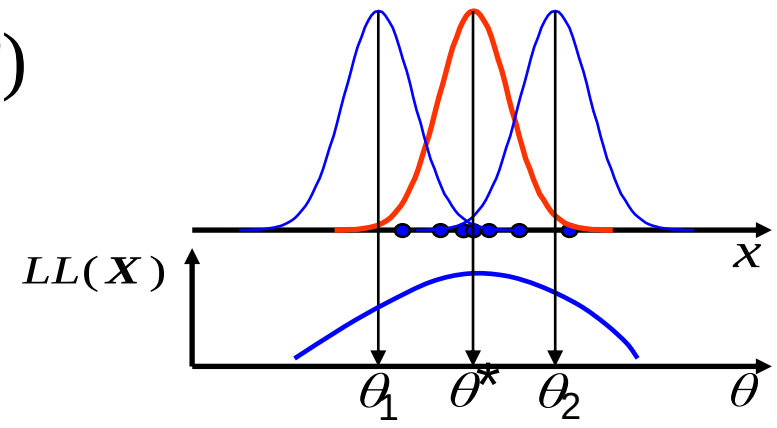
independence

likelihood:

$$L = p(x_1, ..., x_n \mid \theta) = \prod_{i=1}^{n} p(x_i \mid \theta)$$

log-likelihood:

$$LL = \sum_{i=1}^{n} \log \sum_{Q} p(x_i, Q \mid \theta)$$



same solution since log is monotonic

Estimation: maximum likelihood (2)

Two possible outcomes: $x = 0$ or $x = 1$.
Success ($x = 1$) occurs with probability p

Bernoulli distribution: $P(x) = p^x (1-p)^{1-x}$

Likelihood: $P(X_1 = x_1, \ldots, X_n = x_n \mid p) = p^{x_1}(1-p)^{1-x_1} \ldots p^{x_n}(1-p)^{1-x_n}$

$$= p^{n_1}(1-p)^{n-n_1}$$

n_1: # of successes

$$\frac{d(p^{n_1}(1-p)^{n-n_1})}{dp} = 0$$

Maximum at $p = n_1/n$

BioSB

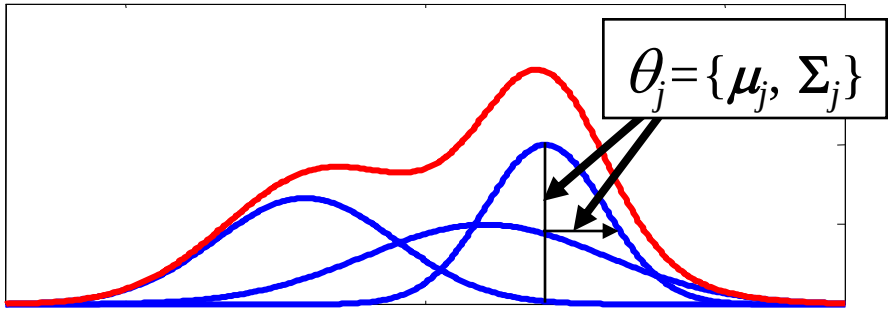
Mixture-of-Gaussians

- Choose Gaussian as component density $p(\boldsymbol{x};\theta_j)$:

$$p(\boldsymbol{x};\theta_j) = \frac{1}{\sqrt{2\pi^p \det(\boldsymbol{\Sigma}_j)}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right)$$

- Describe complete data set as a mixture of $p(\boldsymbol{x};\theta)$'s:

$$p(\boldsymbol{x};\Psi) = \sum_{j=1}^{g} \pi_j p(\boldsymbol{x};\theta_j) \quad \text{with} \quad \sum_{j=1}^{g} \pi_j = 1$$



BioSB

Mixture-of-Gaussians (2)

$$p(\mathbf{x};\Psi) = \sum_{j=1}^{g} \pi_j p(\mathbf{x};\theta_j) \quad \text{with} \quad \sum_{j=1}^{g} \pi_j = 1$$

- Parameters:
  - Set number of clusters, g
  - Estimate other parameters by maximum-likelihood:

$$\Psi = (\boldsymbol{\pi}, \theta = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1\ldots g})$$

mixture coefficients

component density parameters

log-likelihood: $$LL(\boldsymbol{X};\Psi) = \sum_{i=1}^{n} \log \sum_{j=1}^{g} \pi_j p(\mathbf{x}_i;\theta_j)$$

BioSB

EM algorithm

- **Problem:** need to simultaneously estimate two interdependent things...
  - Cluster membership of each object
  - Density parameters of each cluster: $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$

- **Expectation-Maximization algorithm:**
  - General class of algorithms for this type of problem
  - Repeatedly:
    - Recalculate cluster membership of each object *(E)*
    - Recalculate density parameters of each cluster *(M)*

- Introduce a hidden variable *z* to explicitly indicate mixture components

$$\pi_j = p(z = j)$$

BioSB

Intermezzo: probabilities

$n = 20$

	1	2	3	4	5	6
die 1	•	• •	•••	•••	•	•
die 2			•	•	•••	••••

sum rule: $P(x) = \sum_y P(x, y)$

$$1/5 = P(3) = P(3,\ \text{die 1}) + P(3,\ \text{die 2}) = 3/20 + 1/20$$

product rule: $P(x, y) = P(x \mid y)P(y) = P(y \mid x)P(x)$

$$3/20 = P(3,\ \text{die 1}) = P(3 \mid \text{die 1})P(\text{die 1}) = (3/11)(11/20) = 3/20$$

$$= P(\ \text{die 1} \mid 3)P(3) = (3/4)(4/20) = 3/20$$

BioSB

Intermezzo: Bayes' theorem

From product rule

$$P(x \mid y)P(y) = P(y \mid x)P(x)$$

$$\textbf{Bayes}: \; P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)} = \frac{P(y \mid x)P(x)}{\sum_x P(y \mid x)P(x)}$$

$$P(\text{die 1} \mid \ 3) = \frac{P(3 \mid \text{die 1})P(\text{die 1})}{P(3)} = \frac{(3/11)(11/20)}{4/20} = 3/4$$

BioSB

EM algorithm (2)

arbitrary distribution hidden variable

$$\log p(D) = \sum_x \log p(x) = \sum_x \sum_z q(z) \log p(x)$$

$$= \sum_{x,z} q(z) \log \frac{p(x,z)}{p(z \mid x)} = \sum_{x,z} q(z) \log \left(\frac{p(x,z)}{p(z \mid x)} \times \frac{q(z)}{q(z)} \right)$$

$$= \sum_{x,z} q(z) \log \left(\frac{p(x,z)}{q(z)} \right) + \sum_{x,z} q(z) \log \left(\frac{q(z)}{p(z \mid x)} \right)$$

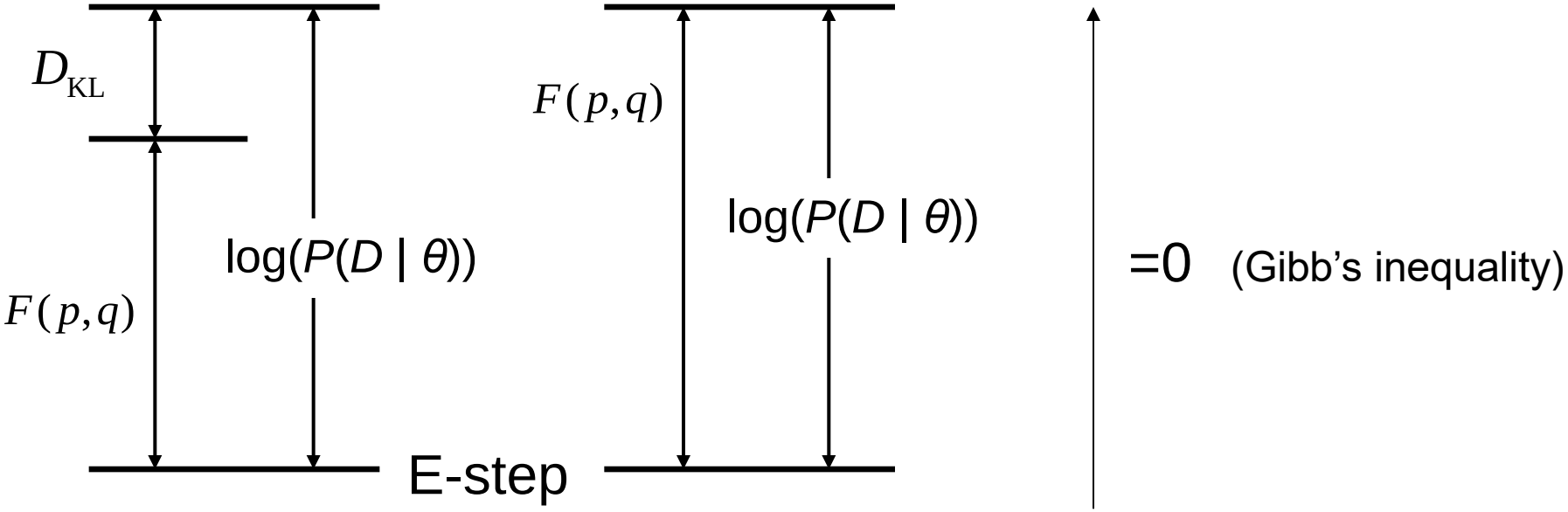
$$= F(p_{\text{joint}}, q) + D_{KL}(q \,\|\, p_{\text{post}})$$

free energy relative entropy (≥ 0)

BioSB

EM algorithm: E-step

$$\log p(D) = \sum_{x,z} q(z) \log \left(\frac{p(x,z)}{q(z)} \right) + \boxed{\sum_{x,z} q(z) \log \left(\frac{q(z)}{p(z \mid x)} \right)}$$



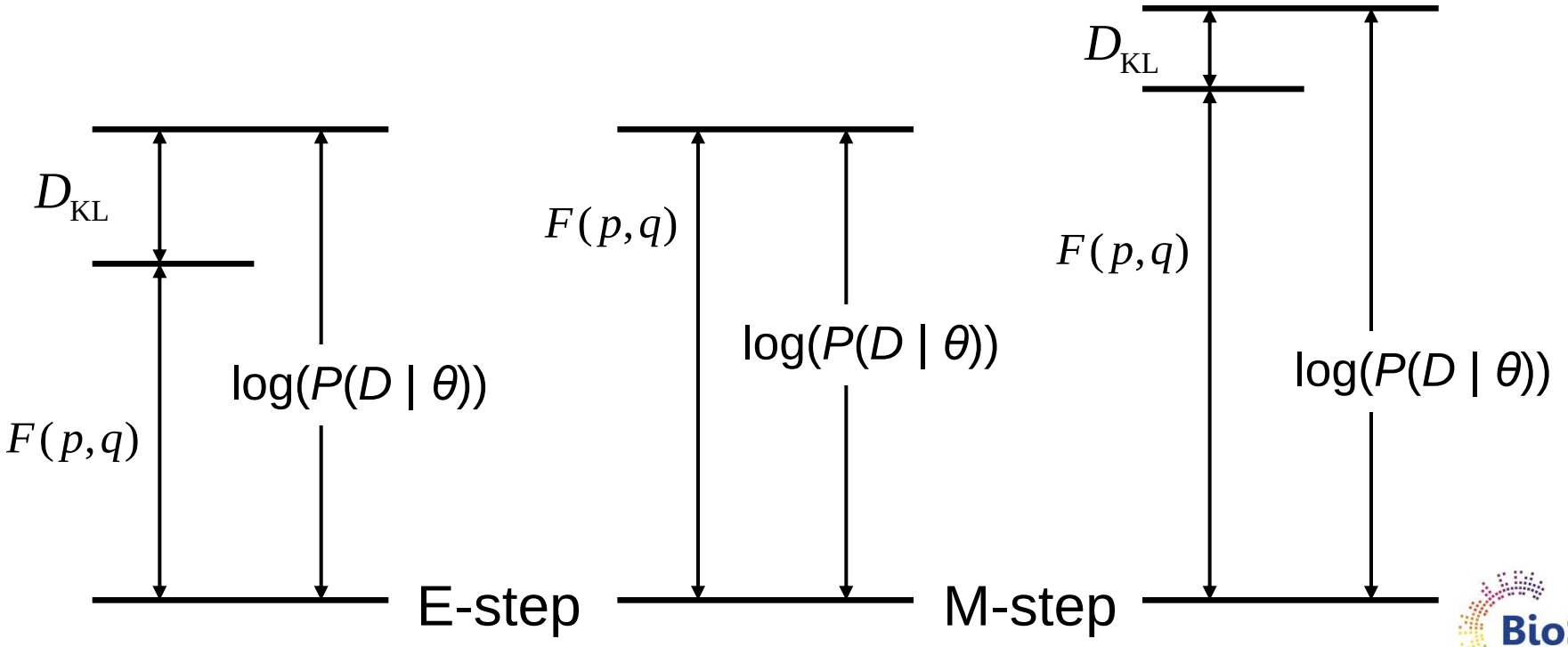
E-step: $q^{\mathrm{new}}(z \mid x) = p_{\mathrm{post}} = p(z \mid x)$

BioSB

EM algorithm: M-step

$$\log p(D) = \sum_{x,z} p(z \mid x) \log \left(\frac{p(x,z)}{p(z \mid x)} \right)$$

M-step: maximize log[$p(D)$] with respect to the parameters



EM algorithm (3)

Iterate to maximize likelihood:

E-step: $p_{\text{post}} = p(z \mid x, \theta)$

Calculate the distribution of the hidden variables given the data and the model parameters

M-step: $\theta^{new} = \arg\max_{\theta} \sum_{x,z} p(z \mid x) \log p(x, z \mid \theta)$

Maximize the expected (with respect to hidden variables) log-likelihood of the complete data.

Compare M-step with MoG log-likelihood: $\sum_{i=1}^{n} \log \sum_{j=1}^{g} \pi_j \, p(\mathbf{x}_i; \theta_j)$

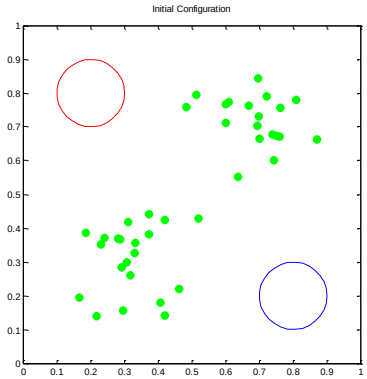
M-step is easier: log within sum

BioSB

EM: mixture model

Very simple example of a model with hidden variables:

2-component mixture model



$$p(x) = \pi_1 p_1(x \mid \theta) + \pi_2 p_2(x \mid \theta)$$

hidden variable z = 1,2 - component label

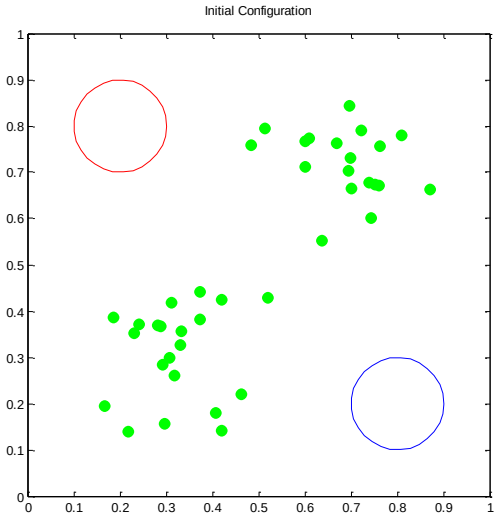
E-step: $p(z = j \mid x, \theta) = \dfrac{p(z = j \mid \theta)\, p(x \mid z = j, \theta)}{p(x \mid \theta)} = \dfrac{\pi_j p_j(x \mid \theta)}{p(x)}$

responsibility

M-step: maximize $\sum_{x, z \in \{1,2\}} p(z \mid x) \log p(x, z \mid \theta)$

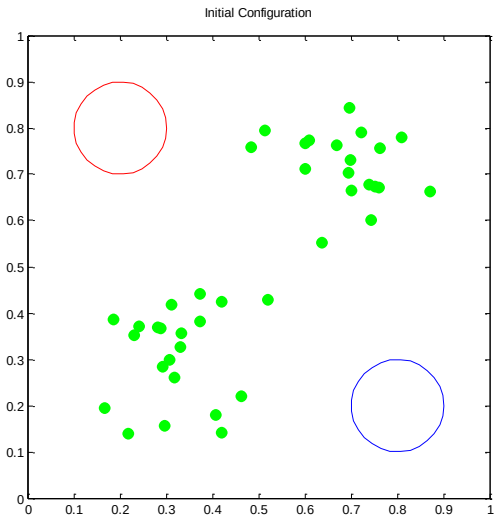
BioSB

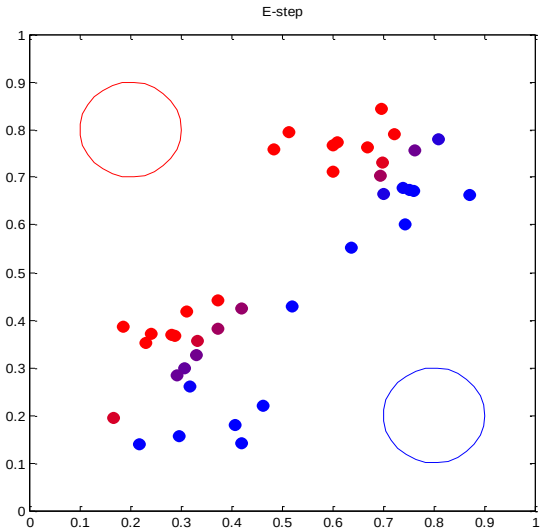
EM: mixture model (2)



Initialization

EM: mixture model (3)





Initialization

E-step

BioSB

EM: mixture model (4)

- **M-step**: Maximization

  Maximize the expected complete LL by updating

  - mixture coefficients π_j
  - cluster means and covariances $\theta_j=\{\mu_j, \Sigma_j\}$, $j=1,\ldots,g$:

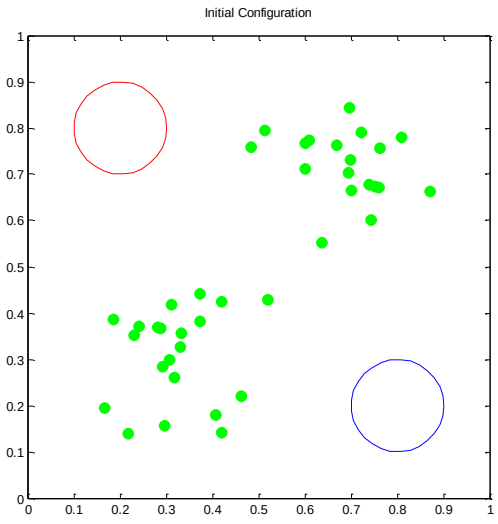
$$\hat{\pi}_j = \frac{1}{n}\sum_{i=1}^{n} p(z=j \mid x_i) = \frac{1}{n}\sum_{i=1}^{n} w_{ij}$$ } "total membership"

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} w_{ij}\boldsymbol{x}_i}{\sum_{i=1}^{n} w_{ij}}$$

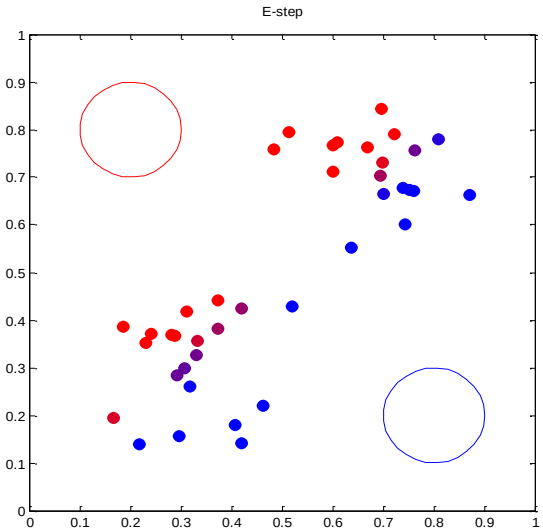
$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{n} w_{ij}(\boldsymbol{x}_i - \hat{\mu}_j)(\boldsymbol{x}_i - \hat{\mu}_j)^T}{\sum_{i=1}^{n} w_{ij}}$$ } weighted sums

BioSB

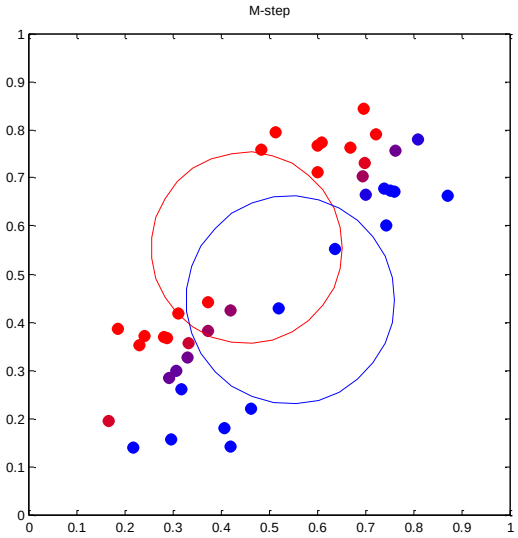
EM: mixture model (5)



Initialization

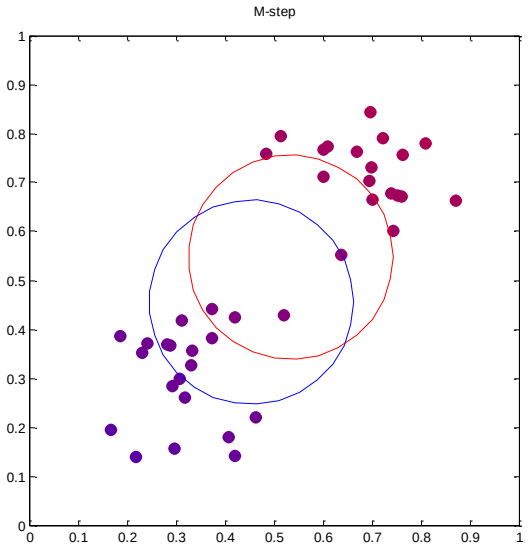


E-step



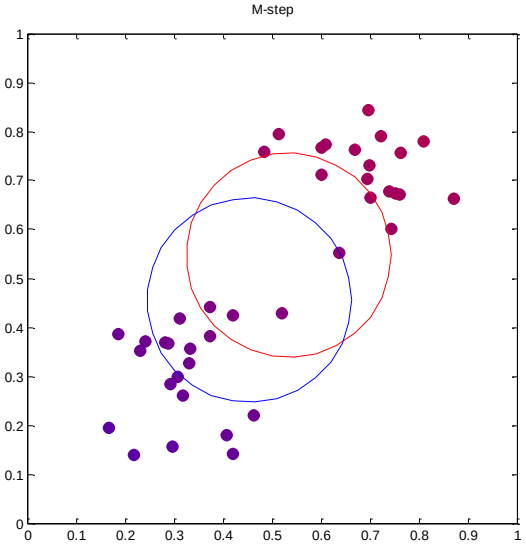
M-step

EM: mixture model (6)

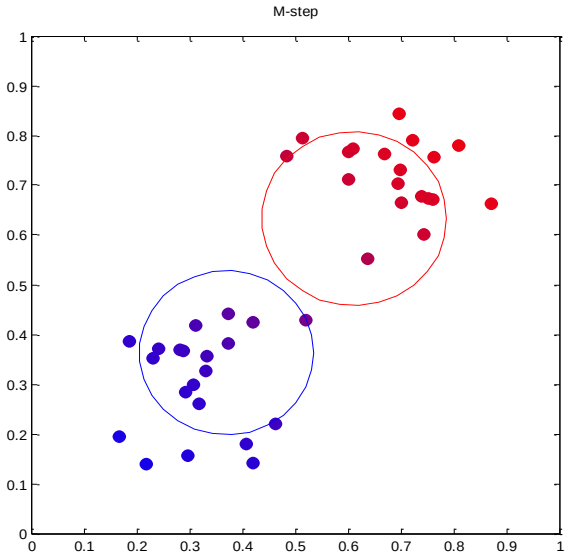


M-step: 3

EM: mixture model (7)



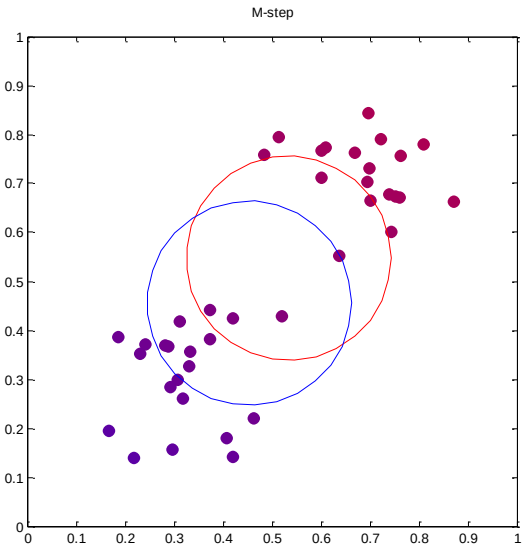
M-step: 3



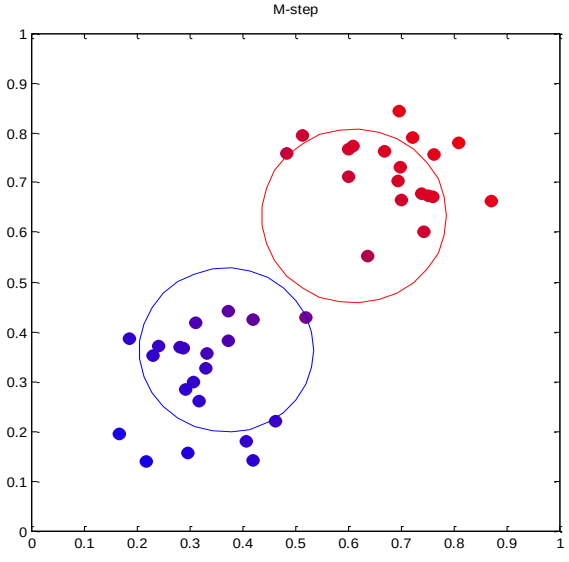
M-step: 5

BioSB

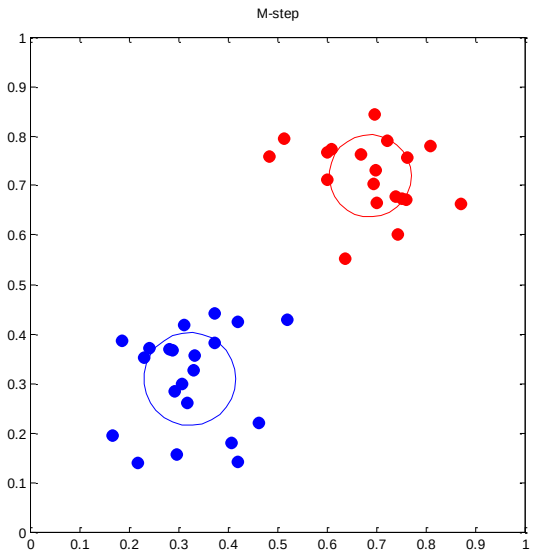
EM: mixture model (8)



M-step: 3



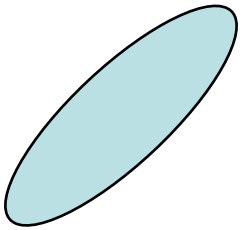
M-step: 5



M-step: 9

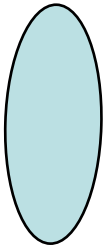
Mixture-of-Gaussians (3)

- ‘Gauss’:



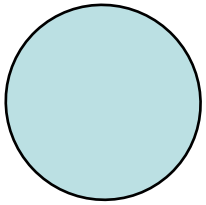
$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

- ‘Aligned’:



$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & 0 \\ 0 & \sigma_{22} \end{bmatrix}$$

- ‘Circular’:



$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix}$$

BioSB

EM: mixture model (9)

- If...
  - all clusters are spherical
  - the variance of each cluster is infinitely small

$$\boldsymbol{\Sigma} = \begin{bmatrix} \varepsilon^2 & 0 & 0 \\ 0 & \varepsilon^2 & 0 \\ 0 & 0 & \varepsilon^2 \end{bmatrix}, \qquad \varepsilon \to 0$$

then the EM algorithm simplifies to the *K*-means algorithm (samples are always assigned to the closest cluster!)

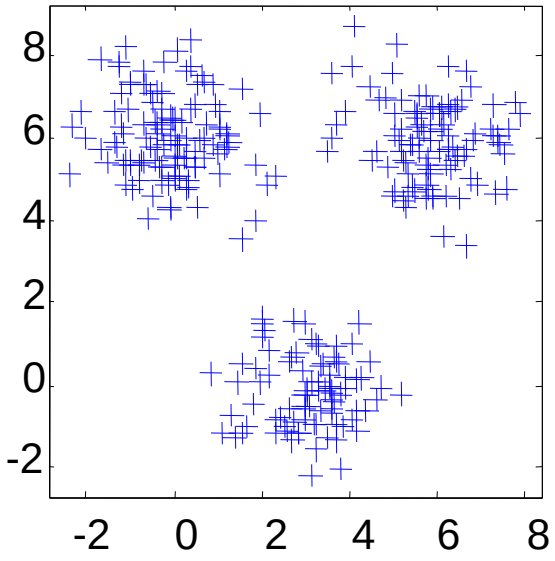
BioSB

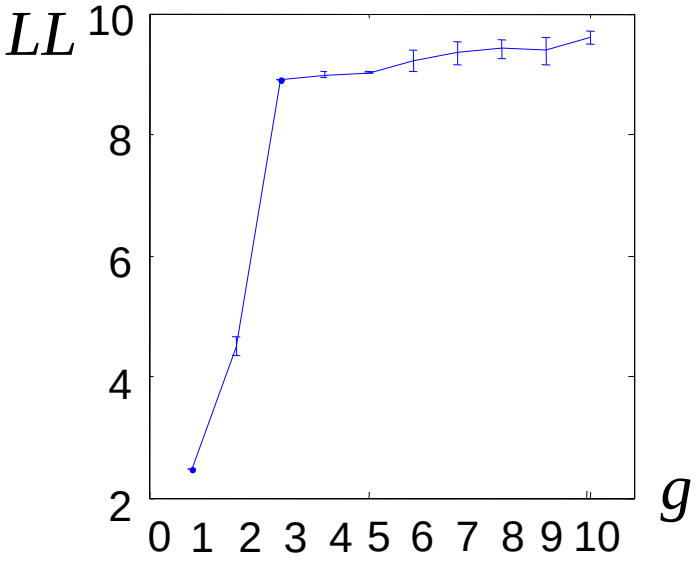
EM algorithm (4)

- Disadvantages:
  - can get stuck in local minima
  - depends on initial conditions
  - convergence can be slow
  - problems with covariance estimates:
    if too few samples are members of a cluster,
    there will not be enough data to base estimate on
- Advantages:
  - simple to implement

Cluster validation: log-likelihood

- For probabilistic models (e.g. mixture-of-Gaussians):
  - Log-likelihood will probably not increase anymore when too many clusters are used
  - Look for “plateau” in log-likelihood graph





- Problem: when $g = n$, the log-likelihood is infinite;

  Solution: information criteria (Day 4)

Recapitulation

- Density based clustering:
  - Assume a *probability density function* per cluster
  - Train using the *EM algorithm*

- Example:
  - *Mixture of Gaussians*
  - But many probability densities fit in the same framework principal component analysis, factor analysis, …

- EM algorithm:
  - problem *decomposition*: simple to implement
  - sensitive to *local minima*

BioSB

Exercise 2.29