Lecture 8, Classification, Recognition Based on Gonzales Woods, chapter 12

Karl Bengtsson Bernander

Centre for Image Analysis

Uppsala University

Image analysis fundamental steps

image acquisition

preprocessing, enhancement

segmentation

feature extraction description

Classification, interpretation, recognition

result

This lecture

- Object vs. pixelwise classification

- Template Matching for segmentation/recognition

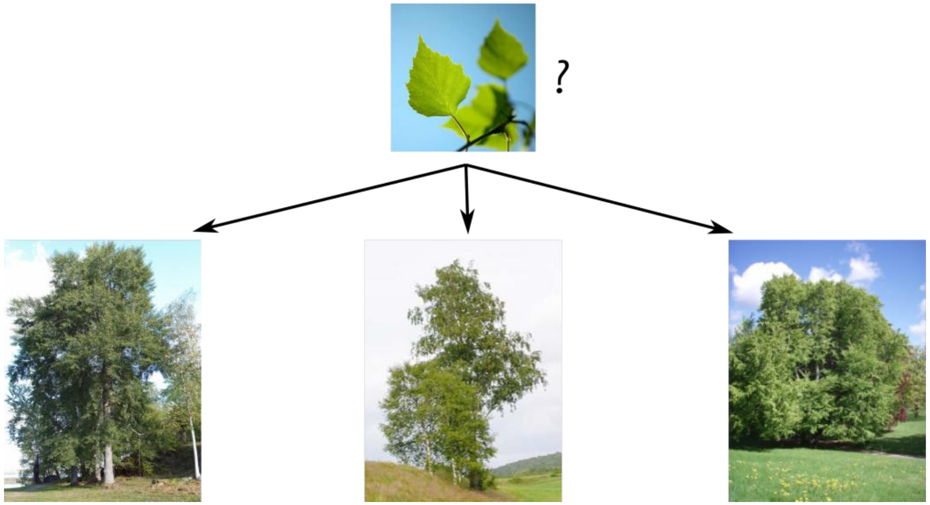
- Feature vectors and feature space, scatterplots

- Supervised classification
    - Min dist
    - Maximum likelihood
    - Decision trees
    - Neural Networks

- Unsupervised classification (clustering)
    - K-means
    - hierarchical

What is classification?

- Classification is a procedure in which individual **items** (objects/patterns/image regions/pixels) are grouped based on the similarity between the item and the description of the group.

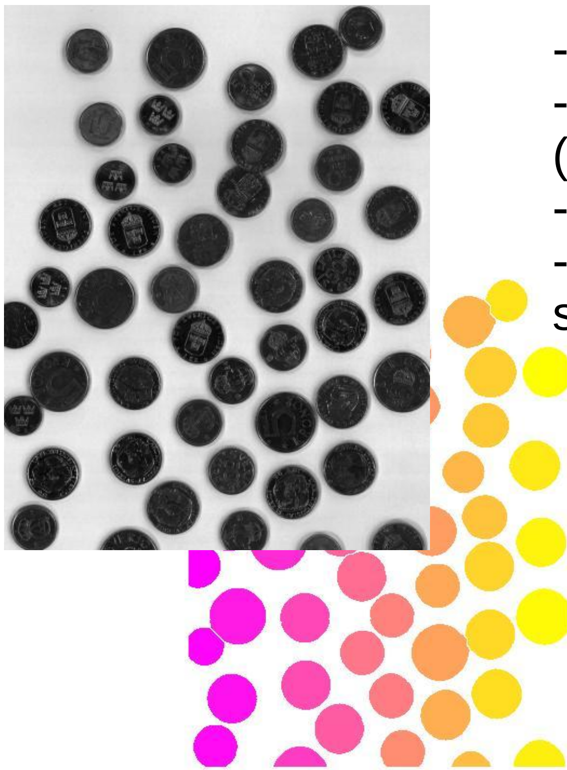


Object-wise and pixel-wise classification

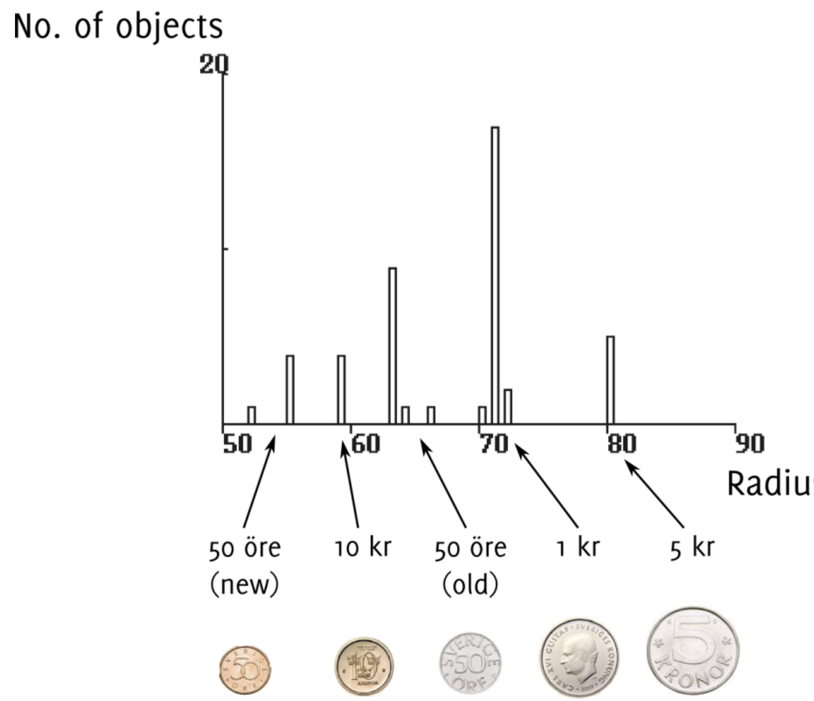
- **Object-wise classification**
  - Uses shape, size, mean intensity, mean color etc. to describe patterns.
- **Pixel-wise classification**
  - Uses intensity, color, texture, spectral information etc. (=>segmentation)

Object-wise classification

- Example



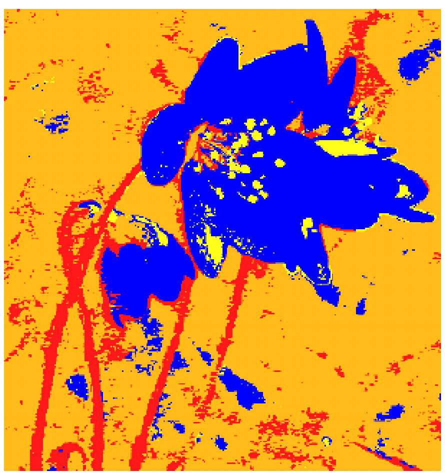
- 48 objects
- 1 feature (radius)
- 4 (5) classes
- 1-D feature space



Object-wise supervised classification

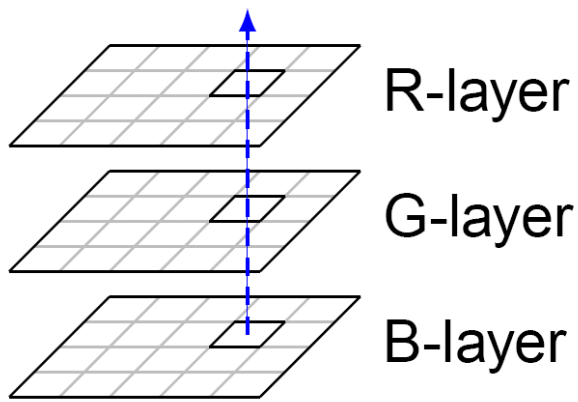
- Segment the image into regions and label them. These are the patterns to be classified.
- Extract (calculate) features for each pattern.
- Train a **classifier** on examples with known class to find **discriminant functions** in the feature space.
- For new examples decide their class using the discriminant function.

Pixel-wise classification



- Color image example.
- 256 x 256 patterns (pixels).
- 3 features (red, green and blue band).
- 4 classes (stamen, leaf, stalk and background).

$$\mathbf{x_{ij}} = [r_{ij}\, g_{ij}\, b_{ij}]^T$$

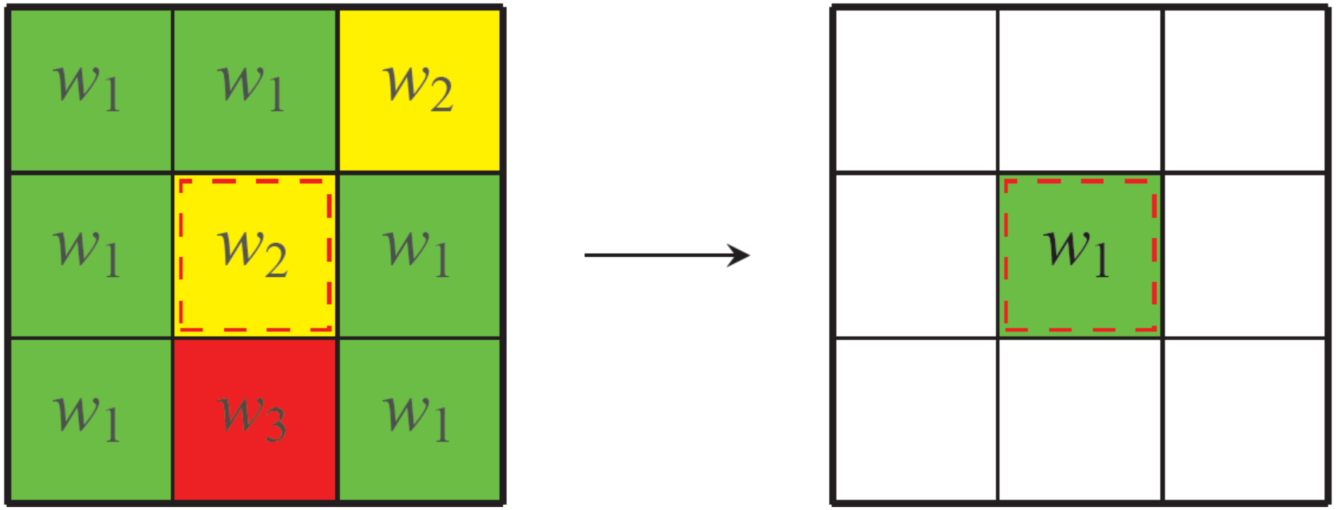


Pixel-wise classification

- The pattern is a pixel in a non-segmented image.
- Extract (calculate) features for each pattern (pixel), e.g., color, gray-level representation of texture, temporal changes.
- Train classifier.
- New samples are classified by classifier.
- Perform relaxation=majority filtering (optional).

Relaxation/majority filtering

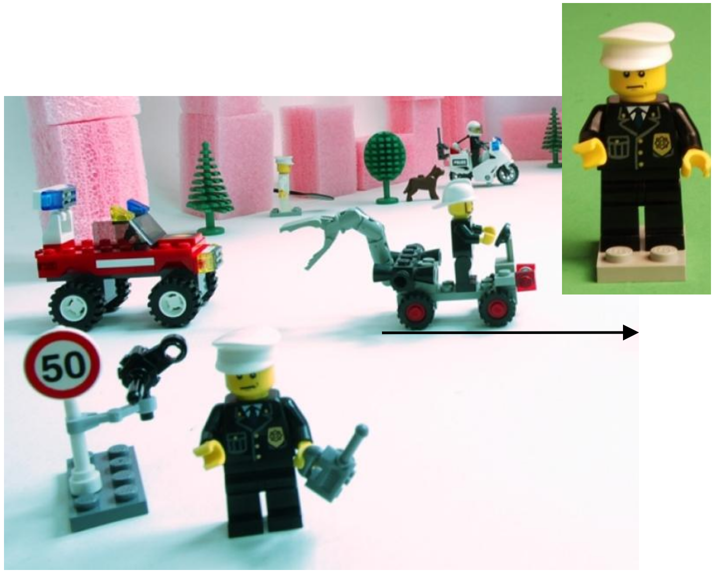
- Used in pixel-wise classification to reduce noise
- Neighborhood size determines the amount of relaxation



Matching by correlation

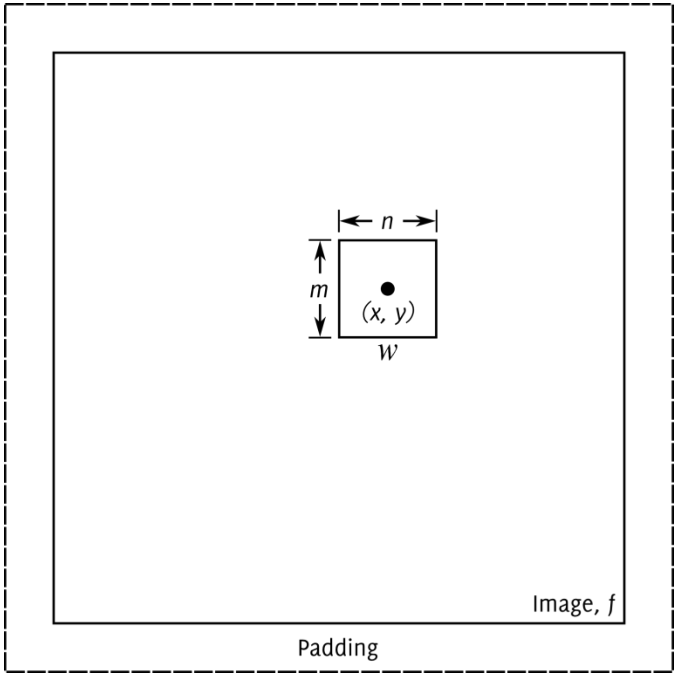
- Variant of object-wise classification.
- Locate specific objects or patterns.
- Often used for segmentation.

http://vicos.fri.uni-lj.si/leegle/



Matching by correlation

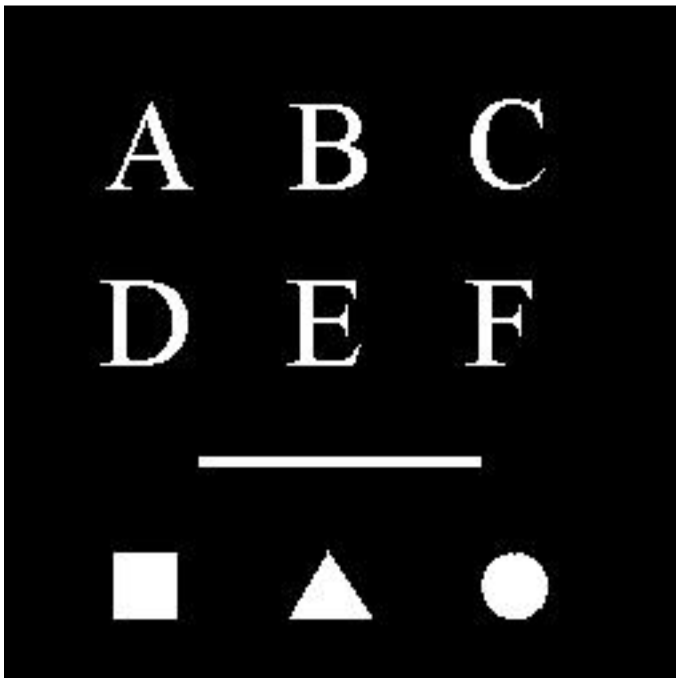
- Use correlation to match mask w(x,y) with image f(x,y).
- Slide the mask over the image and calculate correlation at each position.



$$c(x,y) = \sum_s \sum_t w(s,t) f(x+s, y+t)$$

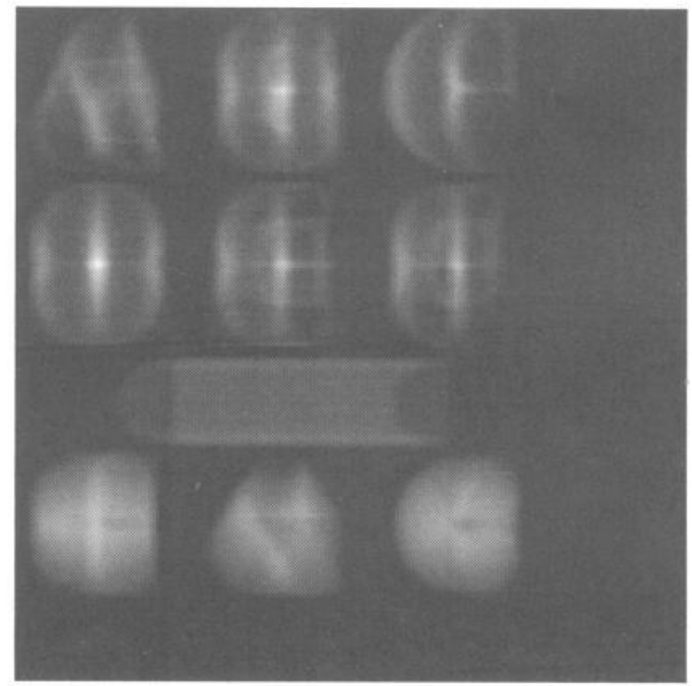
Matching by correlation

- Example



f

w



c

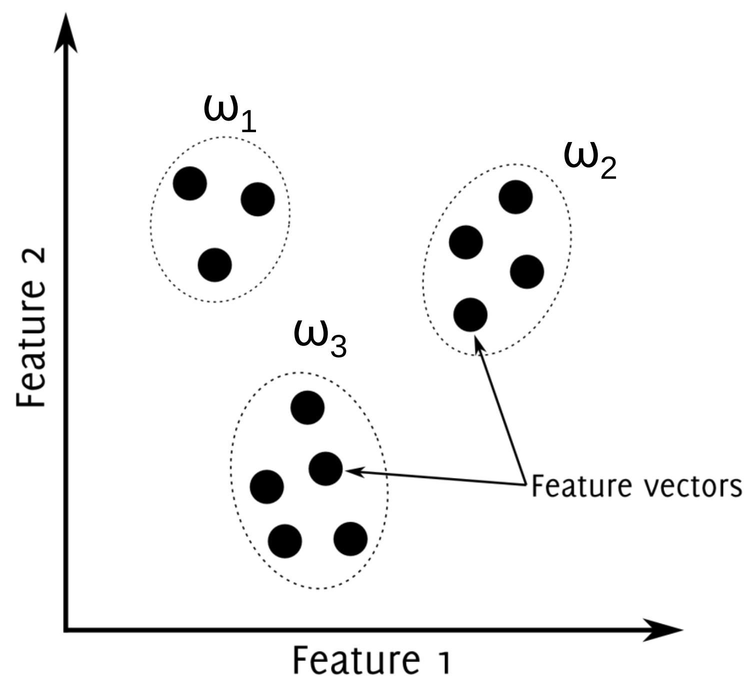
Some important concepts

- Arrangements of descriptors are often called **patterns**.
- Descriptors are often called **features**.
- The most common pattern arrangement is a **feature vector** with n dimensions.
- Patterns are placed in **classes** of objects which share common properties. A collection of W classes are denoted $\omega_{1,}\, \omega_{2,}\; \ldots.\; \omega_{W,}$

Feature vectors and feature space

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

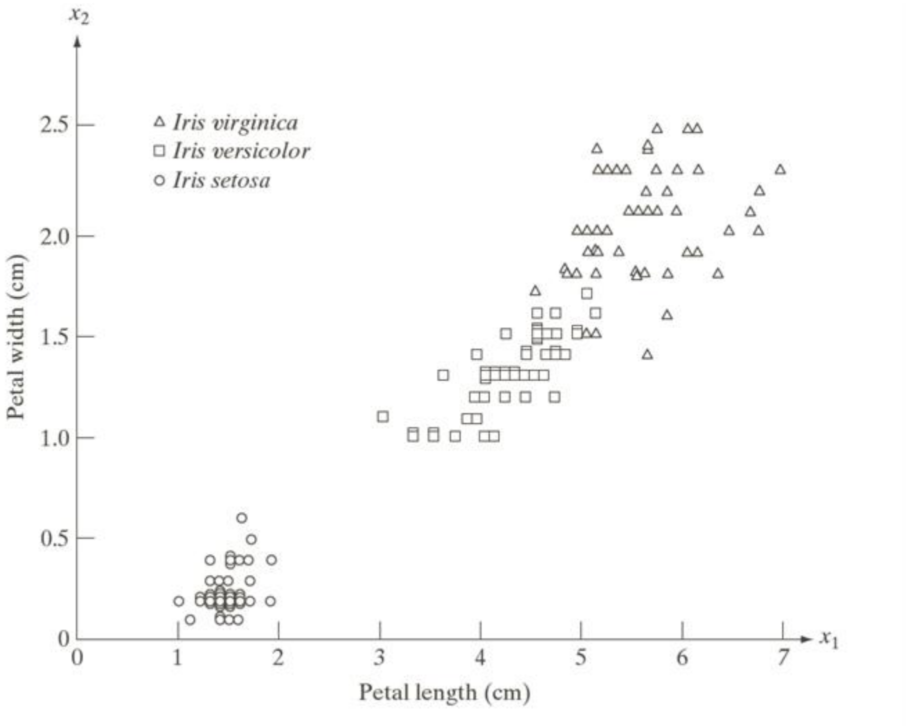
Example of a feature space (N=2)



Two features, three classes

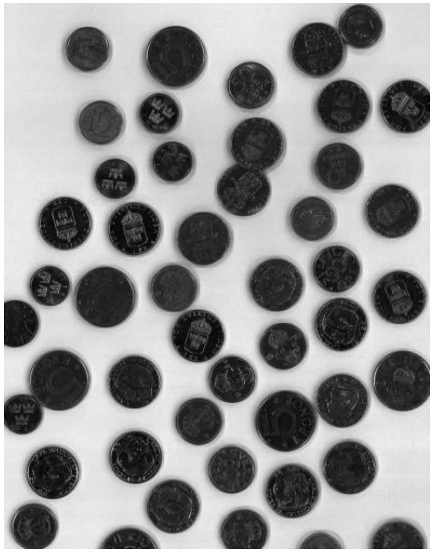
Feature vectors and feature space

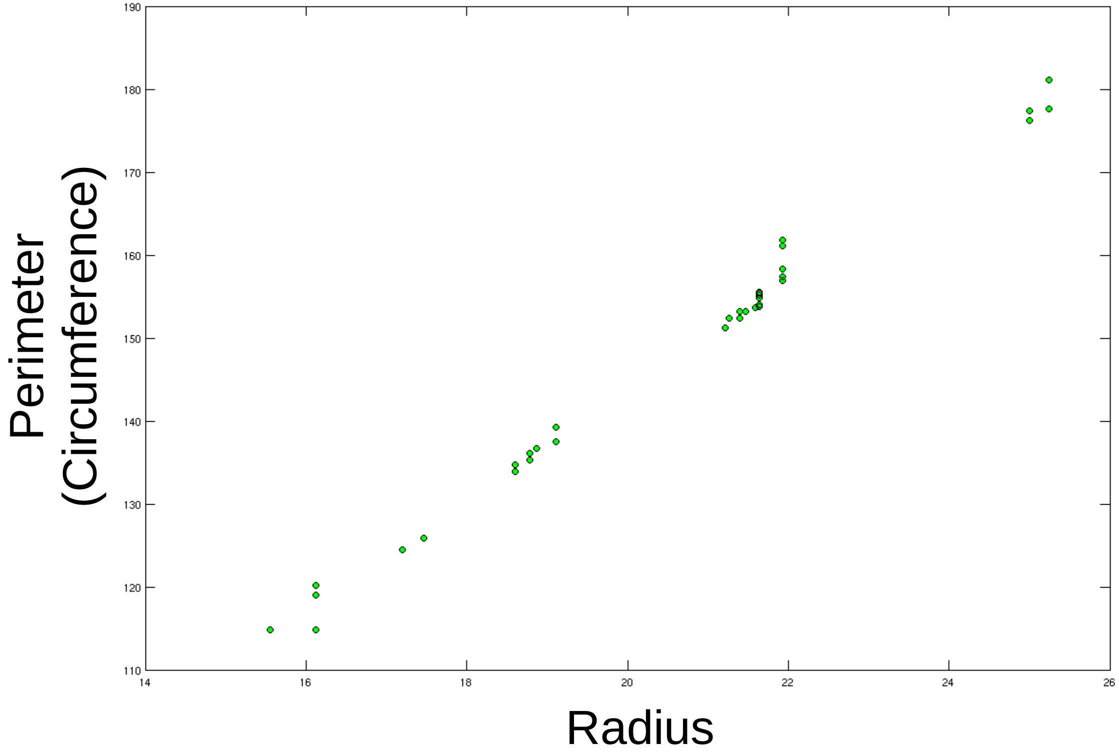
- Flower example



Scatter plots

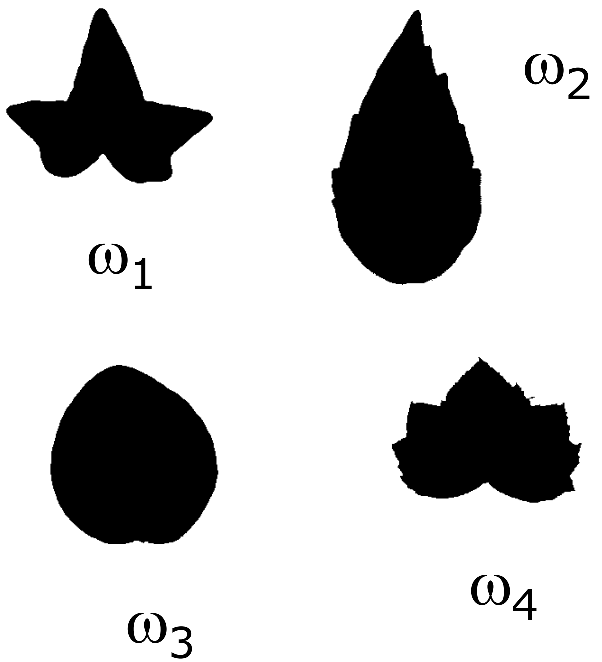
- A good way to illustrate relationships between features.

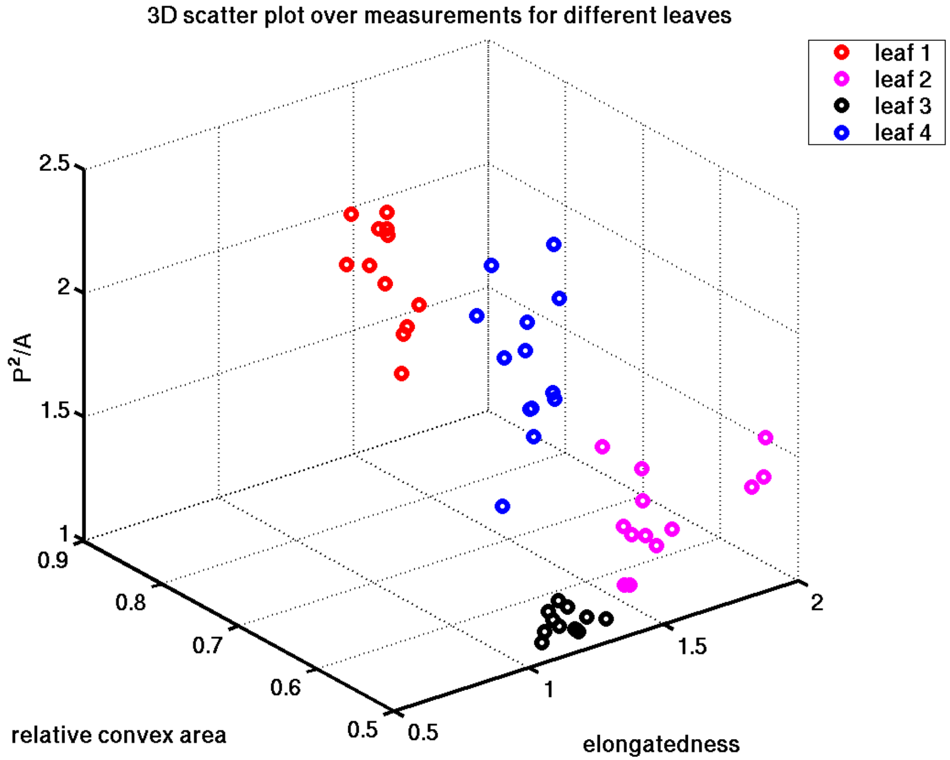




Scatter plots

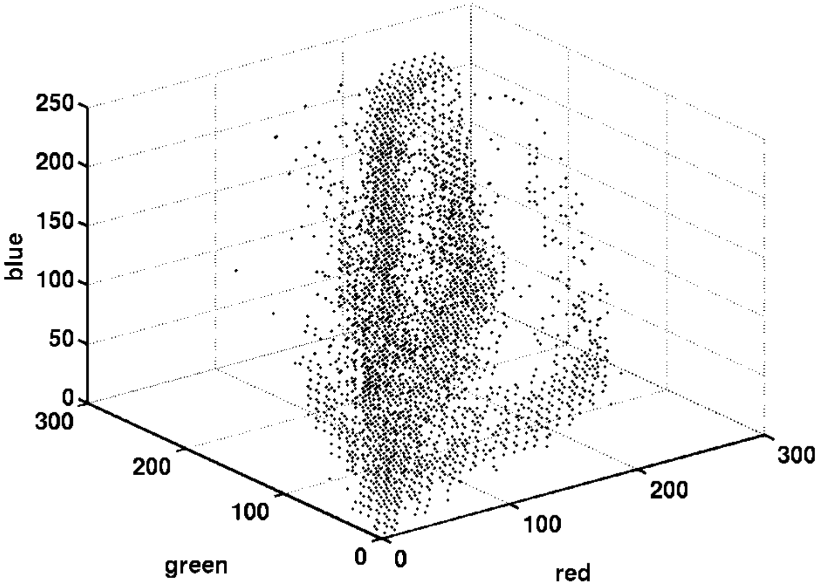
- Example of 3-dimensional plot





Scatter plots

- Example: RGB color image.

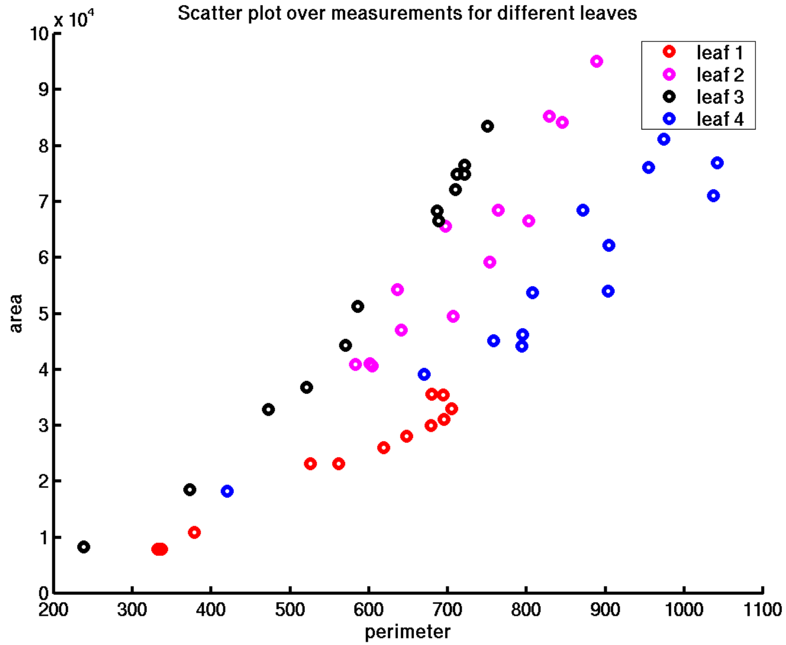


Feature selection

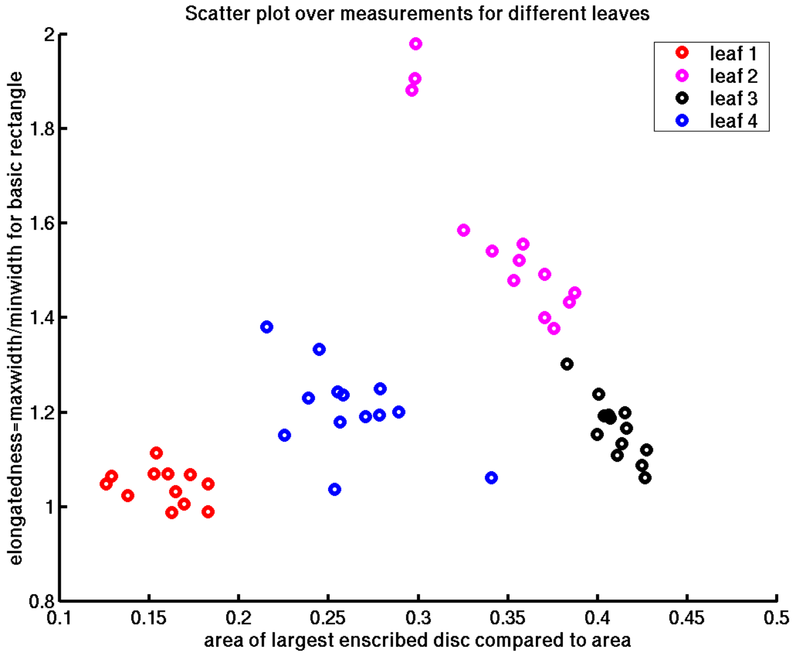
- The goal in feature selection (which is a prerequisite for ALL kinds of classification) is to find a limited set of features that can discriminate between the classes.
- Adding features without “verification” will often NOT improve the result.

Feature selection

- Some examples



Limited separation between classes



Good separation between classes

Train and classify=supervised classification

- **Training**
  - Find rules and discriminant functions that separate the different classes in the feature space using known examples.
- **Classification**
  - Take a new unknown example and put it into the correct class using the discriminant functions.

Supervised classification->
First apply knowledge, then classify

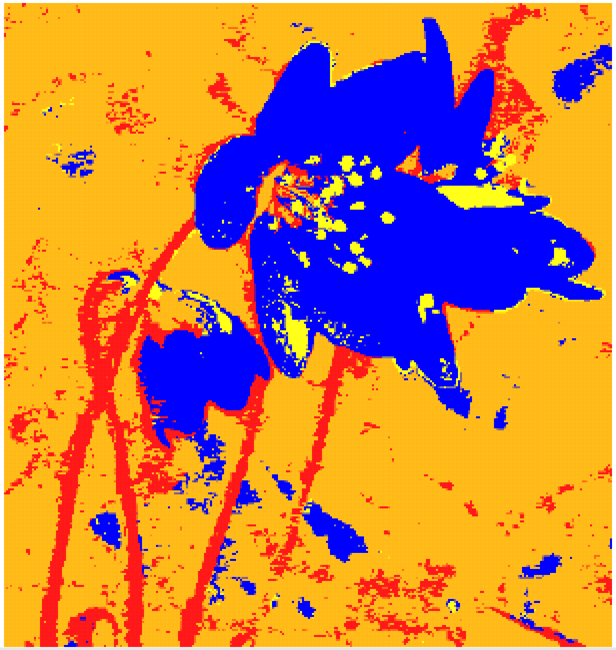
Train and classify

- Regions in the image are used as training examples (pixel-wise classification).

Original image

Training areas

Classification



Discriminant functions

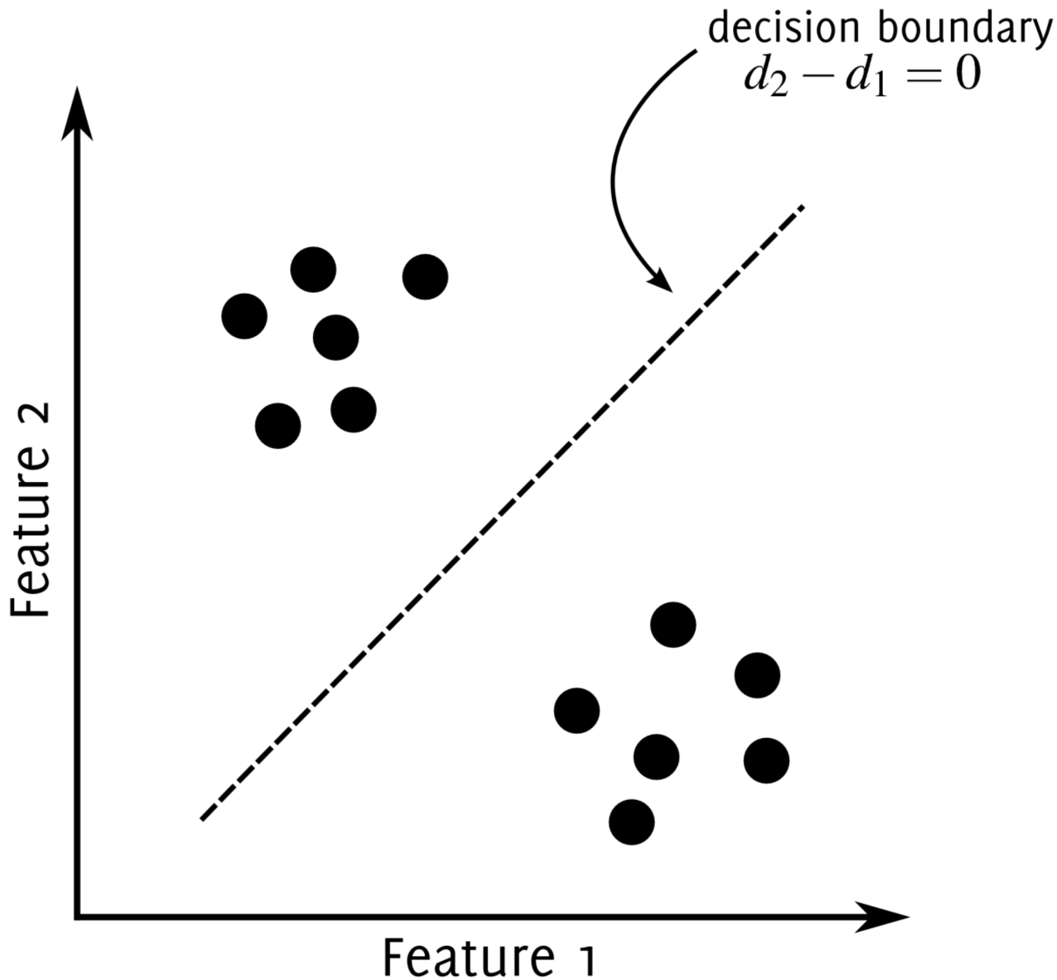
- A **discriminant function** for a class is a function that will yield larger values than functions for other classes if the pattern belongs to the class.

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, ..., W; \quad j \neq i$$

- For W pattern classes, we have W discriminant functions.
- The **decision boundary** between class i and j

$$d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

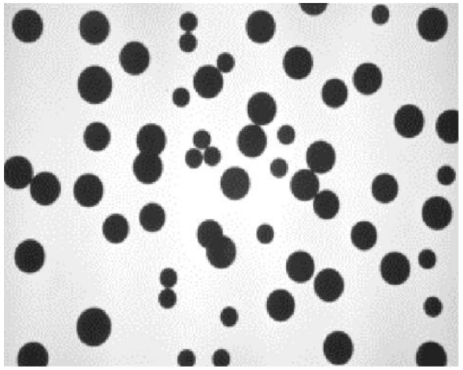
Decision boundary



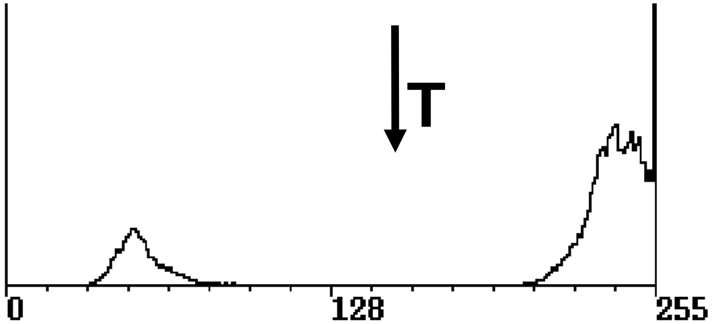
Example:Thresholding (simple classification -1D)

- Classify image into foreground and background.

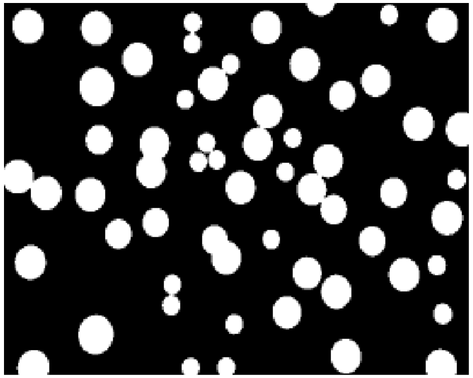
$f(x,y)$



$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T; \\ 0 & \text{if } f(x,y) < T. \end{cases}$$



Object Background

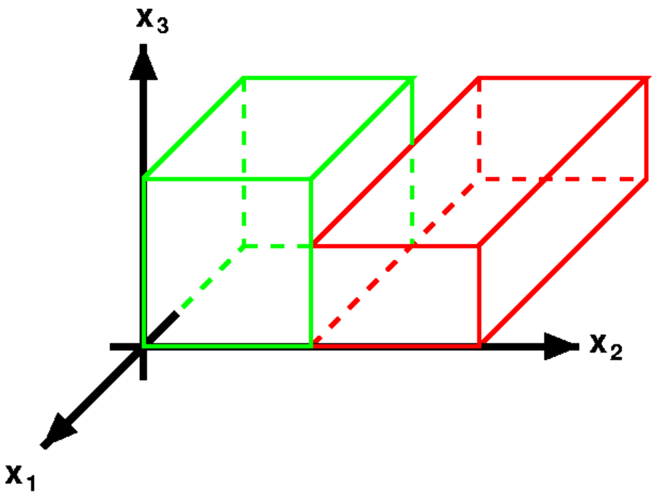


$g(x,y)$

Result: binary image

Example:Box classification

- Intervals for each class and feature
- All objects with feature vectors within the same box belong to the same class
- Generalized thresholding
  - Multispectral thresholding

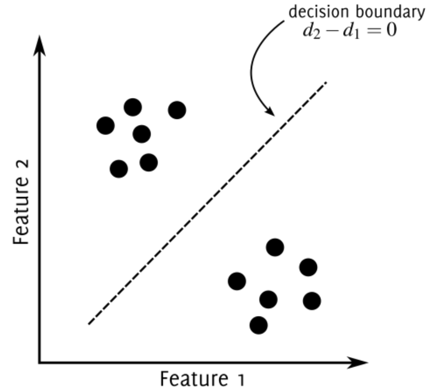


Bayesian classifiers

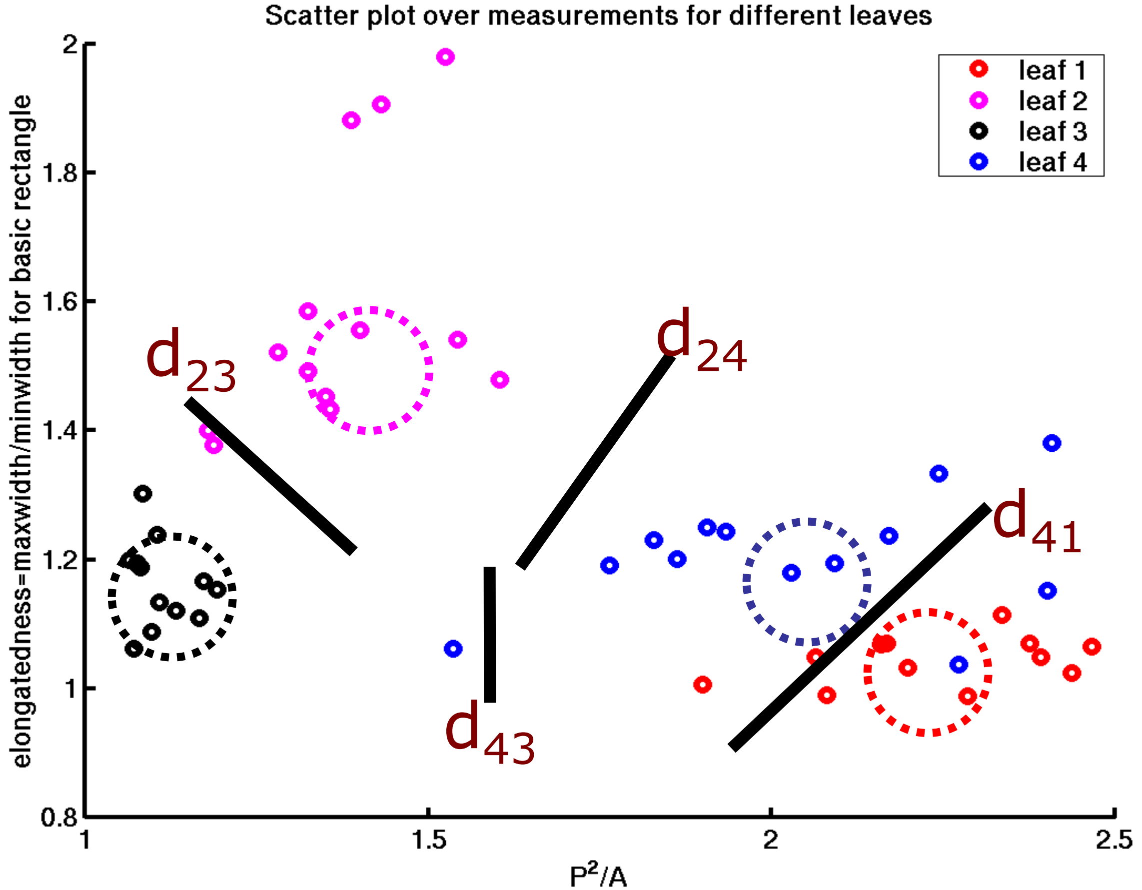
- Bayes 1702-1761
- Includes *a priori* knowledge of class probability
- Cost of errors
- Combination gives an optimum statistical classifier (in theory), minimizes total average loss
- Assumptions to simplify classifier
  - Minimum distance (MD) classifier
  - Maximum likelihood (ML) classifier

Minimum distance classifier

- Each class is represented by its mean vector
- Training is done using the objects/pixels of known class and calculate the mean of the feature vectors for the objects within each class
- New objects are classified by finding the closest mean vector



Minimum distance classifier leaves: elongatedness – P^2/A



Limitation of minimum distance classifier

Minimum distance classifier is suitable to use when:

Distance between means is **large** compared to randomness of each class with respect to its mean

Optimum performance:
distribution forms spherical hypercloud in nD pattern space

Maximum likelihood classifier

- Classify according to the greatest probability (taking variance and covariance into consideration)
- Assume that the distribution within each class is Gaussian
- The distribution within each class can be described by a mean vector and a covariance matrix

Variance & covariance

- Variance: spread or randomness for class&feature
- Covariance: influence/dependency between different features
- Described by *covariance matrix*

	feature 1	feature 2	feature 3
feature 1	1	1 & 2	1 & 3
feature 2	1 & 2	2	2 & 3
feature 3	1 & 3	2 & 3	3

Computation of covariance

- Features: x_1, x_2, x_3, ...
- Feature vector for object i: $x_{1,i}$, $x_{2,i}$, $x_{3,i}$, ...
- Mean for each feature (and class): x_{mean_1} x_{mean_2} x_{mean_3} ...

$$\mathrm{cov}\left(x_i, x_j\right) = \frac{1}{n-1} \sum_{k=1}^{n} \left(x_{i,k} - x_{mean_i}\right) \cdot \left(x_{j,k} - x_{mean_j}\right)$$

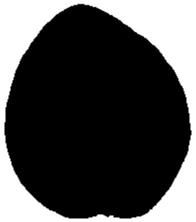
Mean vector for each class

Computed from training data

$$\begin{array}{l} \text{feature 1} \\ \text{feature 2} \\ \text{feature 3} \end{array} \quad \begin{pmatrix} 1.0441 \\ 2.2383 \\ 0.7008 \end{pmatrix} \quad \begin{pmatrix} 1.5855 \\ 1.3763 \\ 0.5275 \end{pmatrix} \quad \begin{pmatrix} 1.1654 \\ 1.3763 \\ 0.5275 \end{pmatrix} \quad \begin{pmatrix} 1.2073 \\ 2.0342 \\ 0.5756 \end{pmatrix}$$

ω_1

ω_2



ω_3

ω_4

Covariance matrix for the leaves

$$C_{\omega_1} = \begin{pmatrix} 0.0014 & 0.0061 & 0.0011 \\ 0.0061 & 0.0299 & 0.0052 \\ 0.0011 & 0.0052 & 0.0010 \end{pmatrix}$$

$$C_{\omega_3} = \begin{pmatrix} 0.0042 & 0.0026 & 0.0002 \\ 0.0026 & 0.0018 & 0.0001 \\ 0.0002 & 0.0001 & 0.0000 \end{pmatrix}$$

$$C_{\omega_2} = \begin{pmatrix} 0.0409 & 0.0244 & 0.0034 \\ 0.0244 & 0.0163 & 0.0022 \\ 0.0034 & 0.0022 & 0.0003 \end{pmatrix}$$

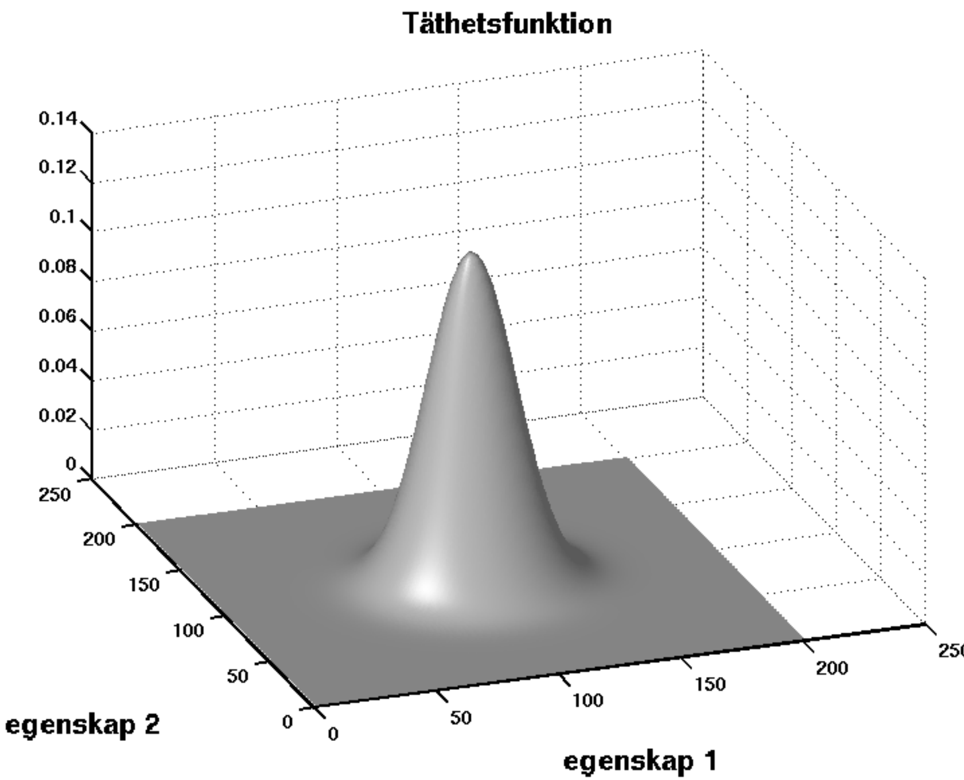
$$C_{\omega_4} = \begin{pmatrix} 0.0088 & 0.0232 & 0.0015 \\ 0.0232 & 0.0680 & 0.0042 \\ 0.0015 & 0.0042 & 0.0003 \end{pmatrix}$$

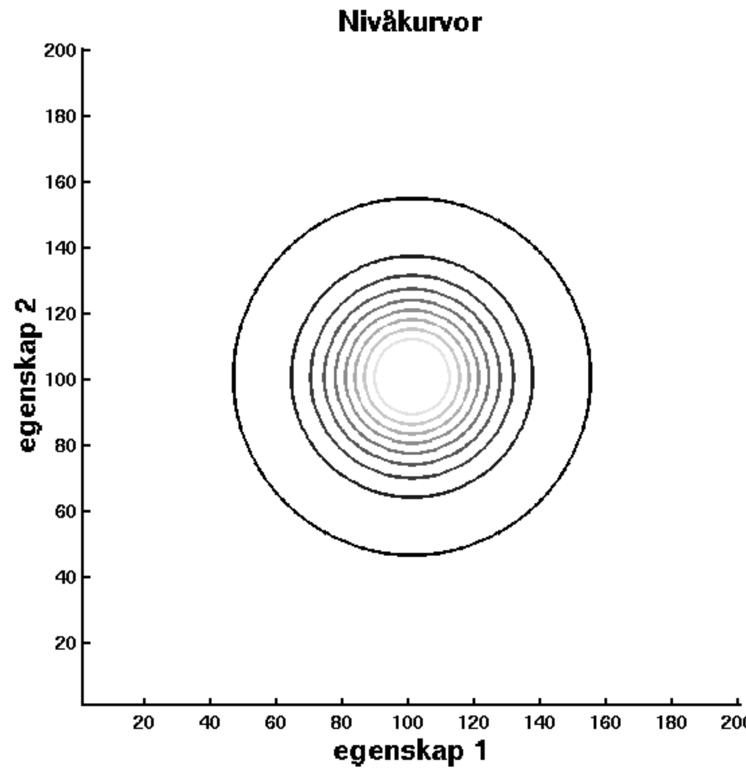
x_1=elongatedness
x_2=P^2/A
x_3=relative convex area

$$C_{\omega_i} = \begin{pmatrix} \operatorname{cov}(x_1, x_1) & \operatorname{cov}(x_1, x_2) & \operatorname{cov}(x_1, x_3) \\ \operatorname{cov}(x_1, x_2) & \operatorname{cov}(x_2, x_2) & \operatorname{cov}(x_2, x_3) \\ \operatorname{cov}(x_1, x_3) & \operatorname{cov}(x_2, x_3) & \operatorname{cov}(x_3, x_3) \end{pmatrix}$$

Density function

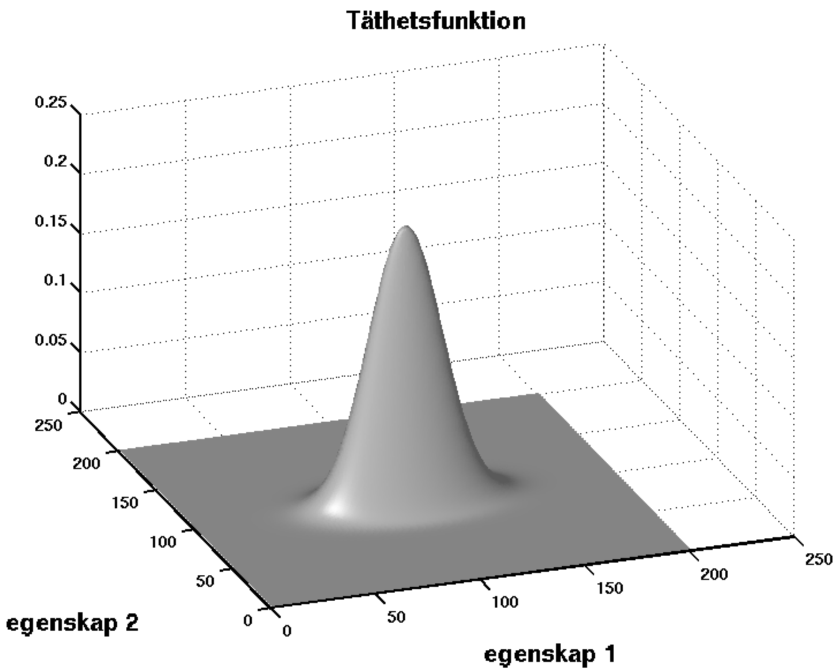
Equal variance

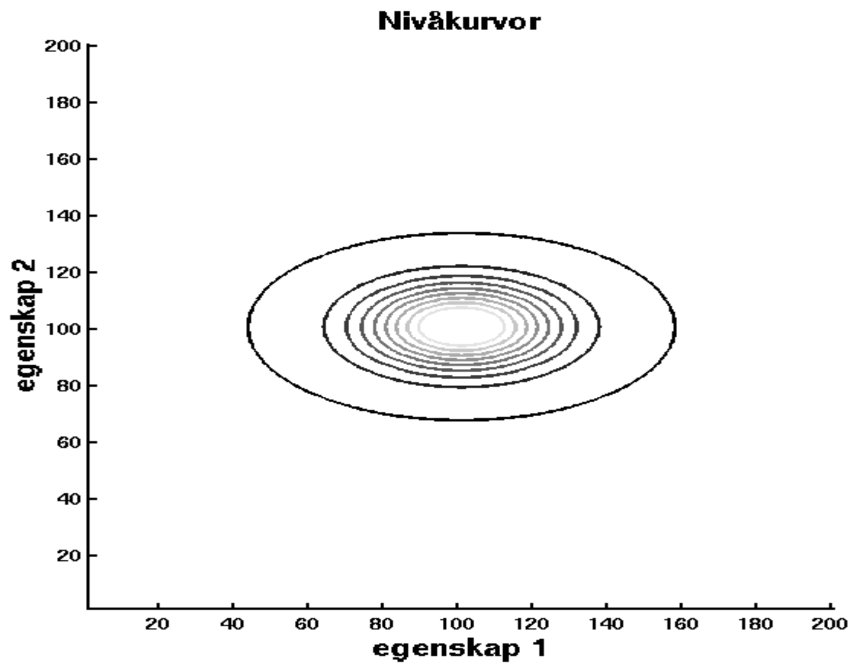




Density function

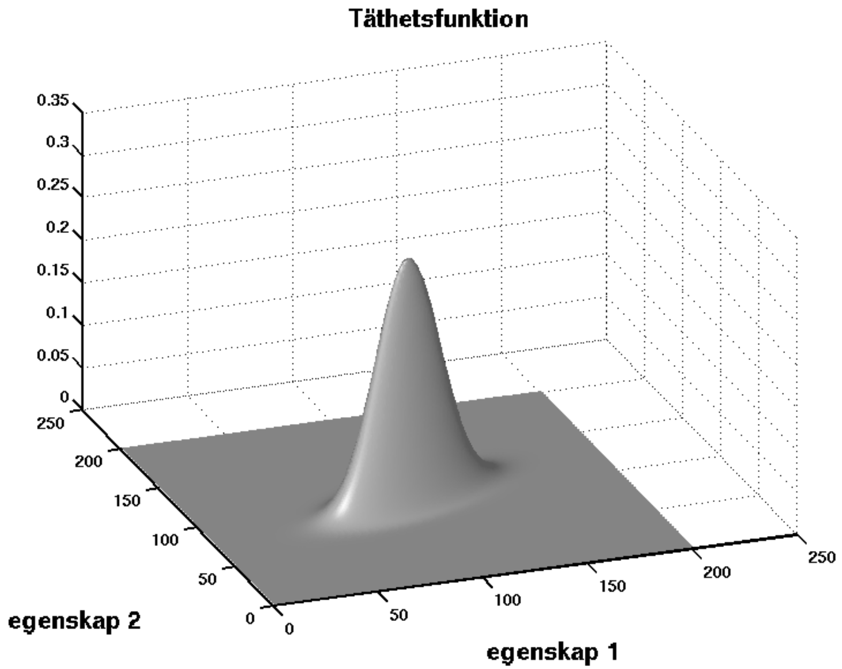
Different variance

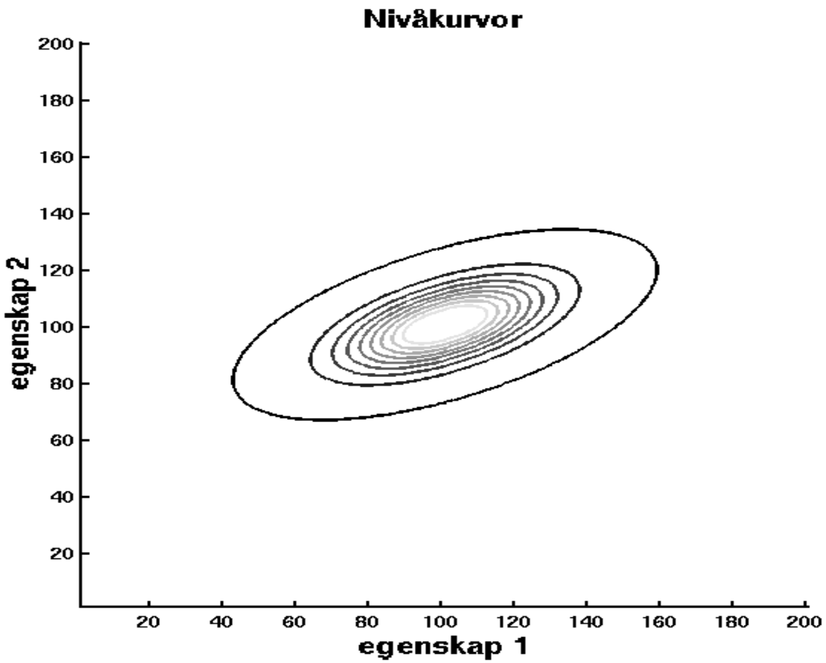




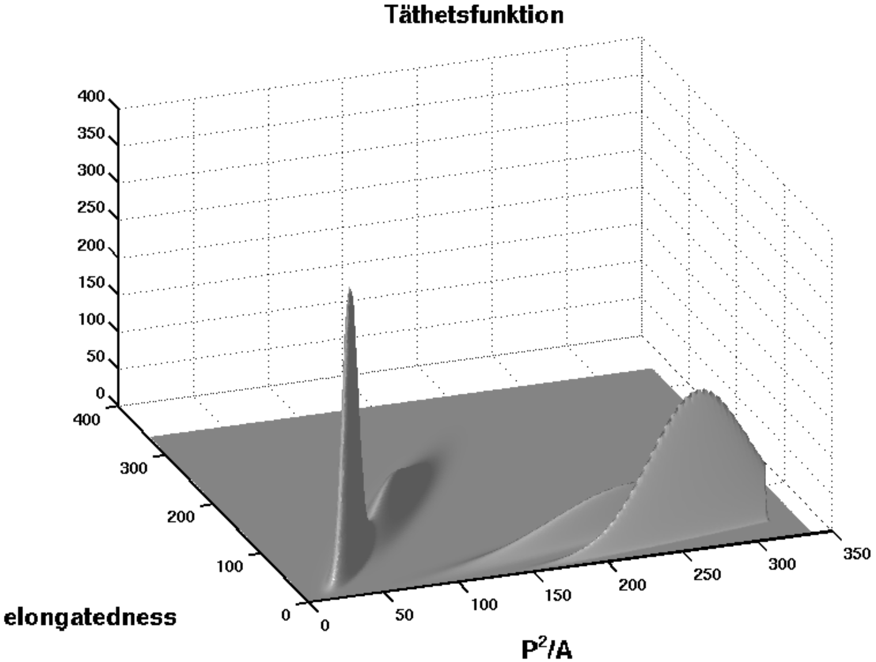
Density function

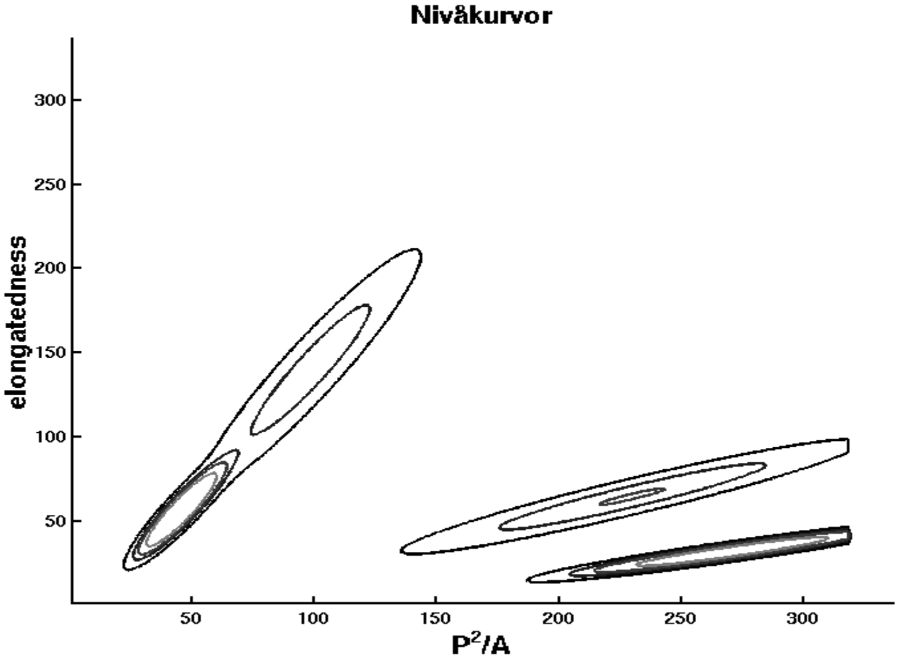
Covariance≠0





Likelihood
Leaves: elongatedness – P^2/A





Assumptions on covariance matrix

- Case 1 (**MinDist**)
  - independent features → no covariance
  - equal variance

$$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - \frac{1}{2}\mathbf{m}_j \mathbf{m}_j$$

- Case 3 (**EqualCovar**)
  - same covariance for all classes

Case 2 (**UnCorrelated**)

independent features → no covariance

different variance for different features

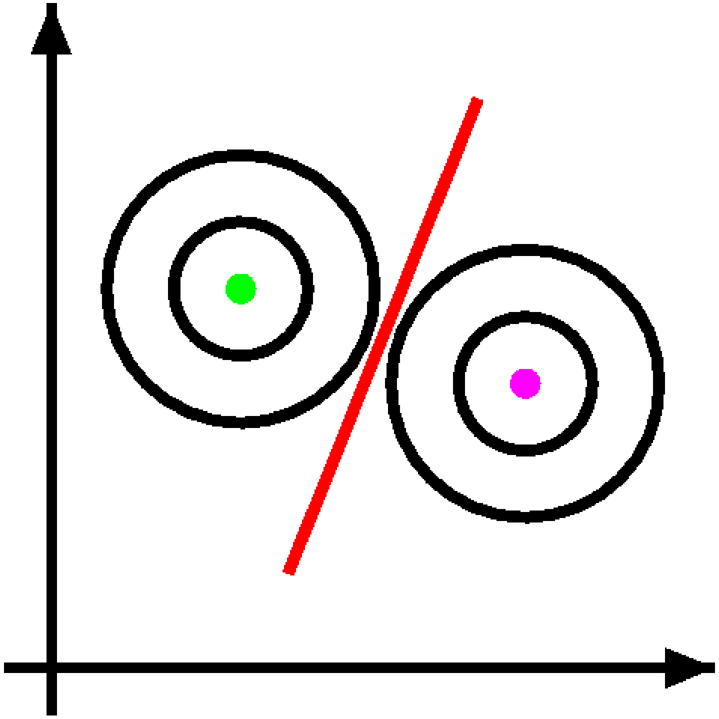
Case 4 (**General**)

different covariance matrices for alla classes

$$d_j(\mathbf{x}) = \ln P(\omega_j) + \mathbf{x}^T \mathbf{C}^{-1} \mathbf{m}_j - \mathbf{m}_j^T \mathbf{C}^{-1} \mathbf{m}_j$$

Case 1

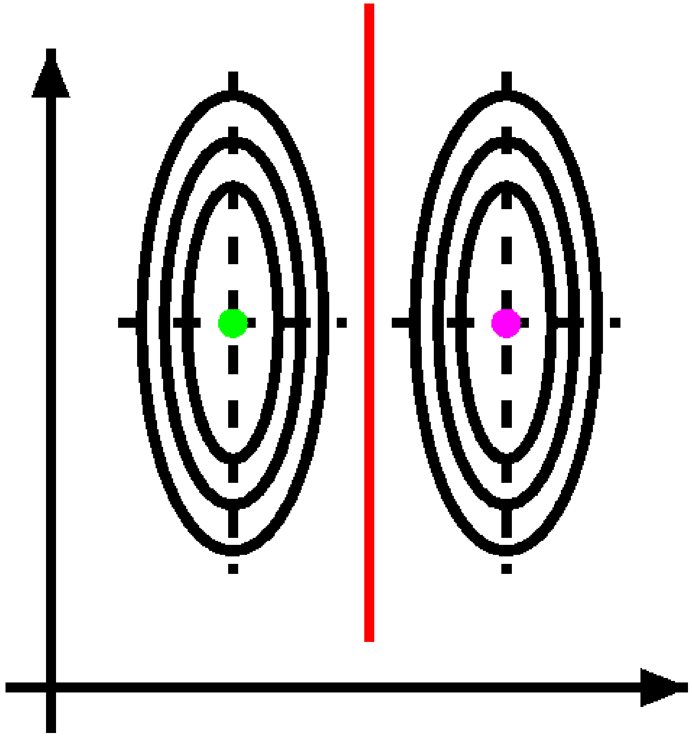
Minimum distance classifier:
- independent features
- equal variance



Case 2

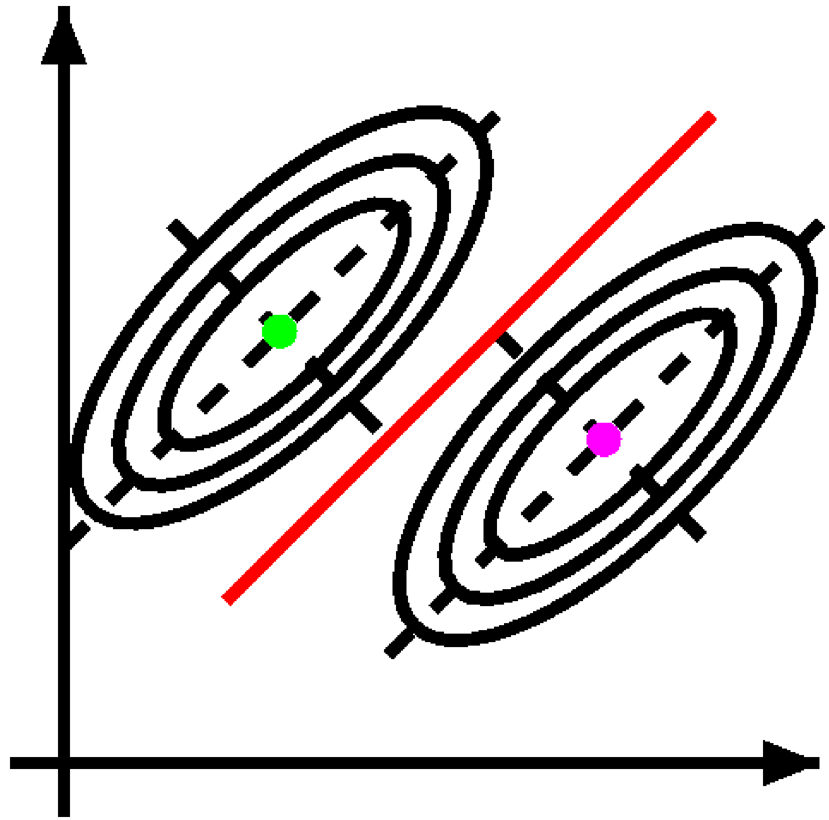
Uncorrelated features:

- independent features
- different variance for different features

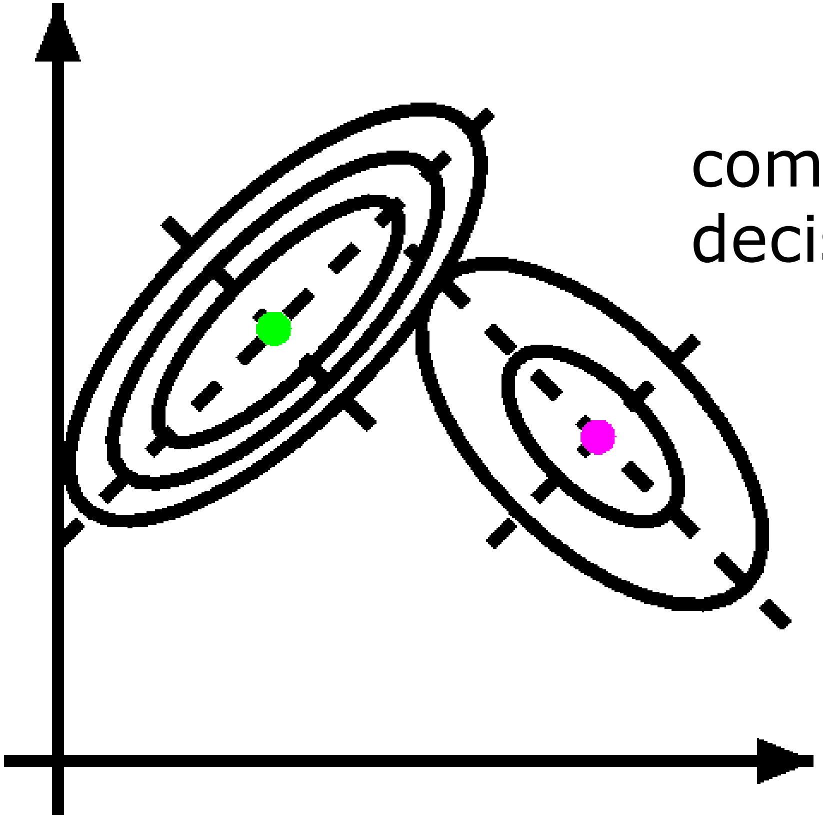


Case 3

equal covariance for all classes



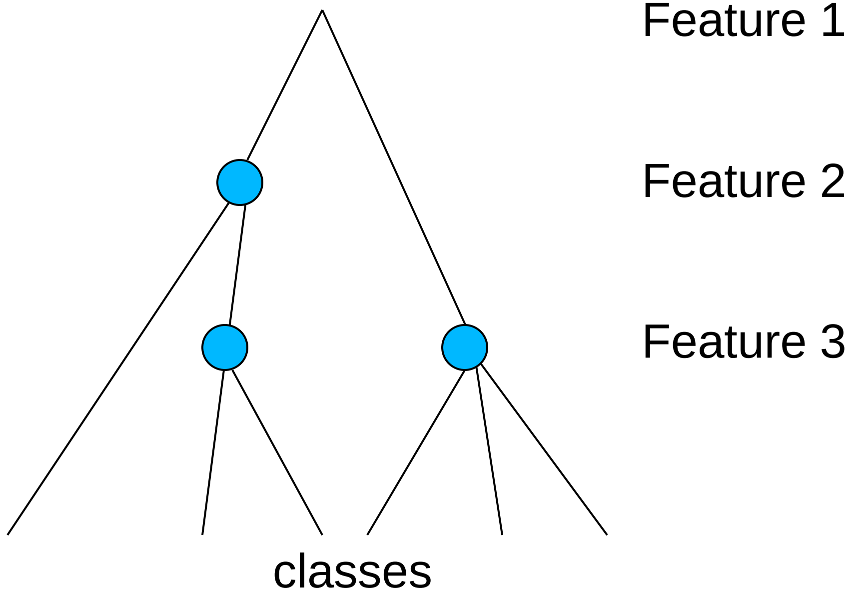
Case 4 (general)



complicated
decision boundaries

Decision Tree

- Divide samples into classes by ”thresholding” one feature at a time
- Training algorithms/automtic tree constructing algorithms exist

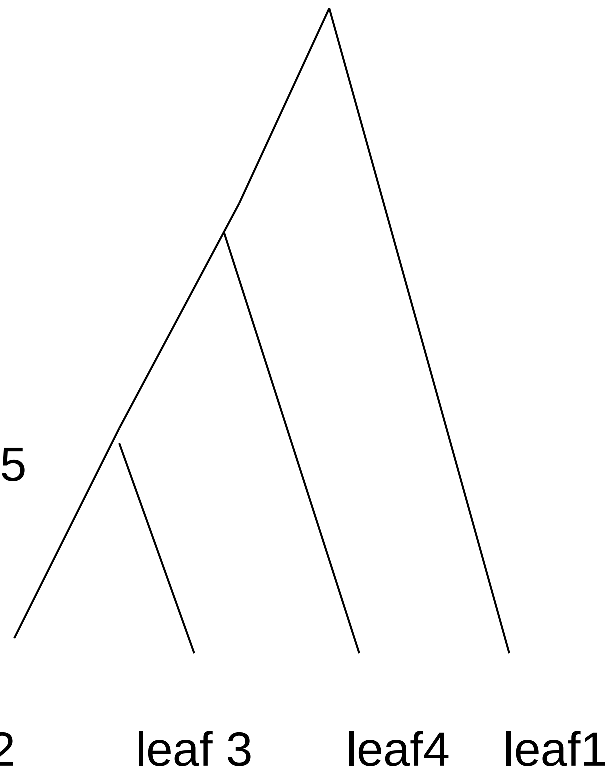


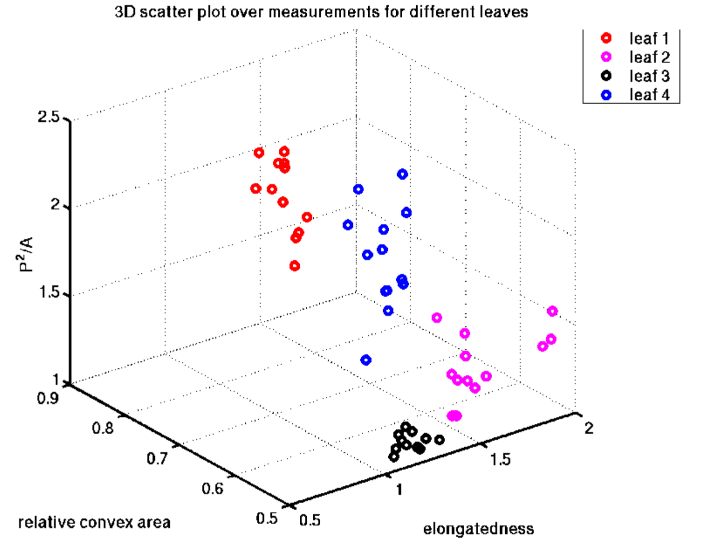
Decision tree example

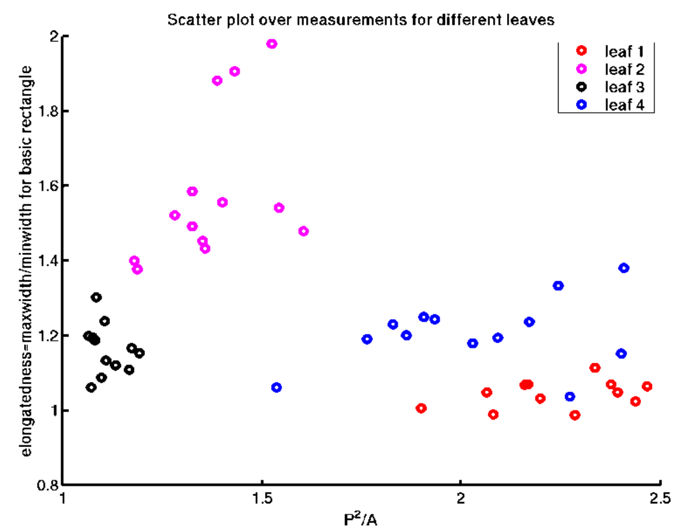
F3 <0.6, F3>0.6

F2<1.5, F2>1.5

F1<1.25,F1>1.25







Artificial Neural Networks (ANNs)

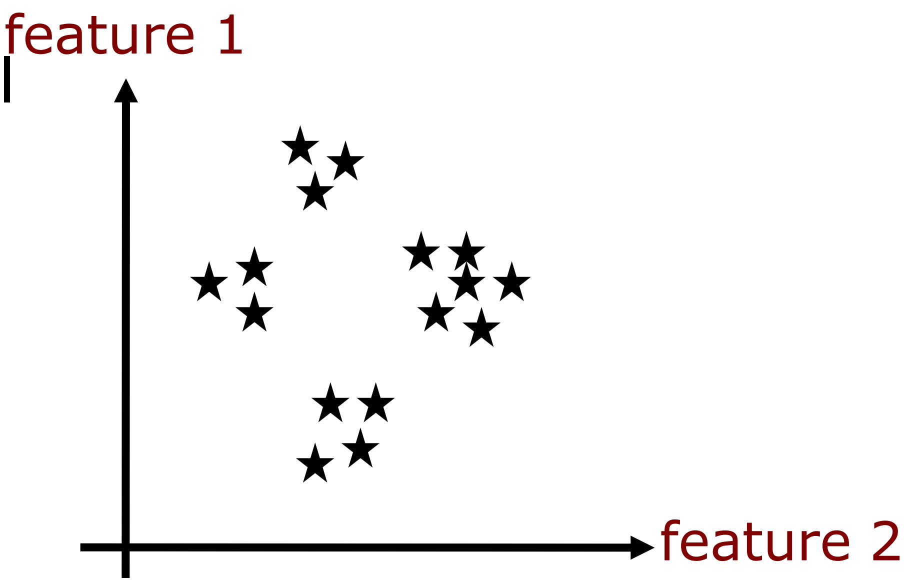
- Create a classifier by adaptive development of coefficients for decisions found via training.
- Do not assume a normal (Gaussian) probability distribution.
- Simulate the association of neurons in the brain.
- Can draw decision borders in feature space that are more complicated than hyper quadratics.
- Require careful training

About trained (supervised) systems

- The features should be based on their ability to separate the classes
- Addition of new features may lead to decreased performance
- The training data should be much larger than the number of features
- Linearly dependent features should be avoided

Unsupervised classification: cluster analysis

- Divide feature space into clusters based on the mutual similarity of the subset elements
- Explorative analysis
- After clustering :
  - Compare with reference data
  - Identify the classes



Unsupervised classification: First classify, then apply knowledge

Unsupervised methods (clustering)

- *k*-means
  - Top down approach (divisive)
  - Predetermined number of clusters
  - Tries to find natural centers in the data
  - Result difficult to illustrate for more than 3 dimensions

Hierarchical

Most often bottom up approach (agglomerative)

Merges patterns until all are one class

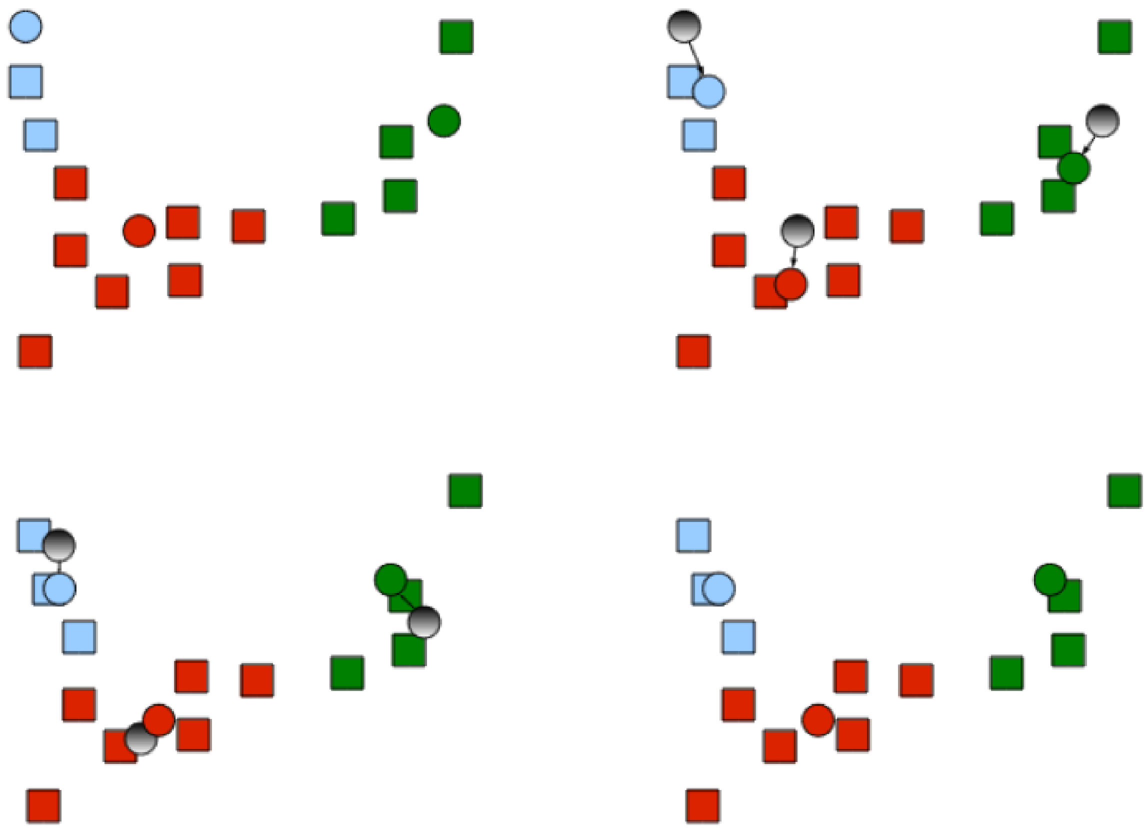
The user decides which clusters are natural

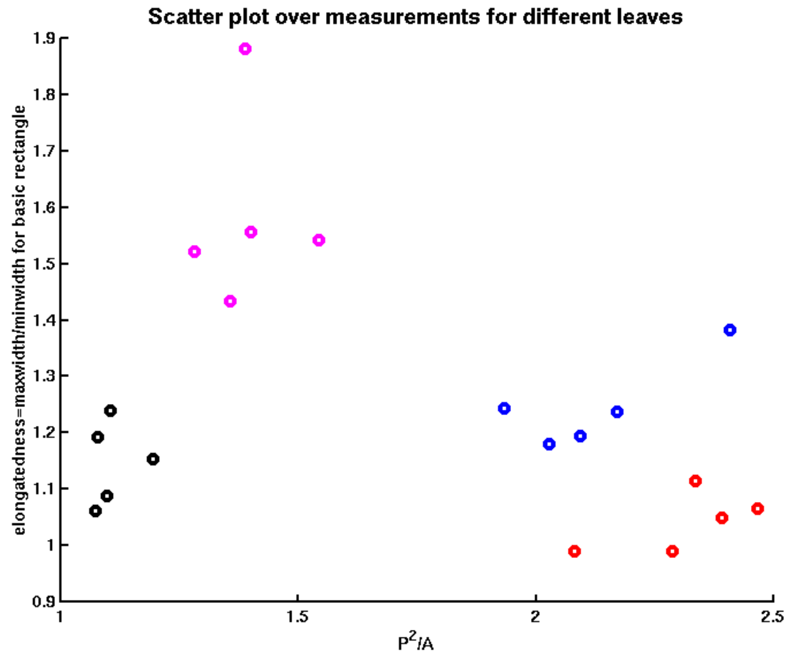
Illustrates results through a dendrogram

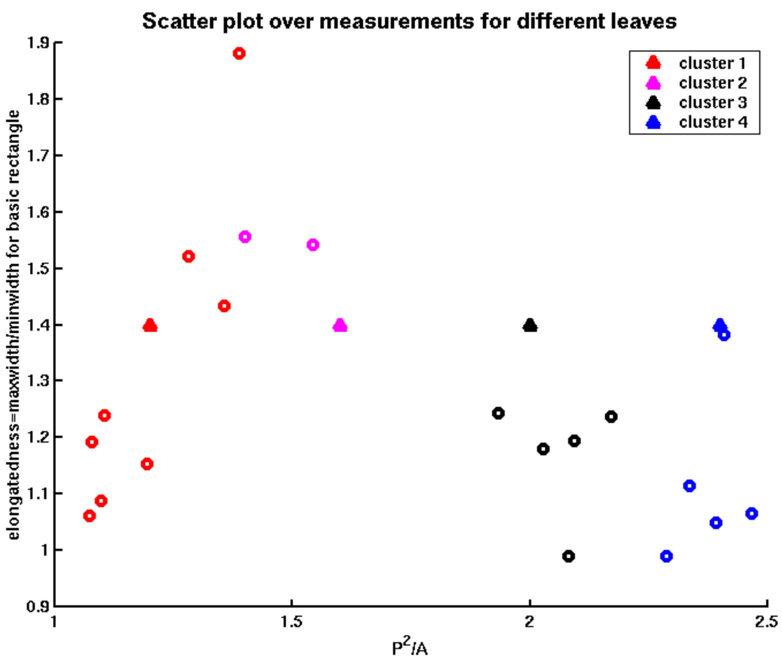
k-means clustering

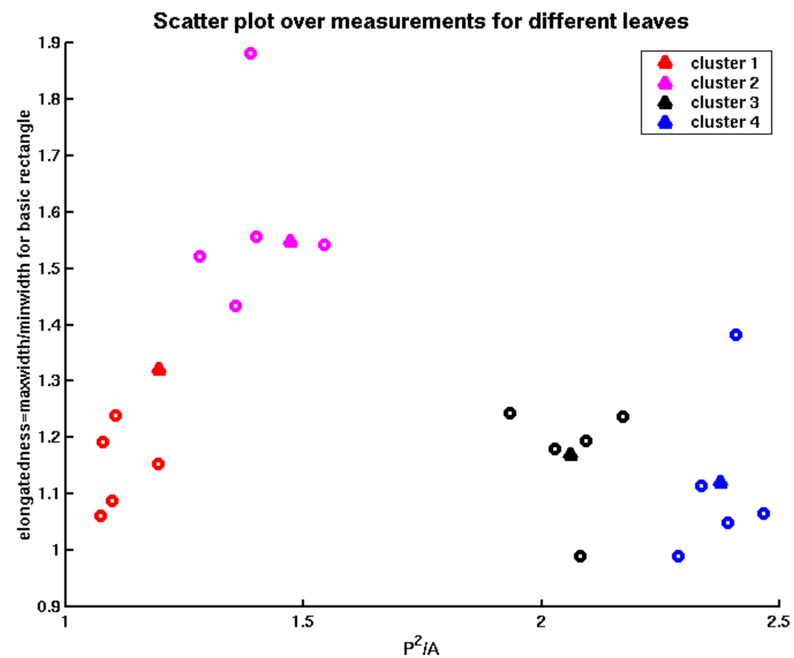
- number of clusters = k
- initialize k starting points (randomly or according to some distribution)
- assign each object to the closest cluster and recompute the centre for that cluster
- move objects between the clusters in order to
  - minimize the variance within each cluster
  - maximize the variance between clusters

k-means clustering




Scatter plot over measurements for different leaves
elongatedness=maxwidth/minwidth for basic rectangle
P^2/A
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
1
1.5
2
2.5


Scatter plot over measurements for different leaves
cluster 1
cluster 2
cluster 3
cluster 4
elongatedness=maxwidth/minwidth for basic rectangle
P^2/A


Scatter plot over measurements for different leaves
cluster 1
cluster 2
cluster 3
cluster 4
elongatedness=maxwidth/minwidth for basic rectangle
P^2/A

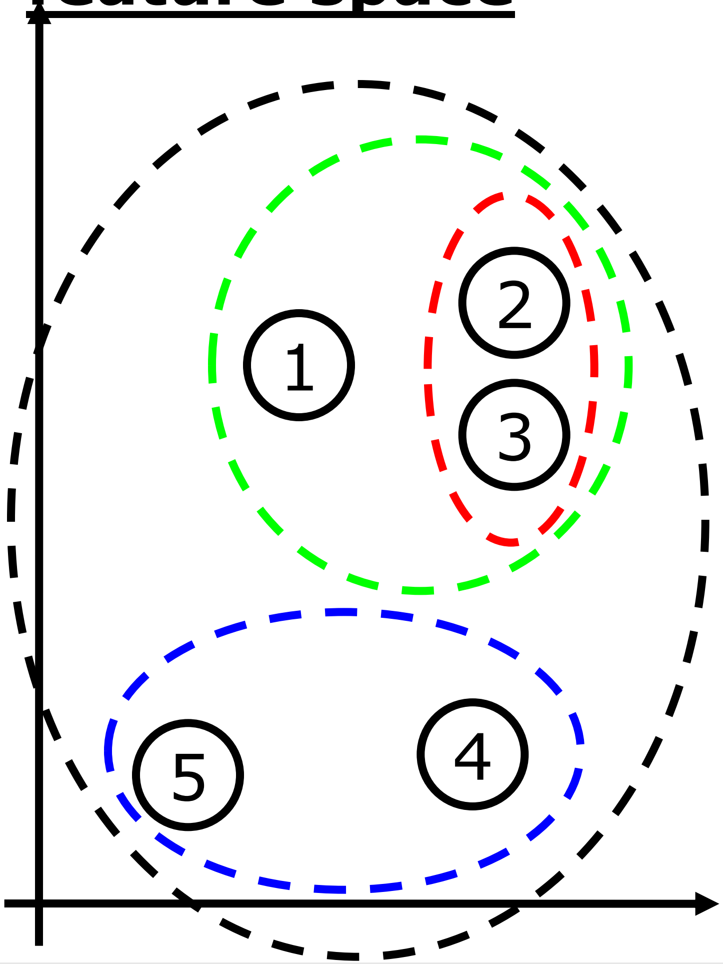
Hierarchical clustering

construct clustering tree (dendrogram)

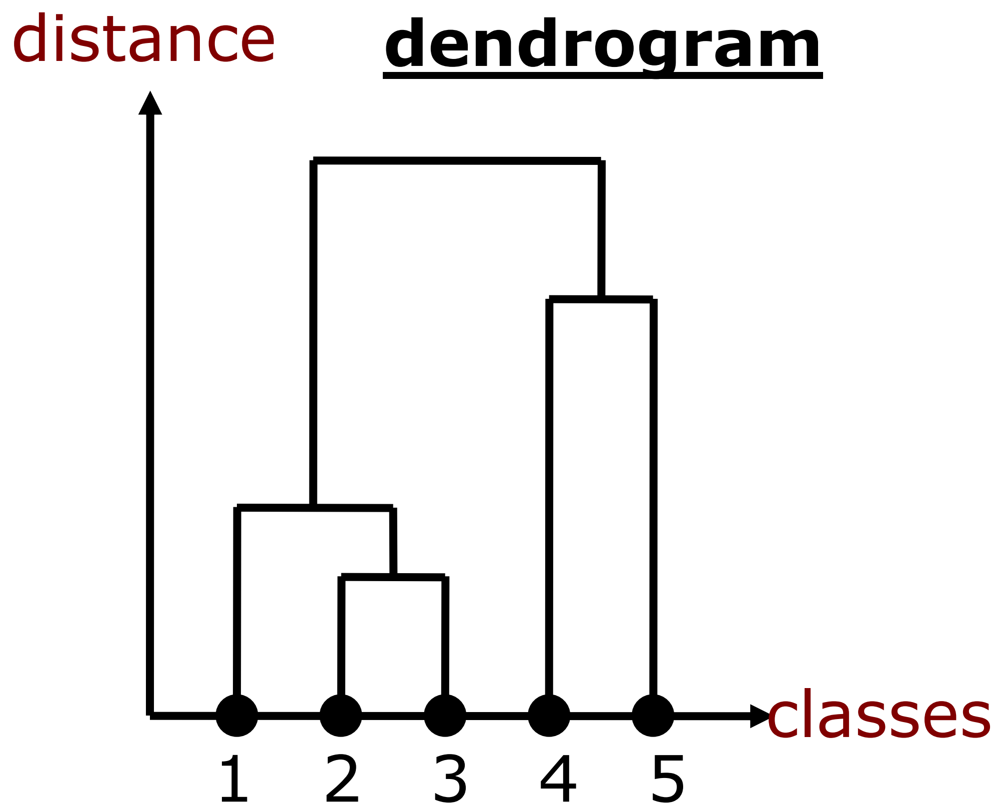
- start with each object/pixel as its own class
- merge the classes that are closest according to some distance measure
- continue until only one class is achieved
- decide the number of classes based on the distances in the tree

Simple dendrogram

feature space

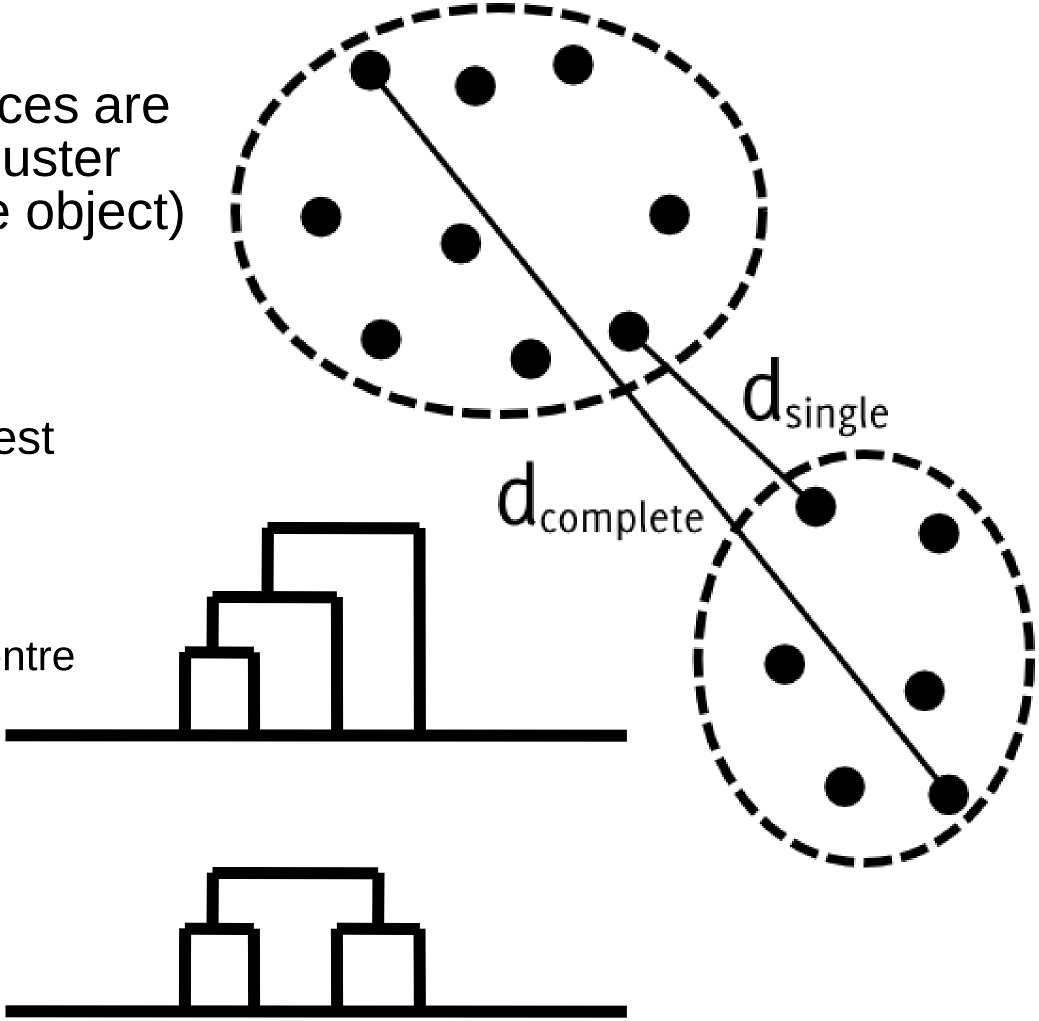


dendrogram

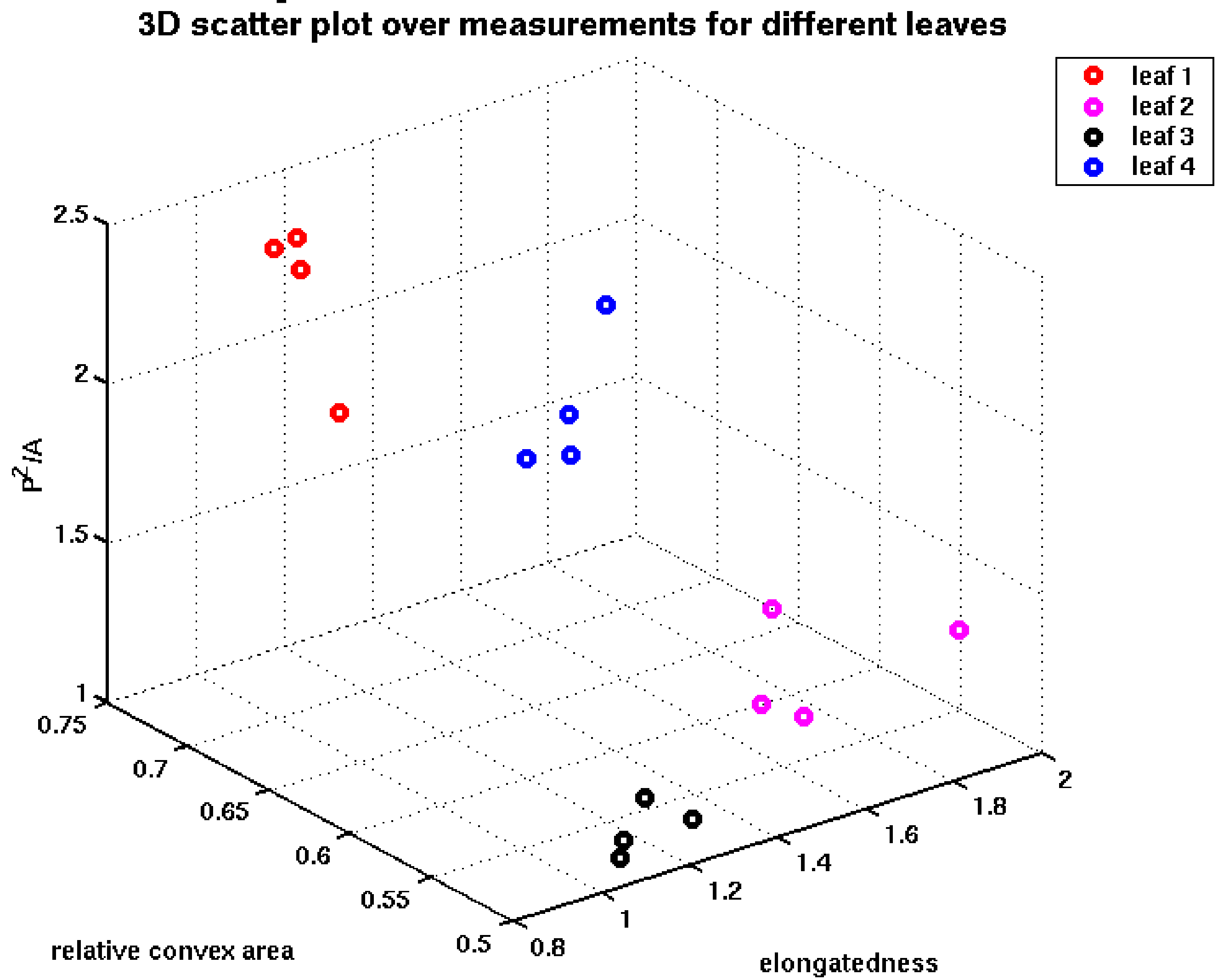


Linking rules

- Linking rules (how distances are measured when the cluster contain more than one object)
  - single linkage (nearest neighbour)
    - shortest distance
  - complete linkage (furthest neighbour)
    - longest distance
  - mean linkage
    - distance to cluster centre

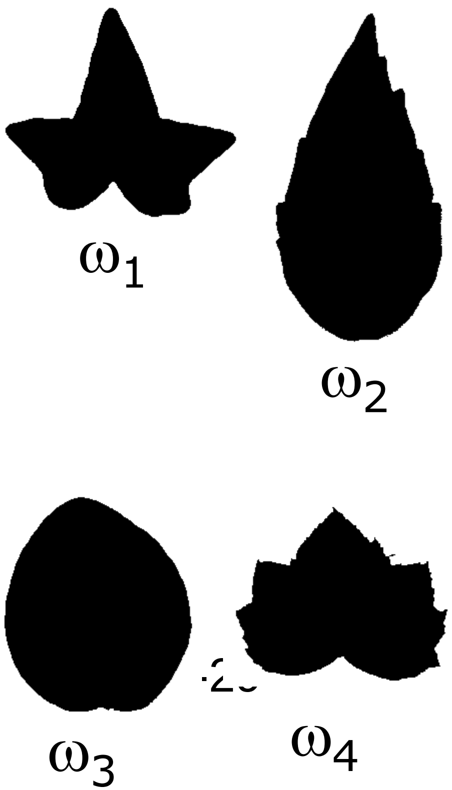


scatterplot for used features

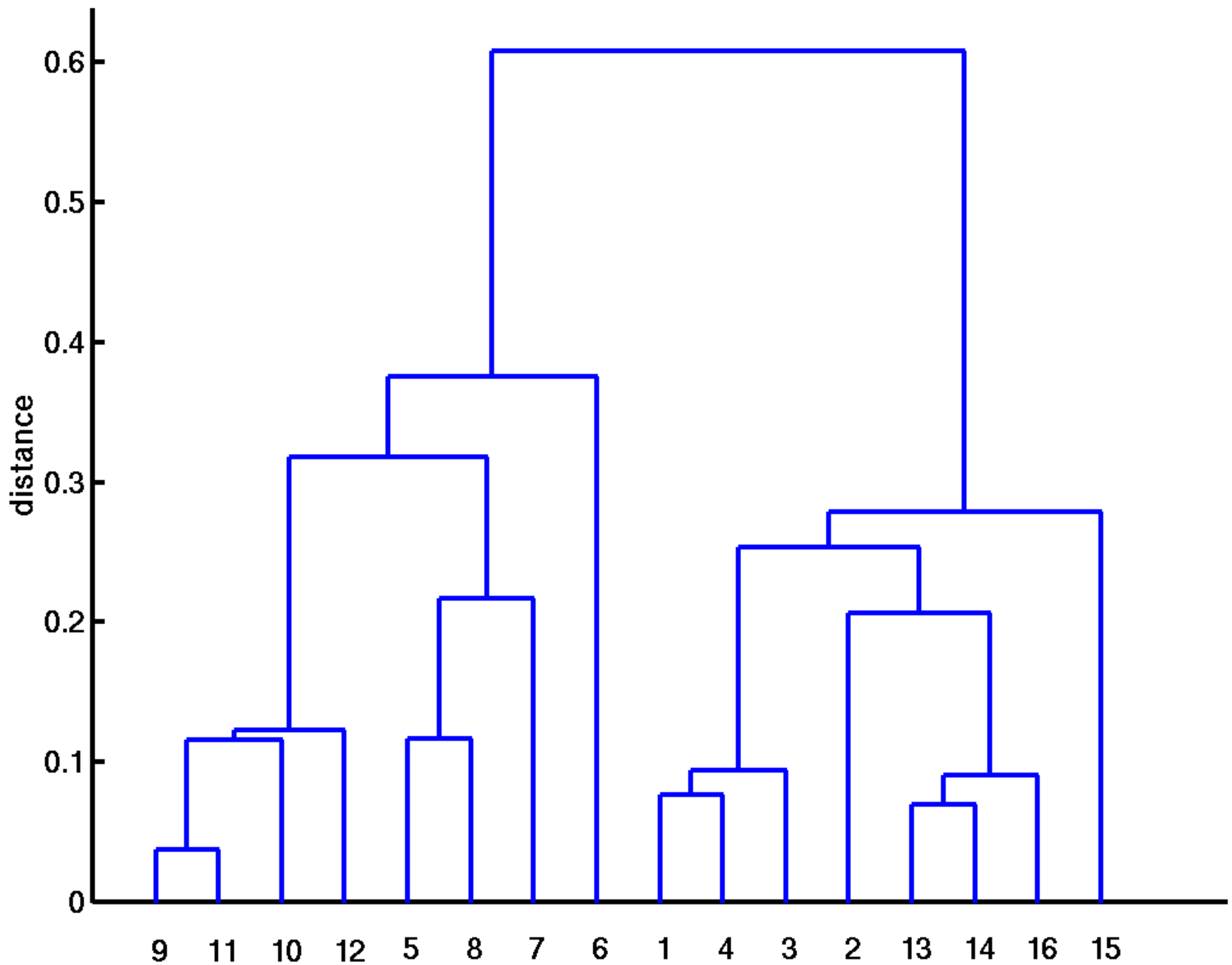


labels:

- ω_1= 1, 2, 3, 4
- ω_2= 5, 6, 7, 8
- ω_3=9,10,11,12
- ω_4=13,14,15,16



Dendrogram over leaves using Euclidean distance and single linkage

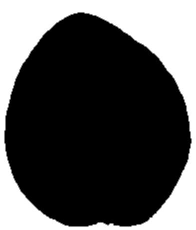


label:

- ω_1= 1, 2, 3, 4
- ω_2= 5, 6, 7, 8
- ω_3=9,10,11,12
- ω_4=13,14,15,16

ω_1

ω_2



ω_3

ω_4

Dendrogram over leaves using Euclidean distance and complete linkage