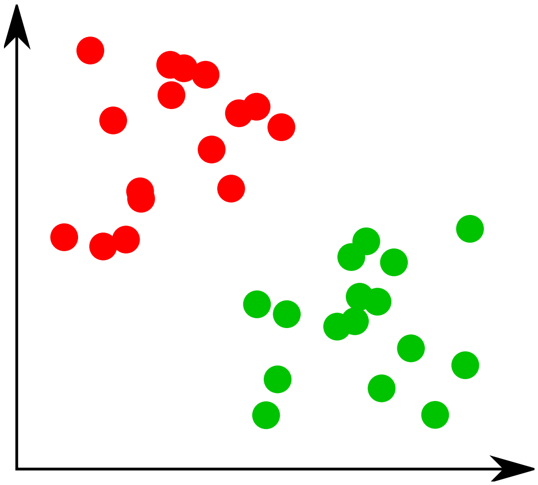
Introduction to classifiers

Henrik Skov Midtiby
hemi@mmmi.sdu.dk

2014-11-05

The classification problem



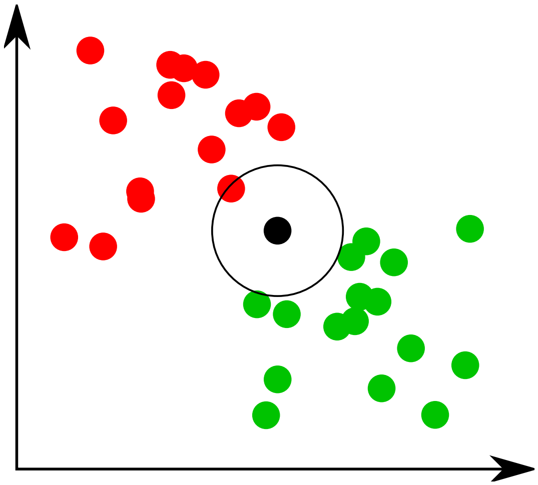
Supervised classification

Instance: Feature vector $\vec{x} \in \mathbb{R}^n$ and class relation ship $y \in -1, 1$.

Training: Given a set of feature vectors and corresponding class label.

Classification: Predict which class a new feature vector belongs to.

Nearest neighbor



Nearest neighbor

Approach

- Find nearest known sample
- Return class of that sample

Properties

- instance based learning
- distance metric

Nearest neighbor

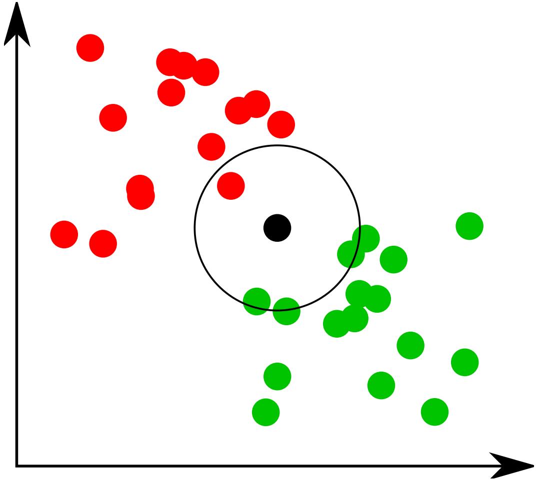
Advantages

- easy training

Disadvantages

- slow classification
- large memory requirements

k Nearest neighbor

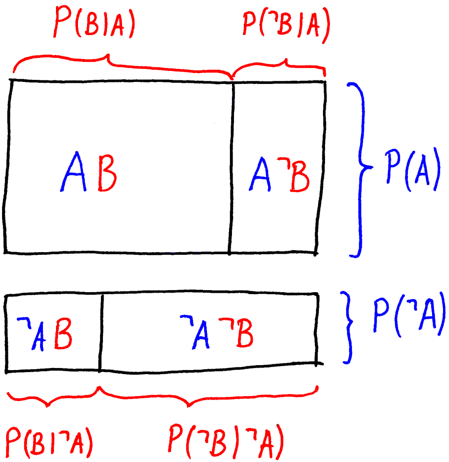


k Nearest Neighbor

- Generalization of Nearest Neighbor
- Find the k nearest neighbours
- Determine class by majority vote

http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

Bayes rule



Bayes rule

The probability of A and B can be expressed in two ways

$$P(A, B) = P(A) \cdot P(B|A) = P(B) \cdot P(A|B)$$

which is rearranged to *Bayes rule*

$$P(B|A) = \frac{P(B) \cdot P(A|B)}{P(A)}$$

often it is used as

$$P(B|A) = \frac{P(B, A)}{P(B, A) + P(\neg B, A)}$$

Naive Bayes

Approximates conditional probability densities
Uses Bayes rule to infer class membership probabilities based on observations and the conditional probability densities.

$$p(C = 1 | F_1, F_2, F_3) = \frac{p(C = 1, F_1, F_2, F_3)}{p(C = 1, F_1, F_2, F_3) + p(C = 2, F_1, F_2, F_3)}$$

Naive Bayes

Advantages

- fast classification
- low memory requirements
- probability of belonging to certain classes
- requires relative few training samples

Disadvantages

- assuming independent features

http://en.wikipedia.org/wiki/Naive_Bayes_classifier

Bayes theorem

Joint probability

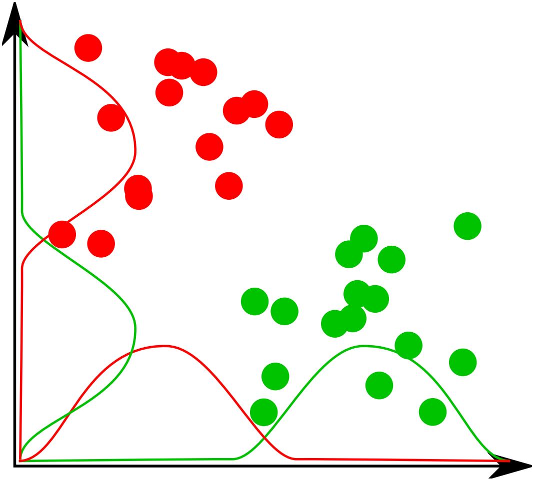
$$\begin{aligned} p(C, F_1, F_2, F_3) &= p(F_1, F_2, F_3|C)\ p(C) \\ &= p(F_1, F_2|F_3, C)\ P(F_3|C)\ p(C) \\ &= p(F_1|F_2, F_3, C)\ P(F_2, |F3, C)\ P(F_3|C)\ p(C) \end{aligned}$$

Naive assumption

$$\begin{aligned} p(C, F_1, F_2, F_3) &\simeq p(F_1|C)\ P(F_2, |C)\ P(F_3|C)\ p(C) \\ &\simeq \left[\prod_{k=1}^{3} p(F_k|C)\right] \cdot p(C) \end{aligned}$$

http://en.wikipedia.org/wiki/Bayes%27_theorem

Conditional probabilities



Support vector machines

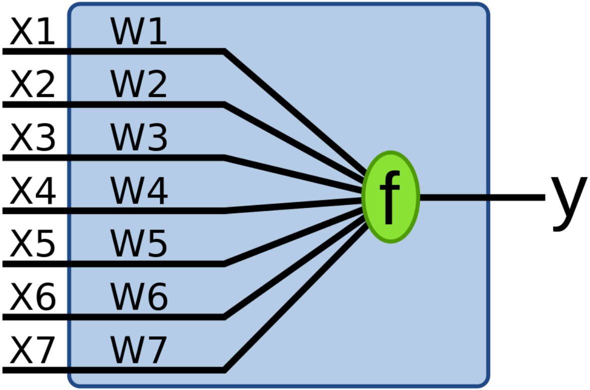
Advantages

- state of art classification results

Disadvantages

- binary classification

Perceptron



$$f(\vec{x}) = \vec{x} \cdot \vec{w}$$

Perceptron

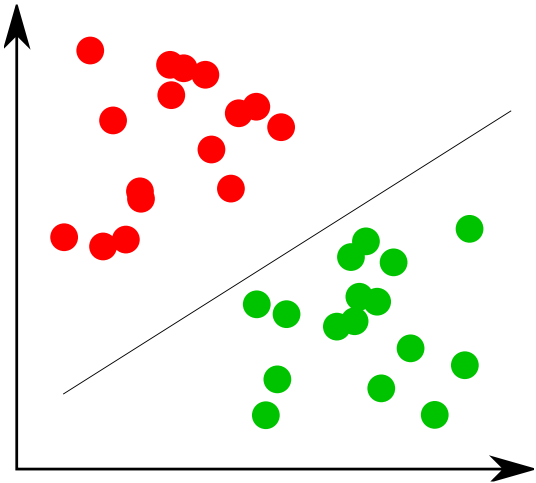
Invented in 1957 by Frank Rosenblatt
Works on linearly separable problems
Decision rule

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x} + b > 0 \\ -1 & \text{otherwise} \end{cases}$$

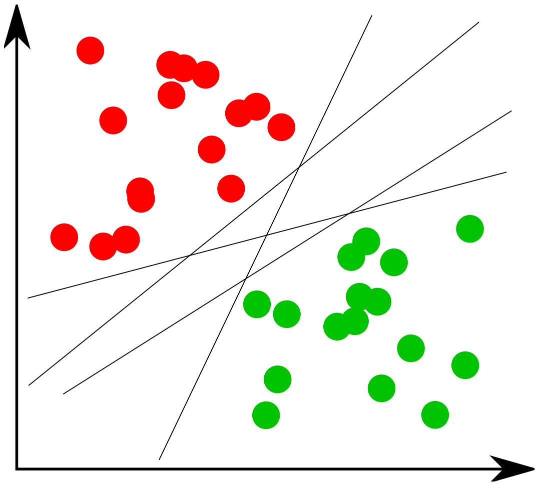
Demonstration at
`https://www.khanacademy.org/cs/perceptron-classifier/993241235`

`http://en.wikipedia.org/wiki/Perceptron`

Perceptron classifier



Perceptron classifier



Perceptron Algorithm

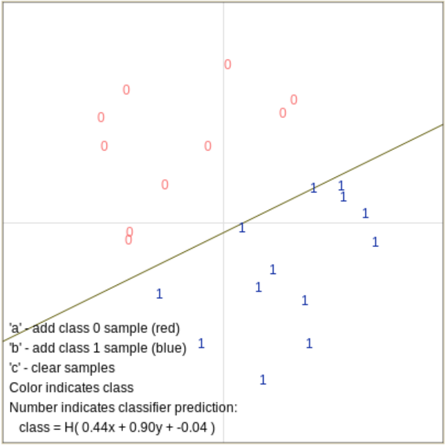
argument: $X := \{x_1, \ldots, x_m\} \subset \mathcal{X}$ (data)
$Y := \{y_1, \ldots, y_m\} \subset \{\pm 1\}$ (labels)

function $(w, b) = \mathrm{Perceptron}(X, Y, \eta)$
  initialize $w, b = 0$
  repeat
    Pick (x_i, y_i) from data
    if $y_i(w \cdot x_i + b) \leq 0$ then

$$\begin{aligned} w' &= w + y_i x_i \\ b' &= b + y_i \end{aligned}$$

  until $y_i(w \cdot x_i + b) > 0$ for all i
end

Perceptron example



Demonstration at
https://www.khanacademy.org/cs/perceptron-classifier/993241235

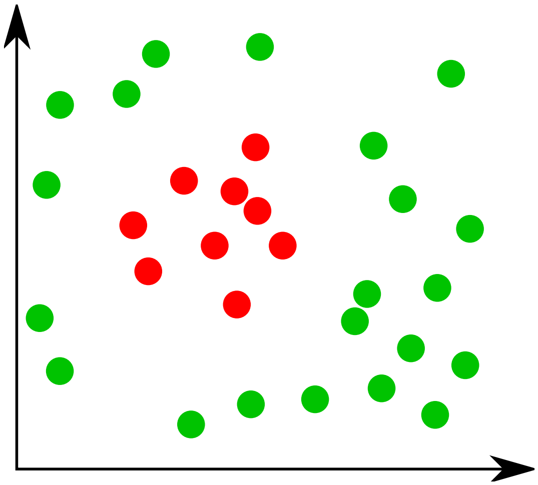
Perceptron training as a optimization problem

Minimize:
$f(\vec{w}, b) = 0$

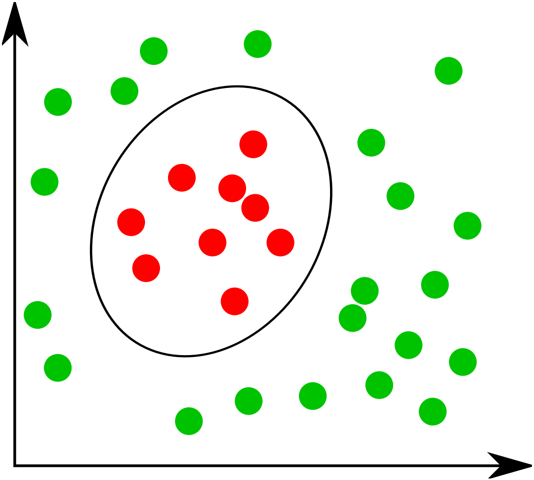
Subject to:
$1 - y_i(\vec{w}^T \vec{x}_i + b) \leq 0$

Find a feasible solution.

Linearly non separable problems



Linearly non separable problems – Decision surface



http://www.youtube.com/watch?v=3liCbRZPrZA

http://www.youtube.com/watch?v=9NrALgHFwTo

Derived features

If the classification problem is not linearly separable...

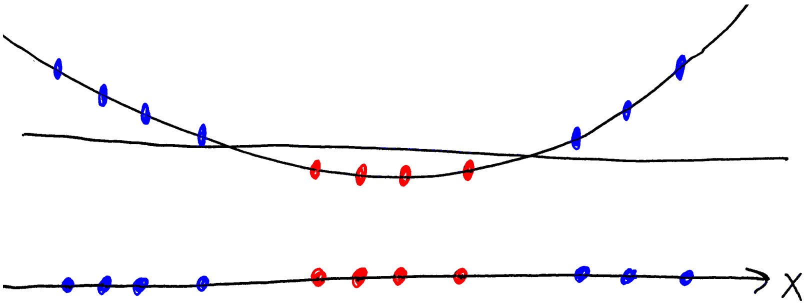
Map the input feature space to an extended feature space.

An example

$$\phi([x_1, x_2, x_3]) = [x_1^2, x_2^2, x_3^2, x_1 x_2, x_1 x_3, x_2 x_3]$$
$$\phi([z_1, z_2, z_3]) = [z_1^2, z_2^2, z_3^2, z_1 z_2, z_1 z_3, z_2 z_3]$$

Derived features

Derived features



Perceptron Algorithm

argument: $X := \{x_1, \ldots, x_m\} \subset \mathcal{X}$ (data)
$Y := \{y_1, \ldots, y_m\} \subset \{\pm 1\}$ (labels)

function $(w, b) = \mathrm{Perceptron}(X, Y, \eta)$
  initialize $w, b = 0$
  repeat
    Pick (x_i, y_i) from data
    if $y_i(w \cdot x_i + b) \leq 0$ then

$$\begin{aligned} w' &= w + y_i x_i \\ b' &= b + y_i \end{aligned}$$

  until $y_i(w \cdot x_i + b) > 0$ for all i
end

Perceptron on Features

argument: $X := \{x_1, \ldots, x_m\} \subset \mathcal{X}$ (data)
$Y := \{y_1, \ldots, y_m\} \subset \{\pm 1\}$ (labels)
function $(w, b) = \mathrm{Perceptron}(X, Y, \eta)$
  initialize $w, b = 0$
  repeat
    Pick (x_i, y_i) from data
    if $y_i(w \cdot \Phi(x_i) + b) \leq 0$ then

$$\begin{aligned} w' &= w + y_i \Phi(x_i) \\ b' &= b + y_i \end{aligned}$$

  until $y_i(w \cdot \Phi(x_i) + b) > 0$ for all i
end

Important detail

$w = \sum_j y_j \Phi(x_j)$ and hence $f(x) = \sum_j y_j (\Phi(x_j) \cdot \Phi(x)) + b$

Kernel Perceptron

argument: $X := \{x_1, \ldots, x_m\} \subset \mathcal{X}$ (data)
$Y := \{y_1, \ldots, y_m\} \subset \{\pm 1\}$ (labels)

function $f = \mathrm{Perceptron}(X, Y, \eta)$
  initialize $f = 0$
  repeat
    Pick (x_i, y_i) from data
    if $y_i f(x_i) \leq 0$ then
      $f(\cdot) \leftarrow f(\cdot) + y_i k(x_i, \cdot) + y_i$
  until $y_i f(x_i) > 0$ for all i
end

Important detail

$w = \sum_j y_j \Phi(x_j)$ and hence $f(x) = \sum_j y_j k(x_j, x) + b$.

The kernel trick

Use the derived features from earlier

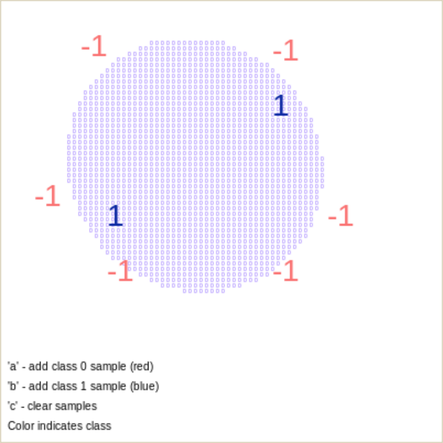
$$\phi([x_1, x_2, x_3]) = [x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3]$$
$$\phi([z_1, z_2, z_3]) = [z_1^2, z_2^2, z_3^2, \sqrt{2}z_1z_2, \sqrt{2}z_1z_3, \sqrt{2}z_2z_3]$$

Now the dot product of $\phi(x)$ and $\phi(z)$ can be computed as follows:

$$\begin{aligned}
&\phi([x_1, x_2, x_3]) \cdot \phi([z_1, z_2, z_3]) \\
&\qquad = x_1^2z_1^2 + x_2^2z_2^2 + x_3^2z_3^2 + 2x_1x_2z_1z_2 + 2x_1x_3z_1z_3 + 2x_2x_3z_2z_3 \\
&([x_1, x_2, x_3] \cdot [z_1, z_2, z_3])^2 \\
&\qquad = x_1^2z_1^2 + x_2^2z_2^2 + x_3^2z_3^2 + 2x_1x_2z_1z_2 + 2x_1x_3z_1z_3 + 2x_2x_3z_2z_3
\end{aligned}$$

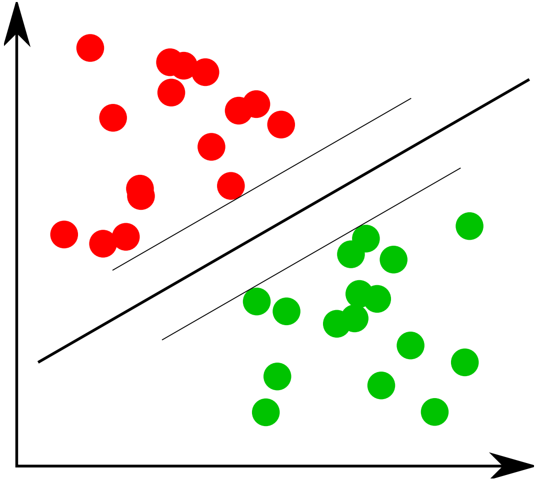
Two different methods of computing the same value, lets use the fastest one.

Kernel perceptron example



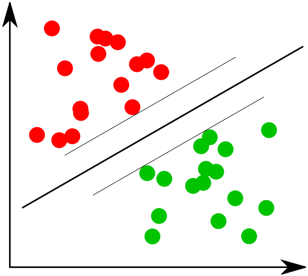
Demonstration at
`https://www.khanacademy.org/cs/perceptron-classifier-kernel-version/1116691580`

SVM



SVM

The "best possible" perceptron (largest margin).
Soft classifier (for handling errors).



http://en.wikipedia.org/wiki/Support_vector_machine

Perceptron training as a optimization problem

Minimize:

$$f(\vec{w}, b) = 0 \tag{1}$$

Subject to:

$$y_i(\vec{w}^T \vec{x}_i + b) \geq 1 \tag{2}$$

Find a feasible solution.

SVM optimization problem

Minimize:

$$f(\vec{w}, b) = \frac{1}{2}||\vec{w}||^2 \tag{3}$$

Subject to:

$$y_i(\vec{w}^T\vec{x}_i + b) \geq 1 \tag{4}$$

Find maximum margin separating hyperplane.

SVM optimization problem - non separable case

Minimize:

$$f(\vec{w}, b) = \frac{1}{2}||\vec{w}||^2 + C\sum_i \varepsilon_i \tag{5}$$

Subject to:

$$y_i(\vec{w}^T \vec{x}_i + b) - \varepsilon_i \geq 1 \tag{6}$$

Find maximum margin separating hyperplane allowing a few misclassifications.

SVM optimization problem - kernelized

Minimize:

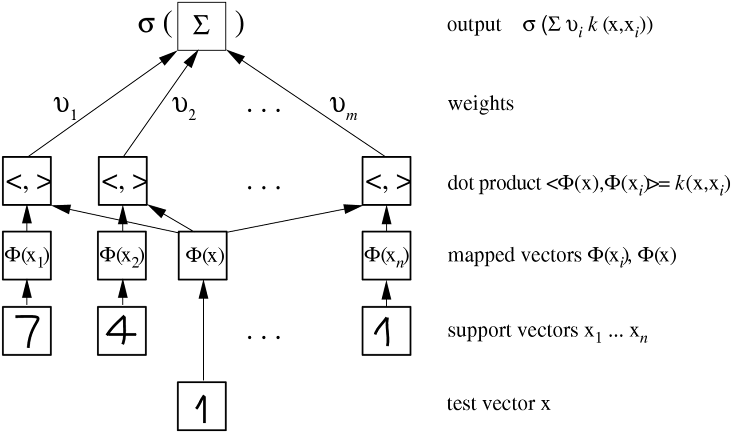
$$f(\vec{w}, b) = \frac{1}{2}||\vec{w}||^2 + C\sum_i \varepsilon_i \tag{7}$$

Subject to:

$$y_i \left(\sum_j w_j k(\vec{x}_j, \vec{x}_i) + b \right) - \varepsilon_i \geq 1 \tag{8}$$

Find maximum margin separating hyperplane allowing a few misclassifications.

SV Classification Machine



Boosting

Combination of several simple classifiers, which are allowed to make certain error types

Advantages

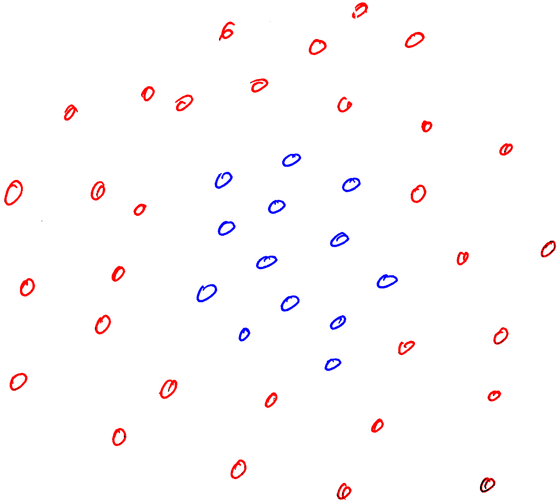
- state of art classification results

Disadvantages

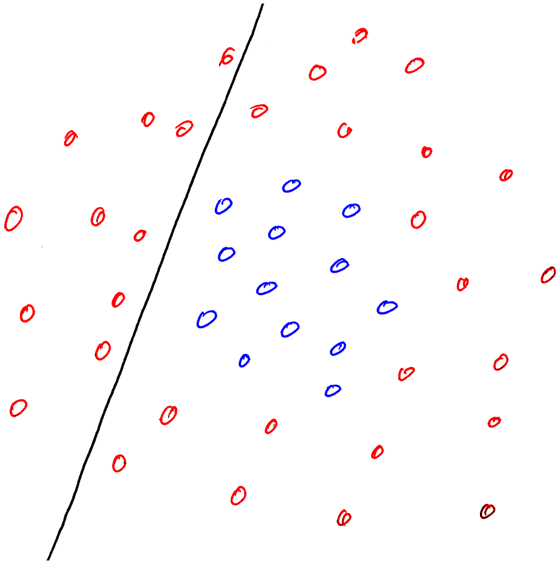
- binary classification

Used for face recognition in
Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. International Journal of Computer Vision, 57(2), 137154. doi:10.1023/B:VISI.0000013087.49260.fb

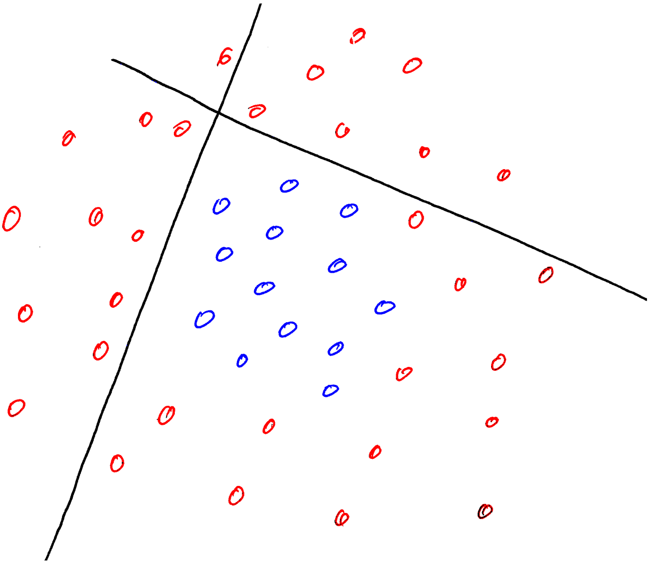
Boosting



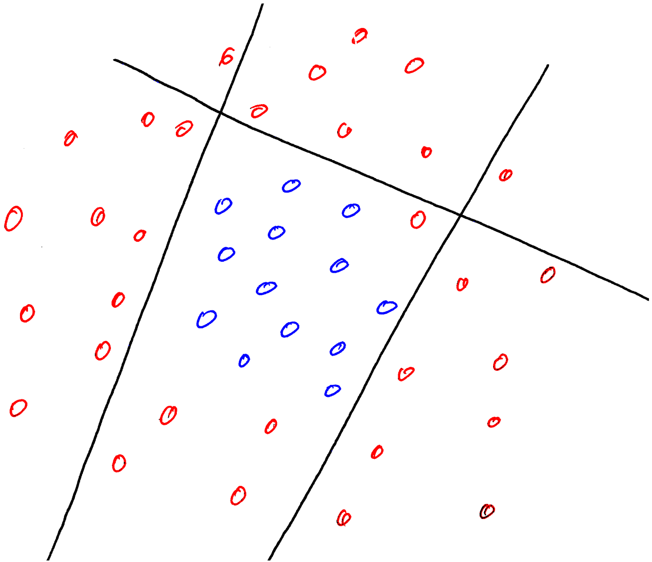
Boosting



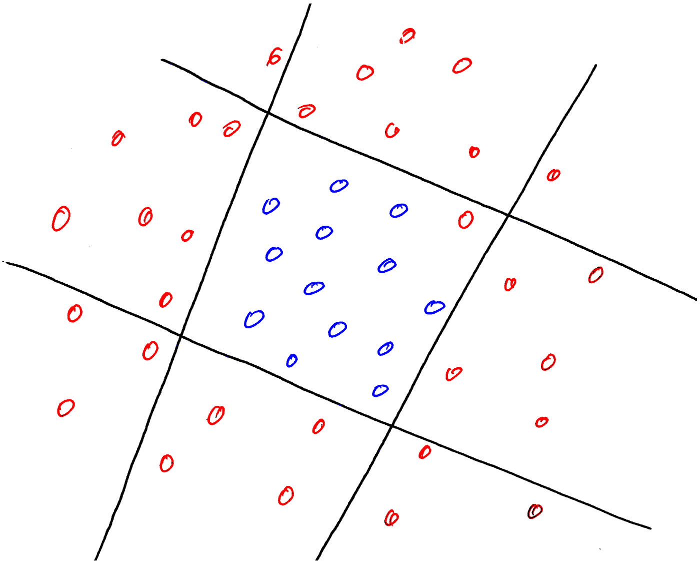
Boosting



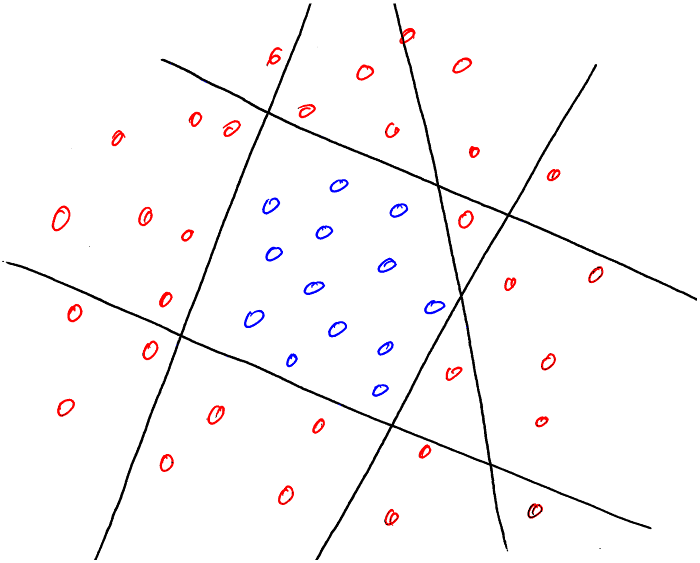
Boosting



Boosting



Boosting

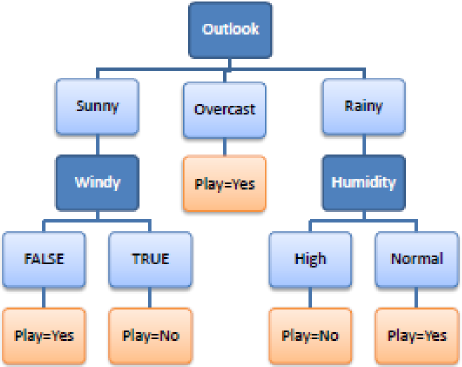


Decision trees

Go golfing based on the parameters

- weather forecast
- wind
- humidity

Decision trees



http://www.saedsayad.com/decision_tree.htm

Random forests

Combination of a large quantity of decision trees

Advantages

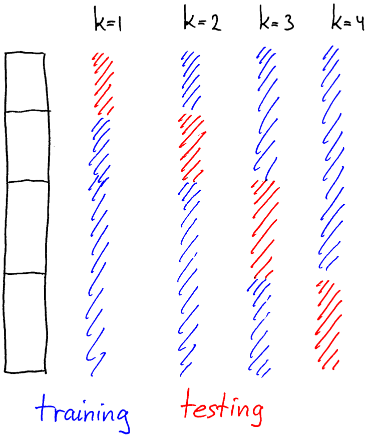
- state of art classification results

Disadvantages

- difficult to explain ...
- ?

Cross validation

Method for using the entire dataset for training and testing.



Ressources

An Introduction to Machine Learning with Kernels, Alexander J. Smola, 2004