Lecture 10: Support Vector Machines and their Applications

Cognitive Systems - Machine Learning

Part II: Special Aspects of Concept Learning

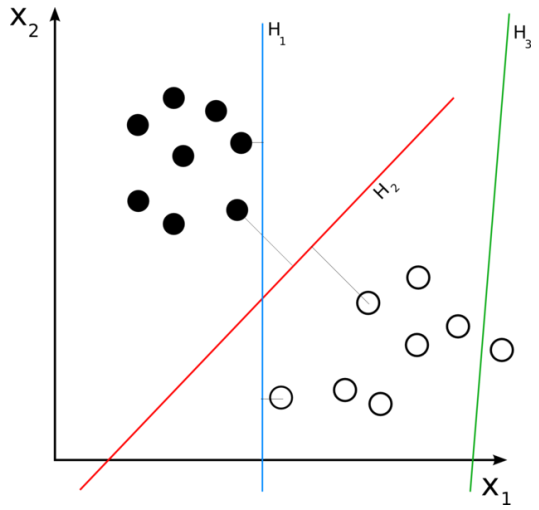
SVM, kernel trick, linear separability, text mining, active learning, "mind reading"

last change: 20. Januar 2011

Motivation

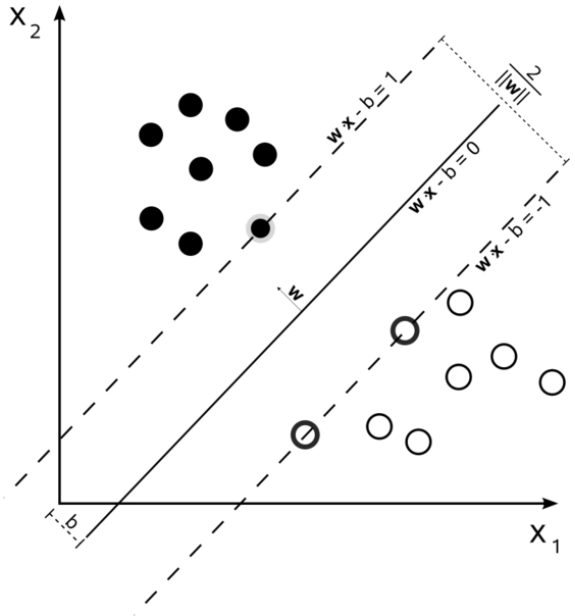
- linear classifiers (e.g. perceptrons): efficiently trainable, but low capacity and only for non-symbolic instances
- non-linear classifiers (e.g. NNs with hidden layers): high capacity, but high time complexity, local optima, overfitting, only numerical data
- idea of kernel methods: non-linearly embed instance space in high dimensional feature space with dot-product where mapped training data is linearly seperable
  - high capacity
  - applicable for symbolical data
  - time efficient training (polynomial with sample size)
  - global optimum
  - prevents overfitting

Initial Idea (Vladimir Vapnik)



- instances are linear separable
- an hyperplane is defined by $w \cdot x - b = 0$, $w \in \mathbb{R}^N$, $b \in \mathbb{R}$
- for positive instances we define $w \cdot x_i - b > 0$
- for negative instances we define $w \cdot x_i - b < 0$
- with $c(x) \in \{-1, +1\}$ holds: $c(x_i) \cdot (w \cdot x_i - b) > 0$

Initial Idea (cont'd)



- w and b not unique!
- w and b can be scaled, so that $|(w \cdot x_i) + b| = 1$ for the x_i closest to the hyperplane
  $\rightsquigarrow c(x_i) \cdot ((w \cdot x_i) + b) \geq 1$
- the distance between the two hyperplanes is given as $\frac{2}{\|w\|}$
- machine learning searches for the *best hyperplane*, i.e. the hyperplane separating all instances while being as far apart from the instances as possible

Initial Idea (cont'd)

Learning

Find a w and a b such that

- $\frac{1}{2}\|w\|^2$ is minimal
- $c(x_i)(w \cdot x_i - b) \geq 1$ for all x_i

Support Vectors

The solution can be expressed as

$$w = \sum_{i=0}^{n} \alpha_i c(x_i) x_i$$

Only few α_i are not zero. The corresponding x_i are called the support vectors. The support vectors satisfy $c(x_i)(w \cdot x_i - b) = 1$, thus lie on the two parallel hyperplanes.

Soft Margins (Corinna Cortes and Vladimir Vapnik)

- data may contain noise
- constraint is relaxed:
  $c(x_i)(w \cdot x_i - b) \geq 1 - \xi_i$
- relaxation is penalized
- ξ_i are called slack variables

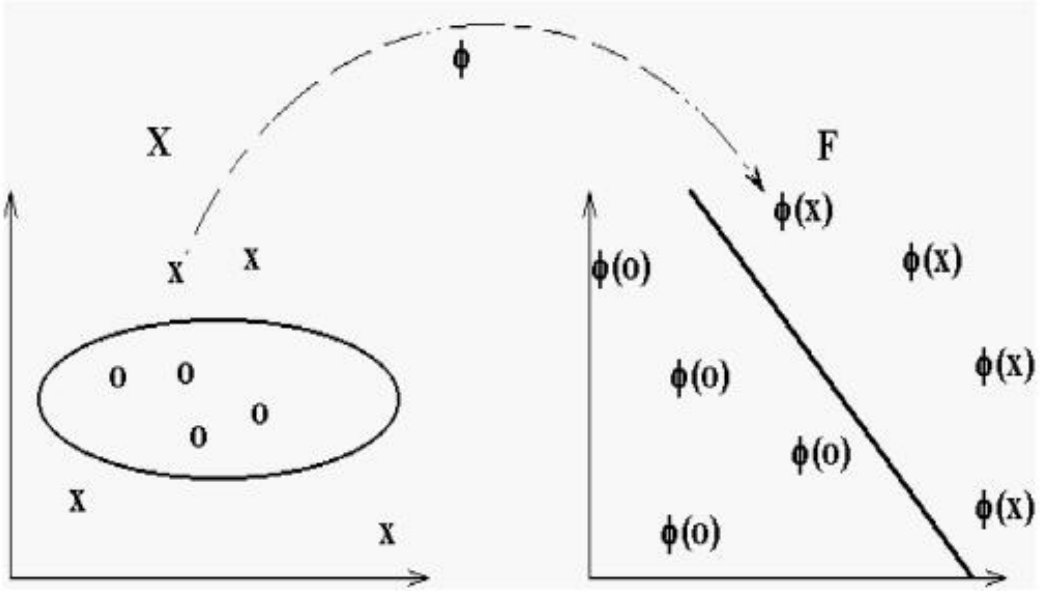
Learning

Find a w, a b and ξ_i's such that

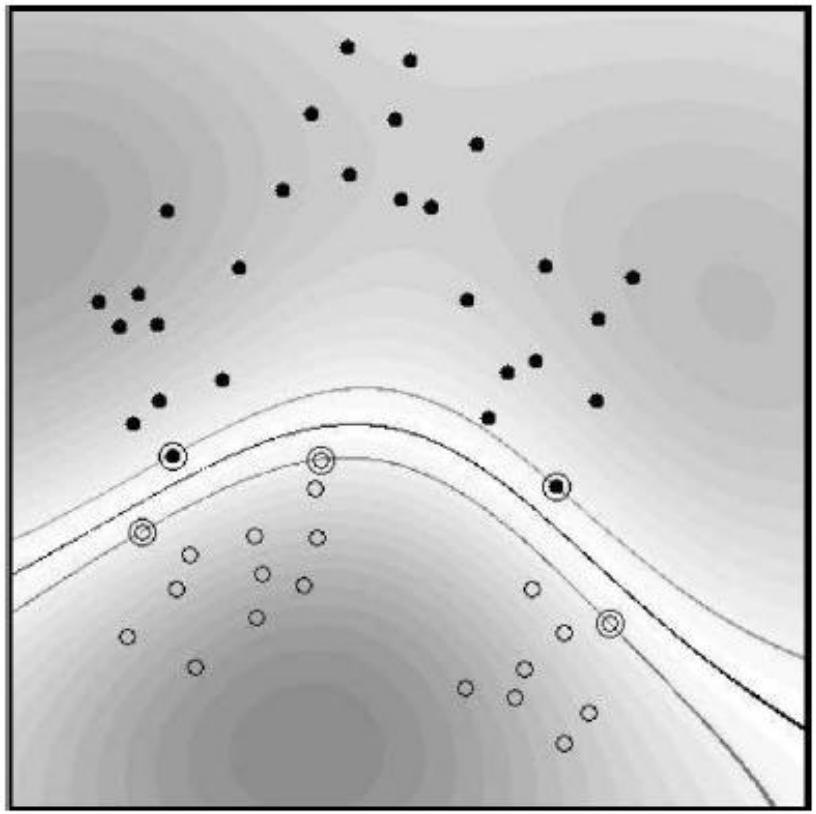
- $\frac{1}{2} \|w\|^2 - C \sum_{i=0}^{n} \xi_i$ is minimal
- $c(x_i)(w \cdot x_i - b) \geq 1 - \xi_i$ for all x_i

The Kernel Trick (Bernhard Boser, Isabelle Guyon and Vladimir Vapnik)

- the data may not be linearly separable at all
- the solution is to transform the feature space: $x_i \rightarrow \Phi(X_i)$
- for example $\Phi\left(\begin{pmatrix} a \\ b \end{pmatrix}\right) = \begin{pmatrix} a^2 \\ 2ab \\ b^2 \end{pmatrix}$



The Kernel Trick (cont'd)



- as every vector occurs only inside a dot product it is not necessary to give Φ explicitly
- every dot-product is replaced by an application of a nonlinear kernel function $\mathcal{K}(x_i, x_j)$

The Kernel Trick (cont'd)

Learning

Find a w, a b and optionally ξ_i's such that

- $\frac{1}{2}\|w\|^2 - C\sum_{i=0}^{n}\xi_i$ is minimal
- $c(x_i)(w \cdot \Phi(x) - b) \geq 1 - \xi_i$ for all i
- or, equivalently $c(x_i)(\sum_j \alpha_j c(x_j)\mathcal{K}(x_j, x_i) - b) \geq 1 - \xi_i$ for all i

Common Kernels

- Polynomial: $\mathcal{K}(x_i, x_j) = (x_i \cdot x_j)^d$
- Gaussian Radial Basis Function (RBF): $\mathcal{K}(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$
- Hyperbolic Tangent: $\mathcal{K}(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c)$

Application

- Parameter Optimization
- Class Learning
- Text Mining
- Active Learning
- “Mind Reading”

Parameter Optimization

- free parameters are:
  - the cost factor C: the higher the more accurate instances are classified (during training)
  - the kernel function $\mathcal{K}$
  - the parameters of the kernel
- normally powers of 2 are used for C: $C \in \{2^{-5}, \ldots, 2^{15}\}$
- the same holds for γ when using a Gaussian RBF: $\gamma \in \{2^{-15}, \ldots, 2^{3}\}$

Class Learning

- support vector machines are designed for concept learning
- there is ongoing research how to handle class learning
- a single SVM approach (called multi-class SVM) tries to solve the optimization problem directly
- other approaches use more than one SVM in a "divide and conquer" manner
  - 1-against-1,
  - 1-against-all,
  - error correcting output codes (ECOC)

Divide and Conquer

1-against-1

- for each pair of classes one SVM is learned
- only examples of the two classes are used
- final classification is assigned by vote

1-against-all

- for each class one SVM is trained
- the concept is whether the example belongs to the class or not
- the class of the SVM with the highest output $(w \cdot x - b)$ is assigned as final classification

A similar meta learner is available (e.g. for perceptrons).

Text Classification

- applications
  - automated tagging
  - author attribution
  - spam filtering
- examples: single documents
- representation/attributes: occurance frequency of single words
- preprocessing:
  - stop word filtering (e.g. *the*, *on*, *for*, *and*)
  - stemming
  - fix typos
  - find synonyms

Text Classification Example

Documents

1. A man is sitting on a bank in the park.
2. We owe the bank $ 1,000.
3. Two players are siting on the bank.
4. We called the bank.

Trainings data

no.	man	sit	bank	park	we	owe	player	call	category
1	1	1	1	1	0	0	0	0	bench
2	0	0	1	0	1	1	0	0	institute
3	0	1	1	0	0	0	1	0	bench
4	0	0	1	0	1	0	0	1	institute

Active Learning

- active learning is a technique where the learner chooses which examples it needs
- it is used when examples are easily available but labeling the examples is cost-intensive (e.g. biological research, expert ratings)
- in each learning step i there is a set of examples with known label ($D_{k,i}$) and a set of examples with unknown label ($D_{u,i}$)
- the algorithm chooses from the unlabeled examples which shall be labeled ($D_{c,i}$)

Using SVMs for Active Learning

- SVMs are used to assign $D_{c,i}$
- the distance of each unlabeled example from the hyperplane is calculated
- used are
  - the closest examples (possibly overfitting),
  - the examples most far apart (possibly low accuracy) or
  - a mixture of both

"Mind Reading"

Study by Mitchell et al. 2004

- functional Magnetic Resonance Imaging (fMRI)
- normal students from the university community
- presented were a sentence and a simple image
- the aim was to classify an image sequence as *sentence* or *picture*
- 40 trials per subject
- 13 subjects

Full Reference

Tom M. Mitchell, Rebecca Hutchinson, Radu S. Niculescu, Francisco Pereira, Xuerui Wang, Marcel Just, Sharlene Newman (2004). Learning to Decode Cognitive States from Brain Images. Machine Learning, 57:145–175

fMRI

- three-dimensional images related to neural activity in the brain through time
- ratio of oxygenated hemoglobin to deoxygenated hemoglobin in the blood (BOLD)
- high spatial resolution (several millimeters)
- about 10,000 voxels (volume elements) per image
- one image per 0.5 seconds

Data Preprocessing

- artifacts due to head motion, signal drift, and other sources were removed
- voxel activity values were represented by the percent difference from their mean value during rest
- 80 examples per subject (1040 total)
- about 10,000 attributes per image (about 160,000 per example)

Results

	error rate w/o feature selection	with feature selection
trained for each subject	0.34	0.11
trained for all subjects	—	0.25

Drawbacks

- each voxel contains on the order of hundreds of thousands of neurons
- the fMRI BOLD response associated with an impulse of neural activity endures for many seconds (9–13)

Learning Terminology

Support Vector Machine

Supervised Learning	unsupervised learning

Approaches:

Concept / Classification	Policy Learning
symbolic	**statistical** / neuronal network
inductive	analytical

Learning Strategy:

$\Rightarrow$ **learning from examples**