Topic 8 – Support Vector Machines and Kernel-Based Learning

Xiaolin Hu

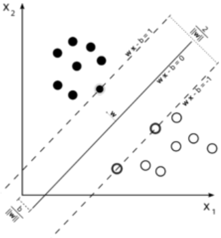
Department of Computer Science and Technology
Tsinghua University

April 7, 2017

Outline

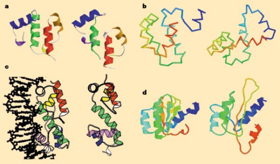
Classification methods

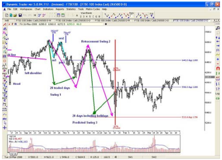
- Decision tree:
  - attributes of instances are nominal data
  - objective function are discrete
- K-nearest neighbor:
  - instances are points in the Euclidean space
  - objective function can be discrete or continuous
- Support vector machine:
  - instances are points in the Euclidean space
  - objective function can be discrete or continuous



- The present form of support vector machine (SVM) was largely developed at AT&T Bell Laboratories by Vapnik and co-workers.
- Known as a maximum margin classifier.
- Originally proposed for classification and soon applied to regression and time series prediction.
- One of the most efficient supervised learning methods.

Applications





Outline

Outline

Problem

- Given a set of training samples

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N), x_i \in R^n, y_i \in \{-1, 1\},$$

find a function $f(x, \alpha)$ to classify the samples, such that

$$f(x_i, \alpha) \begin{cases} > 0, & \forall y_i = +1; \\ < 0, & \forall y_i = -1, \end{cases}$$

where α denotes the parameters.

Problem

- Given a set of training samples

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N), x_i \in R^n, y_i \in \{-1, 1\},$$

find a function $f(x, \alpha)$ to classify the samples, such that

$$f(x_i, \alpha) \begin{cases} > 0, & \forall y_i = +1; \\ < 0, & \forall y_i = -1, \end{cases}$$

where α denotes the parameters.

- For a testing sample x, we can predict its label by $\text{sign}[f(x, \alpha)]$.

Problem

- Given a set of training samples

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N), x_i \in R^n, y_i \in \{-1, 1\},$$

find a function $f(x, \alpha)$ to classify the samples, such that

$$f(x_i, \alpha) \begin{cases} > 0, & \forall y_i = +1; \\ < 0, & \forall y_i = -1, \end{cases}$$

where α denotes the parameters.

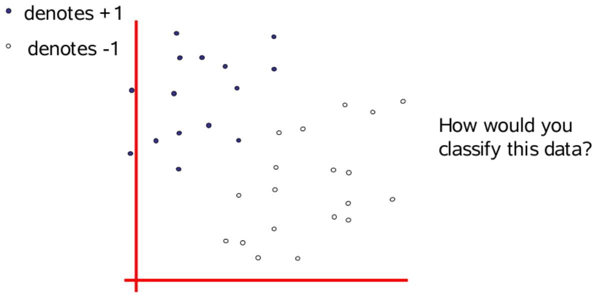
- For a testing sample x, we can predict its label by $\text{sign}[f(x, \alpha)]$.
- $f(x, \alpha) = 0$ is called the separation hyperplane.

Linear classifiers

Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

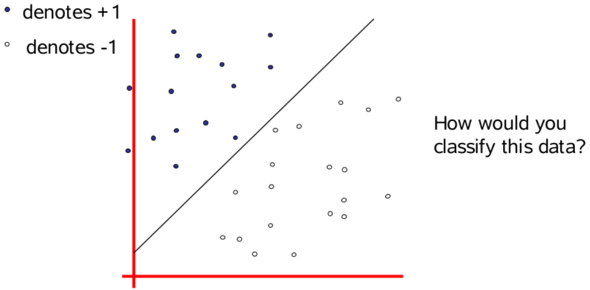


Linear classifiers

Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

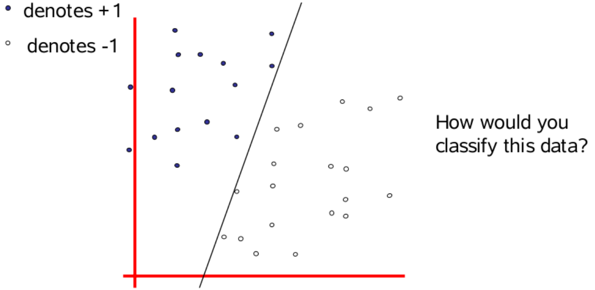


Linear classifiers

Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

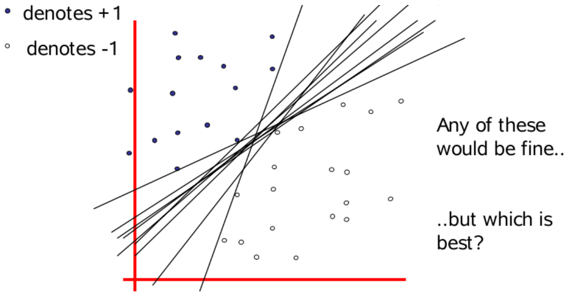


Linear classifiers

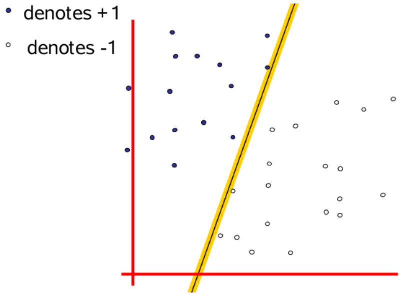
Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

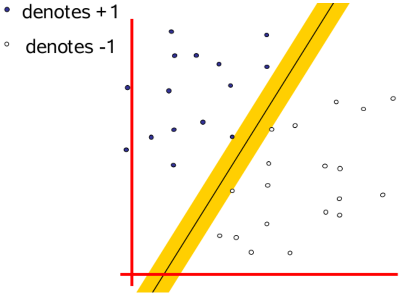


Margin of a linear classifier



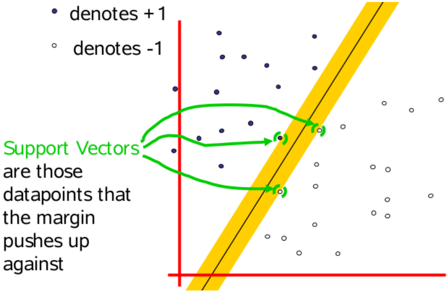
Definition: the width that the boundary could be increased by before hitting a data point.

Maximum margin linear classifier



Definition: the linear classifier with the maximum margin.

Support vectors



Problem formulation

To formulate the margin, we further requires that for all samples

$$f(x_i, \alpha) = \langle x_i, w \rangle + b \begin{cases} \geq +1, & \forall y_i = +1; \\ \leq -1, & \forall y_i = -1. \end{cases}$$

or

$$y_i(\langle x_i, w \rangle + b) \geq 1, \quad i = 1, \ldots, N.$$

Problem formulation

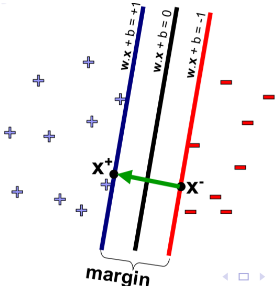
To formulate the margin, we further requires that for all samples

$$f(x_i, \alpha) = \langle x_i, w \rangle + b \begin{cases} \geq +1, & \forall y_i = +1; \\ \leq -1, & \forall y_i = -1. \end{cases}$$

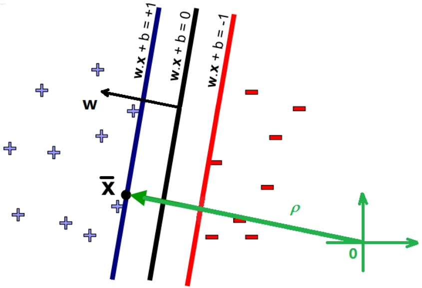
or

$$y_i(\langle x_i, w \rangle + b) \geq 1, \quad i = 1, \ldots, N.$$

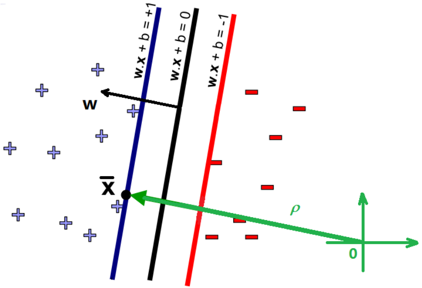
- We have introduced two additional hyperplanes $\langle x, w \rangle + b = \pm 1$ parallel to the separation hyperplane $\langle x, w \rangle + b = 0$



What is the margin? The distance between the two new hyperplanes.

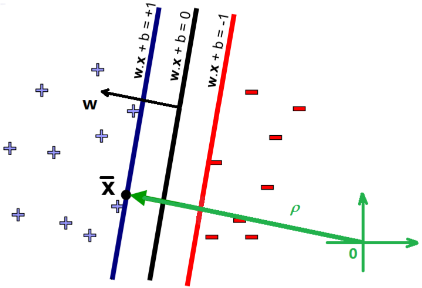


What is the margin? The distance between the two new hyperplanes.



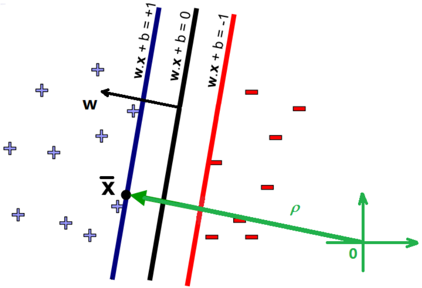
- The minimum distance between the hyperplane $\langle x, w \rangle + b = 1$ and the origin is $\rho_1 = \frac{1-b}{\|w\|}$. (why?)

What is the margin? The distance between the two new hyperplanes.



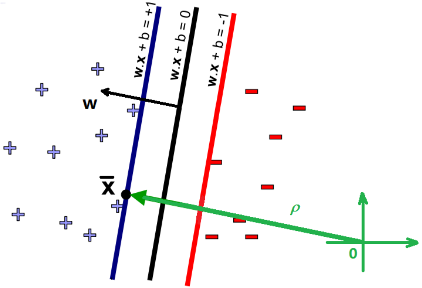
- The minimum distance between the hyperplane $\langle x, w\rangle + b = 1$ and the origin is $\rho_1 = \frac{1-b}{\|w\|}$. (why?)
- The minimum distance between the hyperplane $\langle x, w\rangle + b = -1$ and the origin is $\rho_2 = \frac{-1-b}{\|w\|}$.

What is the margin? The distance between the two new hyperplanes.



- The minimum distance between the hyperplane $\langle x, w \rangle + b = 1$ and the origin is $\rho_1 = \frac{1-b}{\|w\|}$. (why?)
- The minimum distance between the hyperplane $\langle x, w \rangle + b = -1$ and the origin is $\rho_2 = \frac{-1-b}{\|w\|}$.
- The margin is $|\rho_1 - \rho_2| = 2/\|w\|$.

How to calculate ρ_1 and ρ_2?



Note $\bar{x} = \rho_1 w/\|w\|$, where $w/\|w\|$ is the unit vector along the direction w. Since $\bar{x}$ is on the blue hyperplane, then

$$\langle \rho_1 w/\|w\|, w\rangle + b = 1$$

which follows $\rho_1 = \frac{1-b}{\|w\|}$. Similarly, we obtain $\rho_2 = \frac{-1-b}{\|w\|}$.

The optimization problem

$$\begin{aligned} \max_{w,b} \quad & \frac{2}{\|w\|} \\ \text{s.t.} \quad & y_i \left(\langle w, x_i \rangle + b \right) \geq 1, \quad i = 1, \ldots, N \end{aligned}$$

or equivalently

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i \left(\langle w, x_i \rangle + b \right) \geq 1, \quad i = 1, \ldots, N \end{aligned}$$

- Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left(y_i \left(\langle w, x_i \rangle + b\right) - 1\right)$$

- Lagrange function

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left(y_i \left(\langle w, x_i \rangle + b \right) - 1 \right)$$

- KKT conditions

$$\frac{\partial L}{\partial w} = w - \sum_i \alpha_i y_i x_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_i y_i \alpha_i = 0$$

$$\alpha_i \left(y_i \left(\langle w, x_i \rangle + b \right) - 1 \right) = 0$$

$$y_i \left(\langle w, x_i \rangle + b \right) \geq 1$$

$$\alpha_i \geq 0$$

- The dual function is

$$\phi(\alpha) = \inf_{w,b} L(w, b, \alpha).$$

Because w and b are unconstrained, the RHS can be obtained by letting $\partial L/\partial w = 0$ and $\partial L/\partial b = 0$. Substitute the results into $L(w, b, \alpha)$ and get (how?)

$$\phi(\alpha) = -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i.$$

- The dual function is

$$\phi(\alpha) = \inf_{w,b} L(w, b, \alpha).$$

Because w and b are unconstrained, the RHS can be obtained by letting $\partial L / \partial w = 0$ and $\partial L / \partial b = 0$. Substitute the results into $L(w, b, \alpha)$ and get (how?)

$$\phi(\alpha) = -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i.$$

- The dual problem

$$\begin{array}{ll} \min & \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_i \alpha_i \\ \text{s.t.} & \sum_i y_i \alpha_i = 0 \\ & \alpha_i \geq 0 \end{array}$$

Solution to the primal problem

- Normal vector

$$w^* = \sum_i y_i \alpha_i^* x_i = \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* x_i$$

- Bias

$$\begin{aligned} &\alpha_i^* (y_i(\langle w^*, x_i \rangle + b^*) - 1) = 0 \\ &\Rightarrow b^* = y_i - \langle w^*, x_i \rangle \quad \forall \alpha_i^* > 0. \end{aligned}$$

- Hyperplane

$$\begin{aligned} f(x) &= \langle w^*, x \rangle + b^* \\ &= \langle \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* x_i, x \rangle + b^* = \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* \langle x_i, x \rangle + b^* \end{aligned}$$

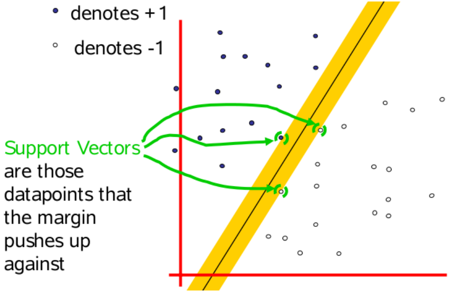
Support vectors

Most α_i's are zero (sparse solution).

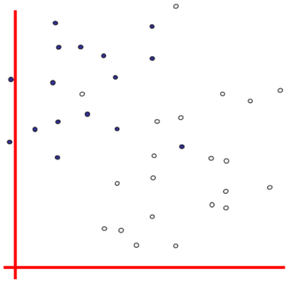
- Note that

$$\alpha_i^*(y_i(\langle w^*, x_i\rangle + b^*) - 1) = 0.$$

α_i is nonzero only if $y_i(\langle w^*, x_i\rangle + b^*) = 1$, i.e., x_i lies on the boundaries of the margin. These x_i's are support vectors.



Non-separable case



Idea: minimize $\langle w, w \rangle$, while minimizing training errors.

- 0/1 loss

$$\min_{w,b} \langle w, w \rangle + C \times (\text{number of training errors})$$

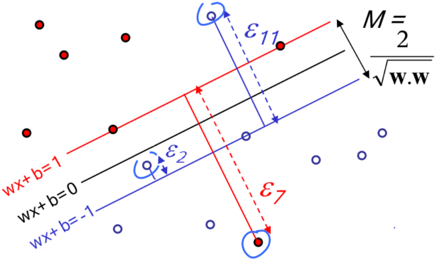
  Disadvantage: discrete optimization problem –hard to solve.

- linear loss

$$\min_{w,b} \langle w, w \rangle + C \times \sum \text{linear loss}$$

  Advantage: can be expressed as a QP problem.

Primal problem



Separable case

$$\min_{w,b} \frac{1}{2}\langle w, w\rangle$$

$$s.t. y_i\left(\langle w, x_i\rangle + b\right) \geq 1$$

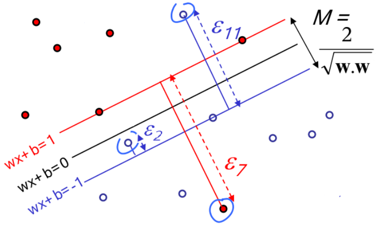
Non-separable case

$$\min_{w,b} \frac{1}{2}\langle w, w\rangle + C\sum_i \varepsilon_i$$

$$s.t. y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

Soft margin



$$\min_{w,b} \frac{1}{2}\langle w, w\rangle + C\sum_i \varepsilon_i$$

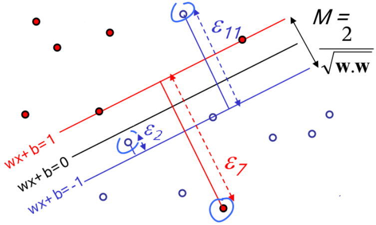
$$s.t. y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

Still want to find the maximum margin hyperplane but this time:

- We will allow some training examples to be misclassified
- We will allow some training examples to fall within the margin region

Soft margin



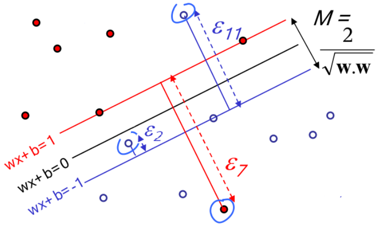
$$\min_{w,b} \frac{1}{2}\langle w, w\rangle + C\sum_i \varepsilon_i$$

$$s.t. y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

- For $\varepsilon_i = 0$, the data point falls on the boundaries of the region of separation or outside the region of separation and on the right side of the decision surface.
- For $0 < \varepsilon_i \leq 1$, the data point falls inside the region of separation but on the right side of the decision surface.
- For $\varepsilon_i > 1$, the data point falls on the wrong side of the separating hyperplane and introduce a wrong decision.

Soft margin



$$\min_{w,b} \frac{1}{2}\langle w, w\rangle + C\sum_i \varepsilon_i$$

$$s.t. y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

The positive constant C controls the balance between large margin and small misclassification error

- large C: prefer small error
- small C: prefer large margin

Dual problem

Separable case

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle x_i, x_j\rangle - \sum_i \alpha_i \\ s.t. & \sum_i y_i\alpha_i = 0 \\ & \alpha_i \geq 0 \end{array}$$

$\Downarrow$homework

Non-separable case

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle x_i, x_j\rangle - \sum_i \alpha_i \\ s.t. & \sum_i y_i\alpha_i = 0 \\ & C \geq \alpha_i \geq 0 \end{array}$$

Solution to the primal problem

- Normal vector

$$w^* = \sum_i y_i \alpha_i^* x_i = \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* x_i$$

- Bias

$$b^* = y_i - \langle w^*, x_i \rangle \quad \forall C > \alpha_i^* > 0.$$

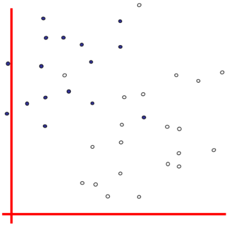
- Hyperplane

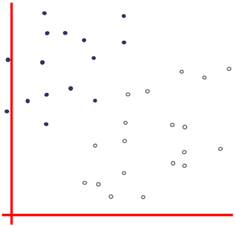
$$\begin{aligned} f(x) &= \langle w^*, x \rangle + b^* \\ &= \langle \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* x_i, x \rangle + b^* = \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* \langle x_i, x \rangle + b^* \end{aligned}$$

Outline

Input space to feature space

$$\Phi(x) : R^n \mapsto F$$





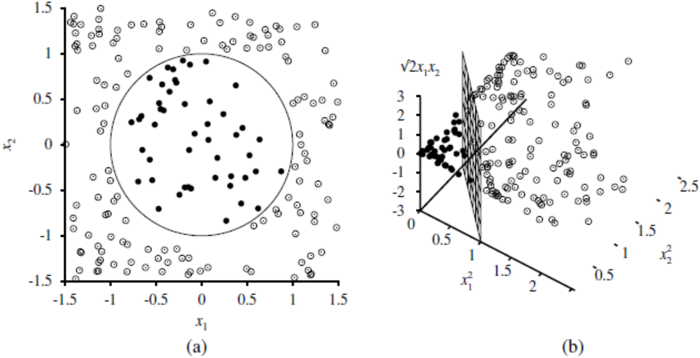


Figure 18.31 (a) A two-dimensional training set with positive examples as black circles and negative examples as white circles. The true decision boundary, $x_1^2 + x_2^2 \leq 1$, is also shown. (b) The same data after mapping into a three-dimensional input space $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$. The circular decision boundary in (a) becomes a linear decision boundary in three dimensions. Figure 18.30(b) gives a closeup of the separator in (b).

Primal problem in feature space

Input space

$$\begin{aligned} \min_{w,b} \quad & \tfrac{1}{2}\langle w, w\rangle + C\sum_i \varepsilon_i \\ s.t. \quad & y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \varepsilon_i \\ & \varepsilon_i \geq 0 \end{aligned}$$

$$\Downarrow$$

Feature space

$$\begin{aligned} \min_{w,b} \quad & \tfrac{1}{2}\langle w, w\rangle + C\sum_i \varepsilon_i \\ s.t.p \quad & y_i\left(\langle w, \Phi(x_i)\rangle + b\right) \geq 1 - \varepsilon_i \\ & \varepsilon_i \geq 0 \end{aligned}$$

Dual problem in feature space

Input space

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle x_i, x_j\rangle - \sum_i \alpha_i \\ s.t. & \sum_i y_i\alpha_i = 0 \\ & C \geq \alpha_i \geq 0 \end{array}$$

$$\Downarrow$$

Feature space

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle \Phi(x_i), \Phi(x_j)\rangle - \sum_i \alpha_i \\ s.t. & \sum_i y_i\alpha_i = 0 \\ & C \geq \alpha_i \geq 0 \end{array}$$

Solution to the primal problem in feature space

- Normal vector

$$w^* = \sum_i y_i \alpha_i^* \Phi(x_i) = \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* \Phi(x_i)$$

- Bias

$$\begin{aligned} b^* &= y_i - \langle w^*, \Phi(x_i) \rangle \\ &= y_i - \sum_{\alpha_j^* \neq 0} y_j \alpha_j^* \langle \Phi(x_j), \Phi(x_i) \rangle \quad \forall C > \alpha_i^* > 0. \end{aligned}$$

- Hyperplane

$$f(x) = \langle w^*, \Phi(x) \rangle + b^* = \langle \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* \Phi(x_i), \Phi(x) \rangle + b^*$$

$$= \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* \langle \Phi(x_i), \Phi(x) \rangle + b^*$$

Outline

Kernel trick

- What we only need to know is

$$\langle \Phi(x), \Phi(y) \rangle$$

  instead of $\Phi(x)$ and $\Phi(y)$.
- Compute dot production in input space instead of feature space

$$\langle \Phi(x), \Phi(y) \rangle = k(x, y)$$

- We do not need to represent the features explicitly.

Dual problem

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(x_i,x_j) - \sum_i \alpha_i \\ \text{s.t.} & \sum_i y_i\alpha_i = 0 \\ & C \geq \alpha_i \geq 0 \end{array}$$

Solution to the primal problem

- Hyperplane

$$\begin{aligned} f(x) &= \langle w^*, \Phi(x)\rangle + b^* \\ &= \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* k(x_i, x) + b^* \end{aligned}$$

where

$$b^* = y_i - \sum_{\alpha_j^* \neq 0} y_j \alpha_j^* k(x_j, x_i) \quad \forall C > \alpha_i^* > 0.$$

If we can know $k(x, y)$ beforehand, we don't need to compute $\Phi(x)$. Is it possible?

If we can know $k(x, y)$ beforehand, we don't need to compute $\Phi(x)$. Is it possible?

- Such kernel functions must guarantee that there exists corresponding $\Phi(x)$.

If we can know $k(x, y)$ beforehand, we don't need to compute $\Phi(x)$. Is it possible?

- Such kernel functions must guarantee that there exists corresponding $\Phi(x)$.
- Mercer's theorem

Theorem

There exists a mapping Φ and an expansion

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

if and only if, for any $g(x)$ such that if $\int g(x)^2 dx$ is finite then

$$\int k(x, y) g(x) g(y) dxdy \geq 0.$$

Commonly used kernels

- Homogeneous polynomials

$$k(x, y) = (\langle x, y \rangle)^d$$

- Inhomogeneous polynomials

$$k(x, y) = (\langle x, y \rangle + 1)^d$$

- Gaussian Kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

- Sigmoid Kernel

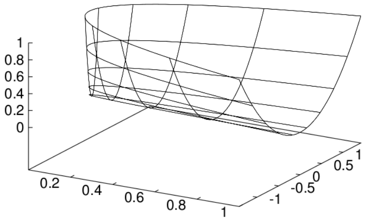
$$k(x, y) = \tanh(\eta\langle x, y \rangle + v)$$

Polynomial kernel

$$k(x, y) = (\langle x, y \rangle)^d$$

Example: $n = 2, d = 2, x = (x_1, x_2)$

- $\Phi(x) = \left(x_1^2, \sqrt{2}x_1 x_2, x_2^2\right)$

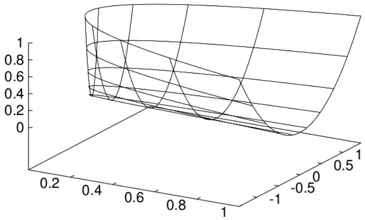


Polynomial kernel

$$k(x, y) = (\langle x, y \rangle)^d$$

Example: $n = 2, d = 2, x = (x_1, x_2)$

- $\Phi(x) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$



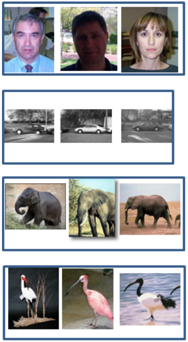
- Neither the mapping Φ nor the feature space is unique
  - $\Phi(x) = \left(x_1^2, x_1x_2, x_1x_2, x_2^2\right)$
  - $\Phi(x) = \frac{1}{\sqrt{2}}\left(x_1^2 - x_2^2, 2x_1x_2, x_1^2 + x_2^2\right)$

Outline

- Libsvm
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- Liblinear
  http://www.csie.ntu.edu.tw/~cjlin/liblinear/
- SVMlight
  http://svmlight.joachims.org/

Example: image classification with LLC and SVM

- Dataset: Caltech101
    - 9144 images
    - 102 categories
- Preprocessing
    - Convert to gray scale
    - Rescaled such that the longer side was 120 pixels
- Extract features for each image with LLC
- Train and test with SVM



Test Results

- Trained with 15 images per category: 70.16%
- Trained with 30 images per category: 73.44%

Outline

General concepts

Formal definition

A function $k : \mathcal{R}^d \times \mathcal{R}^d \to \mathcal{R}$ is called a kernel on $\mathcal{R}^d$ if there is *some* function $\Phi : \mathcal{R}^d \to \mathcal{F}$ such that

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}} \quad \forall x, x' \in \mathcal{R}^d$$

Principle

- If any learning method involves $\langle x, x' \rangle$ we can substitute it with $k(x, x')$,
- we then work in the feature space $\mathcal{F}$ induced by $\Phi(\cdot)$.

The mapping Φ

- If $\Phi(x) = x$ then $k(x, x')$ is a *linear* kernel.
- We do not need to compute the function Φ explicitly.

Constructing kernels

Constructing kernels

1. Choose a feature function $\Phi(x)$ then construct kernel

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

Constructing kernels

1. Choose a feature function $\Phi(x)$ then construct kernel

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

2. Choose a valid kernel without constructing the function $\Phi(x)$ explicitly

Theorem

There exists a mapping Φ and an expansion

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

if and only if, for any $g(x)$ such that if $\int g(x)^2 dx$ is finite then

$$\int k(x, y) g(x) g(y) dx dy \geq 0.$$

Constructing kernels

3. Building new kernels from simple kernels
Given valid kernels $k_1(x, x')$ and $k_2(x, x')$, the following new kernels will also be valid:

$$
\begin{aligned}
k(x, x') &= ck_1(x, x') \\
k(x, x') &= f(x)k_1(x, x')f(x') \\
k(x, x') &= q(k_1(x, x')) \\
k(x, x') &= \exp(k_1(x, x')) \\
k(x, x') &= k_1(x, x') + k_2(x, x') \\
k(x, x') &= k_1(x, x')k_2(x, x') \\
k(x, x') &= x^T A x'
\end{aligned}
$$

where $c > 0$ is a constant, $f(\cdot)$ is any function, $q(\cdot)$ is a polynomial with nonnegative coefficients, and A is a symmetric positive semidefinite matrix.

Gaussian kernel

It is in the form

$$k(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$$

Notice that

$$\|x - x'\|^2 = \langle x, x\rangle + \langle x', x'\rangle - 2\langle x, x'\rangle$$

and

$$k(x, x') = \exp\left(-\frac{\langle x, x\rangle}{2\sigma^2}\right)\exp\left(\frac{\langle x, x'\rangle}{\sigma^2}\right)\exp\left(-\frac{\langle x', x'\rangle}{2\sigma^2}\right)$$

Gaussian kernel

It is in the form

$$k(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$$

Notice that

$$\|x - x'\|^2 = \langle x, x\rangle + \langle x', x'\rangle - 2\langle x, x'\rangle$$

and

$$k(x, x') = \exp\left(-\frac{\langle x, x\rangle}{2\sigma^2}\right)\exp\left(\frac{\langle x, x'\rangle}{\sigma^2}\right)\exp\left(-\frac{\langle x', x'\rangle}{2\sigma^2}\right)$$

- Since $\langle x, x'\rangle$ is a kernel, according to the rules in the previous slide the gaussian function is a valid kernel.

Gaussian kernel

It is in the form

$$k(x, x') = \exp(-\|x - x'\|^2 / 2\sigma^2)$$

Notice that

$$\|x - x'\|^2 = \langle x, x \rangle + \langle x', x' \rangle - 2\langle x, x' \rangle$$

and

$$k(x, x') = \exp\left(-\frac{\langle x, x \rangle}{2\sigma^2}\right) \exp\left(\frac{\langle x, x' \rangle}{\sigma^2}\right) \exp\left(-\frac{\langle x', x' \rangle}{2\sigma^2}\right)$$

- Since $\langle x, x' \rangle$ is a kernel, according to the rules in the previous slide the gaussian function is a valid kernel.
- Note that the feature vector that corresponds to the Gaussian kernel has infinite dimensionality.

Various methods

- Kernel SVM
- Kernel Fisher discriminant
- Kernel logistic regression
- Kernel linear and ridge regression
- Kernel SVD or PCA

Outline

Summary

- Support vector machine
  - linear SVM (separable and nonseparable)
  - feature space and kernel SVM
- Kernel-based learning

Outline

Preliminaries on optimization theory

The optimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_i(x) \geq 0, \quad i = 1, \ldots, m \end{array}$$

where $x \in R^n$ and $f(x), g_i(x)$ are differentiable.

- Lagrange function

$$L(x, \alpha) = f(x) - \sum_i \alpha_i g_i(x)$$

- KKT conditions

$$\begin{cases} \nabla_x L(x^*, \alpha^*) = \nabla f(x^*) - \sum_i \alpha_i^* \nabla g_i(x^*) = 0 \\ g_i(x^*) \geq 0, \alpha_i^* \geq 0, \alpha_i^* g_i(x^*) = 0, \quad i = 1, \ldots, m \end{cases}$$

- The dual function

$$\phi(\alpha) = \inf_x L(x, \alpha) = \inf_x \left(f(x) - \sum_i \alpha_i g_i(x)\right)$$

- The dual problem

$$\begin{array}{ll} \max & \phi(\alpha) \\ \text{s.t.} & \alpha_i \geq 0, \quad i = 1, \ldots, m \end{array}$$

- Properties
  - $\phi(\alpha^*) \leq f(x^*)$
  - If $f(x), -g_i(x)$ are all convex, and some other mild conditions are satisfied, then we have: $\phi(\alpha^*) = f(x^*)$, and KKT conditions are both sufficient and necessary optimal conditions.

Further reading

- J.C. Burges
  A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121-167, 1998
- Alex Smola and Bernhard Schoelkopf
  A tutorial on support vector regression. Statistics and Computing. 14(3):199-222, 2004.
- J. Platt
  Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods Support Vector Learning, pages 185-208, Cambridge, MA, 1999. MIT Press.

Coffee Time

最强大脑：机器VS人类

王峰VS小度：匆匆那年

两轮比赛分别需要根据儿时照片识别真人和根据真人识别儿时照片

难点：年龄跨度长达二十多岁

结果：2：3（共三题）

孙亦廷VS小度：人声识别

由嘉宾周杰伦在 21 位专业合唱团员中任选出三位歌唱者进行现场通话，而人机需要共同根据通话中的只言片语，在随后的合唱表演中将三位歌唱者找出。

难点：唱歌时的声音与说话时不同；合唱团的成员音质比较相似

结果：1：1（共三题）

王昱珩VS小度：核桃计划

依据模糊的视频图像从现场 30 名「嫌疑人」中找到 3 名真正的「盗贼」

难点：弱光、模糊、动态

结果：0：2（共三题）

小度的成功

- 大规模的深度学习平台
- 大量的训练数据