《数据挖掘技术》

⭐ CH07 支持向量机

➡️ Created by *Wang JingHui*

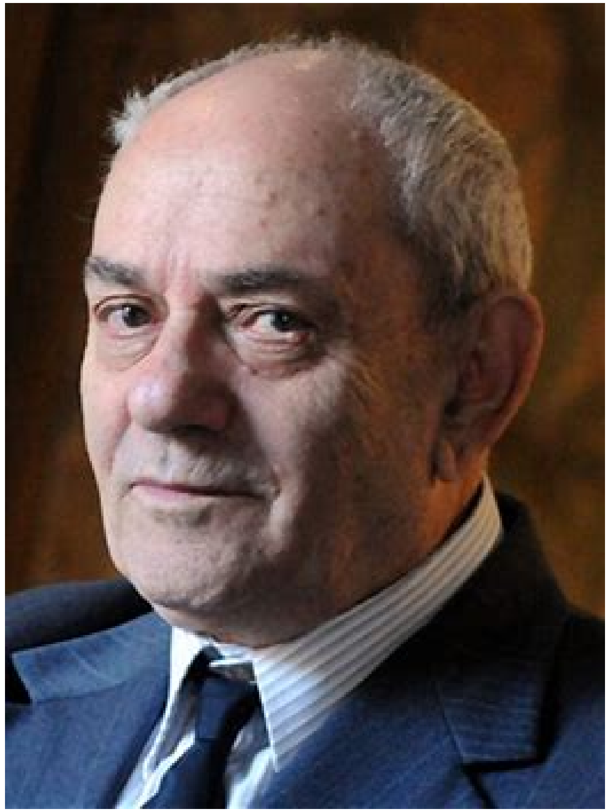
➡️ Version: *4.0*

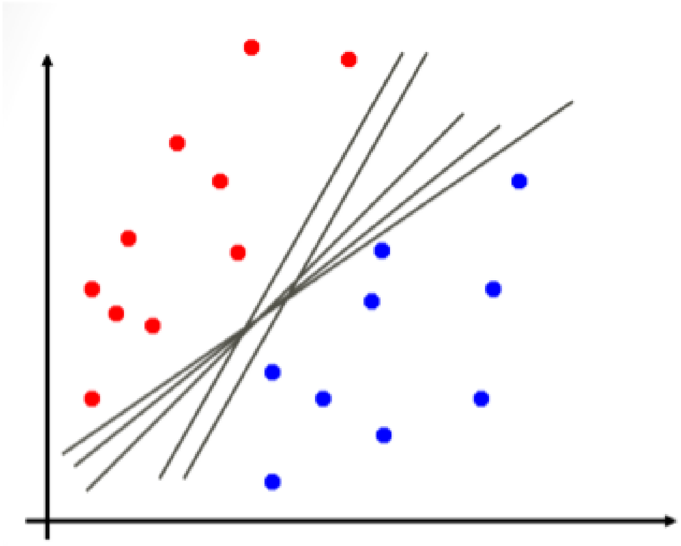
主要内容

1. 线性可分支持向量机与硬间隔最大化
2. 线性支持向量机与软间隔最大化
3. 非线性支持向量机与核函数
4. 序列最小最优化算法

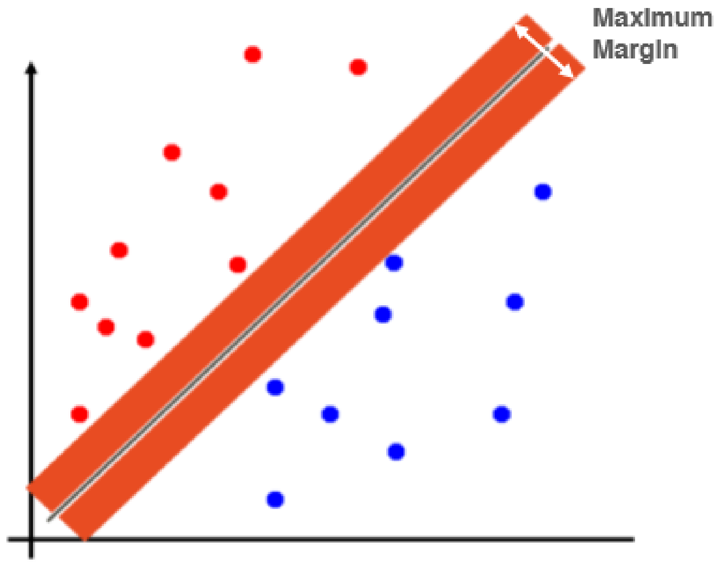
SVM早期工作来自前苏联学者Vladimir N. Vapnik和Alexander Y. Lerner在1963年发表的研究。

- 1964年，Vapnik和Alexey Y. Chervonenkis对广义肖像算法进行了讨论并建立了硬边距的线性SVM。
- 在二十世纪70-80年代，模式识别中最大边距决策边界的理论、松弛变量的规划问题求解技术的出现和VC维的提出，SVM理论化成为统计学习理论的一部分。
- 1992年，Bernhard E. Boser、Isabelle M. Guyon和Vapnik通过核方法得到了非线性SVM。
- 1995年，Corinna Cortes和Vapnik提出了软边距的非线性SVM并将其应用于手写字符识别问题，这份研究在发表后为SVM在各领域的应用提供了参考。





There are many lines that can be linear classifiers.
Which one is the optimal classifier ?



- Define the margin of a linear classifier as the width that the boundary could be increase by before hitting a data data point.[1]

- The maximum margin linear classifier is the simplest kind of SVG(called Linear SVM) [2]

线性可分支持向量机与硬间隔最大化

线性可分支持向量机

线性可分支持向量机（硬间隔支持向量机）：给定线性可分训练数据集，通过间隔最大化或等价地求解相应地凸二次规划问题学习得到分离超平面为

$$w^* \cdot x + b^* = 0$$

以及相应的分类决策函数

$$f(x) = sign(w^* \cdot x + b^*)$$

称为线型可分支持向量机。

函数间隔和几何间隔

函数间隔

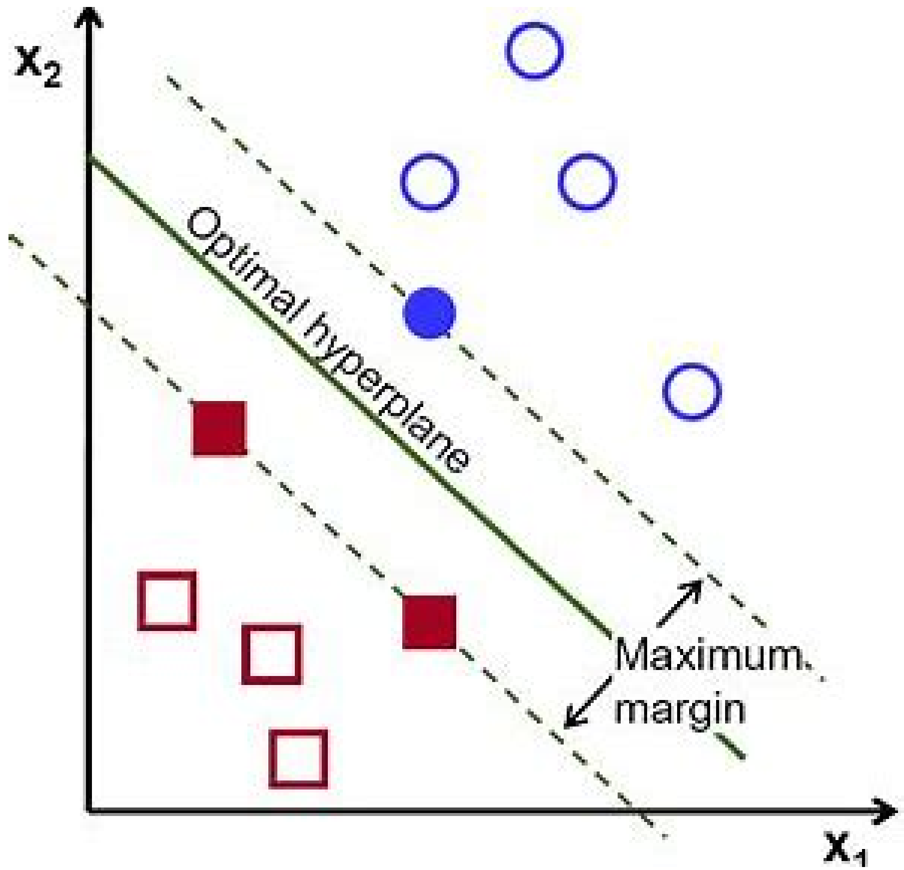
对于给定数据集T和超平面(w,b)，定义超平面(w,b)关于样本点(x_i,y_i)的函数间隔为

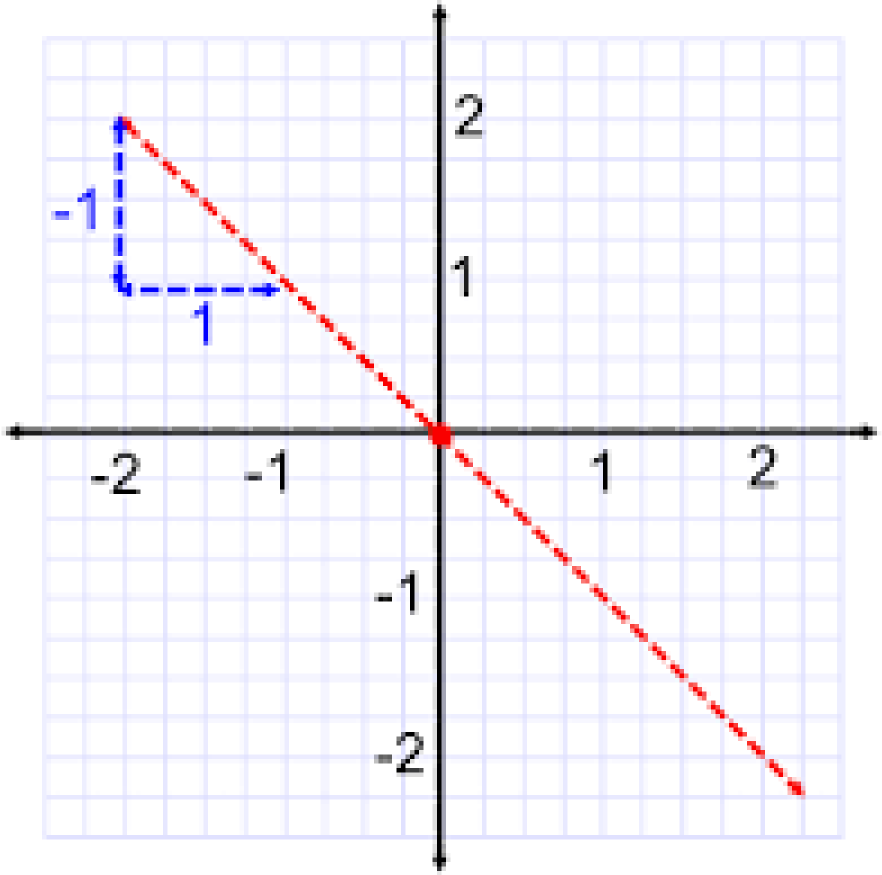
$$\hat{\gamma}_i = y_i(w \cdot x_i + b)$$

定义超平面(w,b)关于训练数据集T的函数间隔为超平面(w,b)关于T中所有样本点(x_i,y_i)的函数间隔之最小值，即

$$\hat{\gamma} = \min_{i=1,\cdots,N} \hat{\gamma}_i$$

函数间隔可以表示分类预测的**正确性**及**确信度**。


x2
Optimal hyperplane
Maximum
margin
x1


-1
1
2
1
-2
-1
1
2
-1
-2

几何间隔

超平面(w, b)关于样本点(x_i, y_i)的几何间隔为

$$\gamma_i = y_i \left(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right)$$

超平面(w, b)关于训练集T的几何间隔

$$\gamma = \min_{i=1,2,\cdots,N} \gamma_i$$

即超平面(w, b)关于训练集T中所有样本点(x_i, y_i)的几何间隔的最小值。

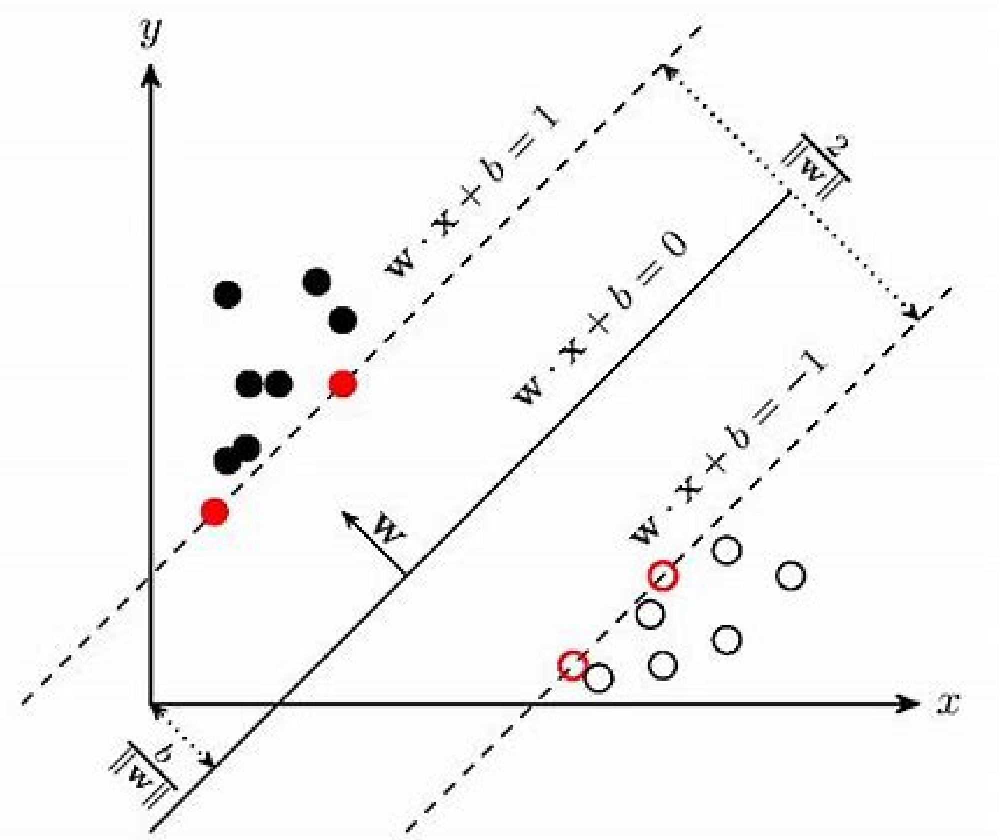
函数间隔和几何间隔的关系

$$\gamma_i = \frac{\hat{\gamma}_i}{\|w\|}$$

$$\gamma = \frac{\hat{\gamma}}{\|w\|}$$

如果$||w|| = 1$,那么函数间隔和几何间隔相等。

间隔表示



y
x
$\mathbf{w} \cdot \mathbf{x} + b = 1$
$\mathbf{w} \cdot \mathbf{x} + b = 0$
$\mathbf{w} \cdot \mathbf{x} + b = -1$
$\frac{2}{\|\mathbf{w}\|}$
$\frac{b}{\|\mathbf{w}\|}$
$\mathbf{w}$

间隔最大化

最大间隔分离超平面

最大间隔分离超平面等价为求解

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

等价的

$$
\begin{aligned}
&\max_{w,b} \quad \frac{\hat{\gamma}}{\|w\|} \\
&s.t. \quad y_i\left(w\cdot x_i+b\right)\geq\hat{\gamma},\quad i=1,2,\cdots,N
\end{aligned}
$$

最大化$\frac{1}{||w||}$和最小化$\frac{1}{2}||w||^2$是等价的

$$
\begin{aligned}
&\min_{w,b} \quad \frac{1}{2}\|w\|^2 \\
&s.t. \quad y_i\left(w \cdot x_i + b\right) - 1 \geq 0, \quad i = 1, 2, \cdots, N
\end{aligned}
$$

线性可分支持向量机学习算法-最大间隔法

输入：线性可分训练数据集$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中$x_i \in \mathcal{X}=R^n, y_i \in \mathcal{Y}=\{+1,-1\}, i=1,2,\cdots,N$

输出：最大间隔分离超平面和分类决策函数

1. 构建并求解约束最优化问题

$$\min_{w,b} \frac{\hat{\gamma}}{\|w\|}$$

$$s.t. \quad y_i(w\cdot x_i+b)-1 \geqslant 0, i=1,2,\ldots,N$$

这是个凸二次规划问题,求出了上述方程的解w^*, b^*。

2. 分离超平面 $w^* \cdot x + b^* = 0$
3. 相应的分类决策函数 $f(x)=sign(w^* \cdot x + b^*)$

支持向量和间隔边界

由于支持向量在确定分离超平面中起着决定作用，所以将这种分类模型称为支持向量机。

支持向量对应

$$y_i(w \cdot x_i + b) - 1 = 0$$

上式对应两个超平面

分离超平面对应

$$w \cdot x + b = 0$$

注意，在算法7.1中，并没有说明这个问题如何求得最优解，在7.1.4节中有描述该如何求解。

对偶算法

1. 对偶问题往往更容易求解
2. 自然引入核函数，进而推广到非线性分类问题。

构造拉格朗日函数

针对每个不等式约束，定义拉格朗日乘子$\alpha_i \geq 0$，定义拉格朗日函数

$$
\begin{aligned}
L(w,b,\alpha) &= \frac{1}{2} w \cdot w - \left[\sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1]\right] \\
&= \frac{1}{2}\|w\|^2 - \left[\sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1]\right] \\
&= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^{N} \alpha_i
\end{aligned}
$$

$$
\alpha_i \geqslant 0, i = 1, 2, \ldots, N
$$

其中$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$为拉格朗日乘子向量

原始问题是极小极大问题

根据**拉格朗日对偶性**，原始问题的**对偶问题是极大极小问题**:

$$\max_{\alpha} \min_{w,b} L(w,b,\alpha)$$

(1) 求解 $\min\limits_{w,b} L(w,b,\alpha)$，求偏导数并令其等于0。

得到:

$$\min_{w,b} L(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i$$

(2) 求$\max\limits_{\alpha} \min\limits_{w,b} L(w,b,\alpha)$

得到转换后的对偶问题

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N} \alpha_i$$

$$s.t. \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i \geqslant 0, i = 1, 2, \ldots, N$$

对于任意线性可分的两组点，他们在分类超平面上的投影都是线性不可分的。

α不为零的点对应的实例为支持向量，通过支持向量可以求得b值

核心公式两个

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$$

$$b^* = y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j)$$

这里面比较重要的是b^*的公式的理解，通过$\arg\max \alpha^*$实现，因为支持向量共线，所以通过任意支持向量求解都可以。

支持向量

原始最优化问题及对偶最优化问题，将训练数据集中对应于$\alpha_i^* > 0$的样本点(x_i, y_i)的实例$x_i \in \mathcal{R}^n$称为支持向量。

支持向量一定在间隔边界上。

例: 正例$x_1 = (3,3)^T, x_2 = (4,3)^T$，负例$x_3 = (1,1)^T$

解:对偶问题是

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{j=1}^{N} \alpha_i$$

$$= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$s.t. \quad \alpha_1 + \alpha_2 - \alpha_3 = 0$$

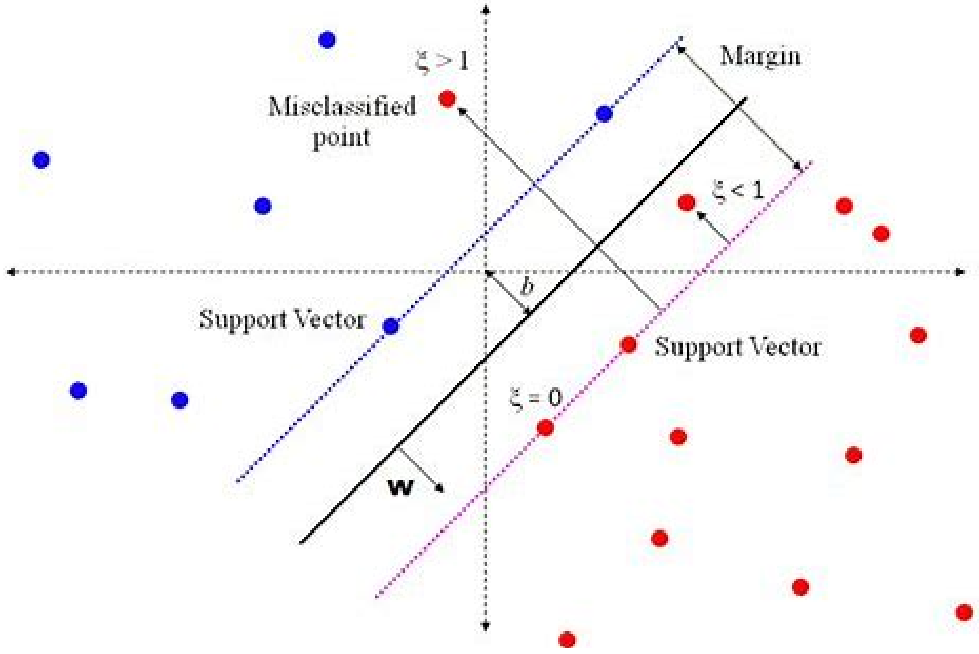
$$\alpha_i \geqslant 0, i = 1, 2, \ldots, N$$

线性支持向量机与软间隔最大化

线性支持向量机

$$
\begin{aligned}
& \min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\
& s.t. \quad y_i(w \cdot x_i + b) \geqslant 1 - \xi_i, i = 1, 2, \ldots, N \\
& \qquad\quad \xi_i \geqslant 0, i = 1, 2, \ldots, N
\end{aligned}
$$

ξ_i 为松弛变量，$C > 0$ 为惩罚参数。



松弛变量

若所研究的线性规划模型的约束条件全是小于类型，那么可以通过标准化过程引入M个非负的松弛变量。

对偶问题描述

原始问题里面有两部分约束，涉及到两个拉格朗日乘子向量

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

$$0 \leqslant \alpha_i \leqslant C, i = 1, 2, \ldots, N$$

通过求解对偶问题， 得到α，然后求解w, b的过程和之前一样

线性支持向量机的解w^*唯一但b^*不一定唯一

线性支持向量机是线性可分支持向量机的超集。

支持向量

在线性不可分的情况下，对偶问题的解，将训练数据集中对应于$\alpha_i^* > 0$的样本点(x_i, y_i)的实例$x_i \in \mathcal{R}^n$称为支持向量。

软间隔的支持向量x_i或者在间隔边界上，或者在间隔边界与分离超平面之间，或者在分离超平面误分一侧。

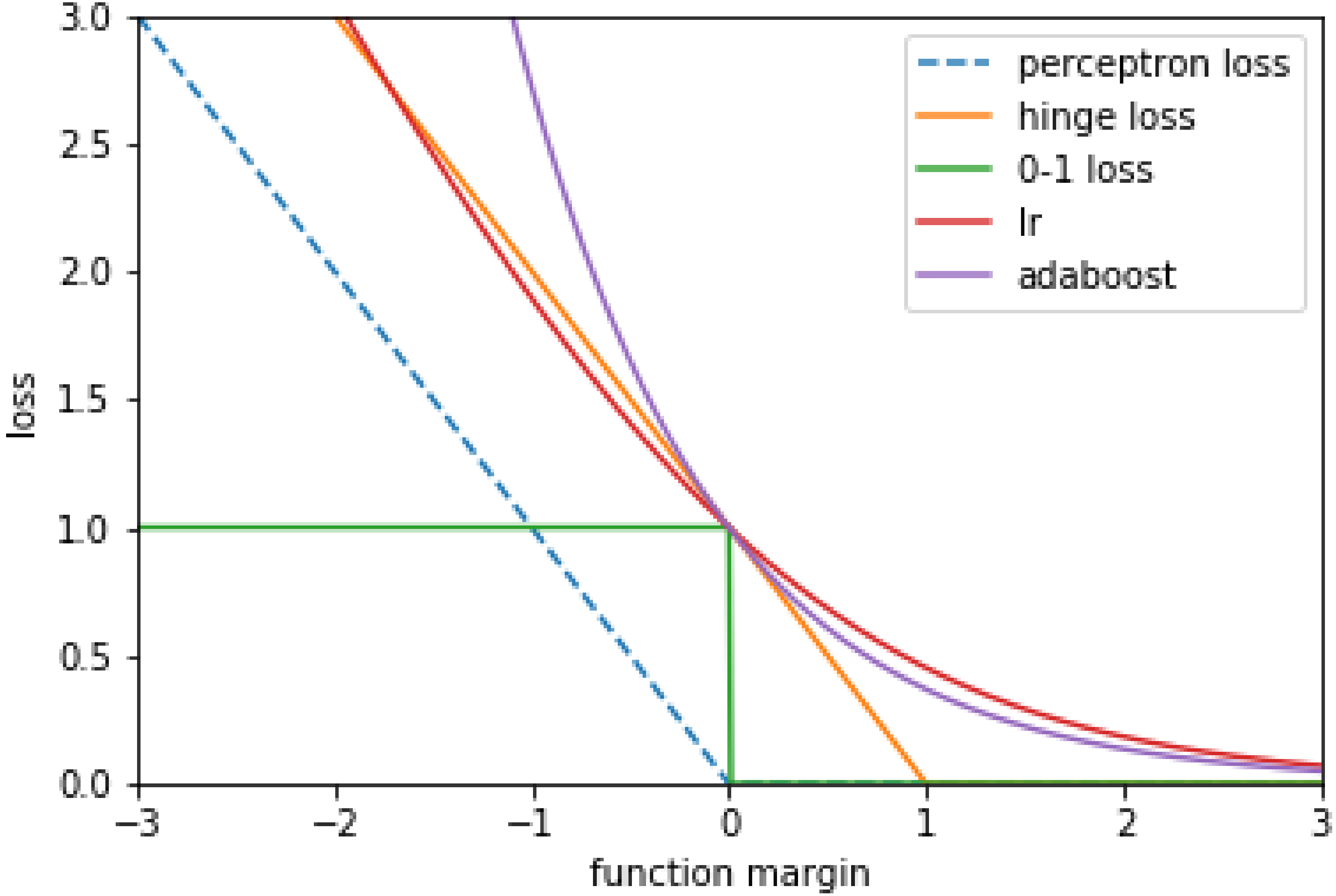
合页损失

- 最小化目标函数

$$\min_{w,b} \sum_{i=1}^{N} [1 - y_i(w \cdot x + b)]_+ + \lambda \|w\|^2$$

其中

- 第一项是经验损失或经验风险，函数$L(y(w \cdot x + b)) = [1 - y(w \cdot x + b)]_+$称为合页损失，可以表示成$L = \max(1 - y(w \cdot x + b), 0)$
- 第二项是**系数为**λ**的**w**的**L_2**范数的平方**，是正则化项



perceptron loss
hinge loss
0-1 loss
lr
adaboost
loss
function margin
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−3
−2
−1
0
1
2
3

总结:

- 0-1损失函数不是连续可导
- 合页损失认为是0-1损失函数的上界,
- 感知机误分类驱动， 选择函数间隔作为损失考虑分类的正确性，合页损失不仅要考虑分类正确， 还要考虑确信度足够高时损失才是0.

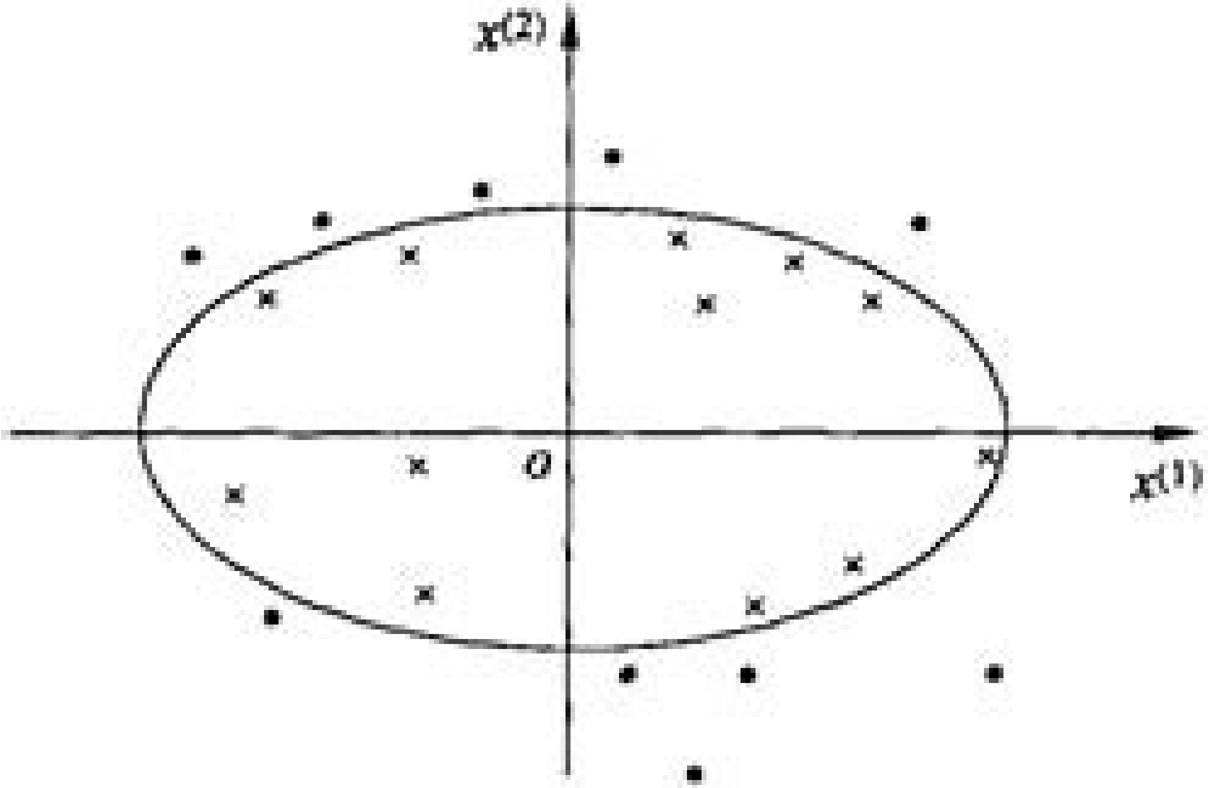
非线性支持向量机与核函数

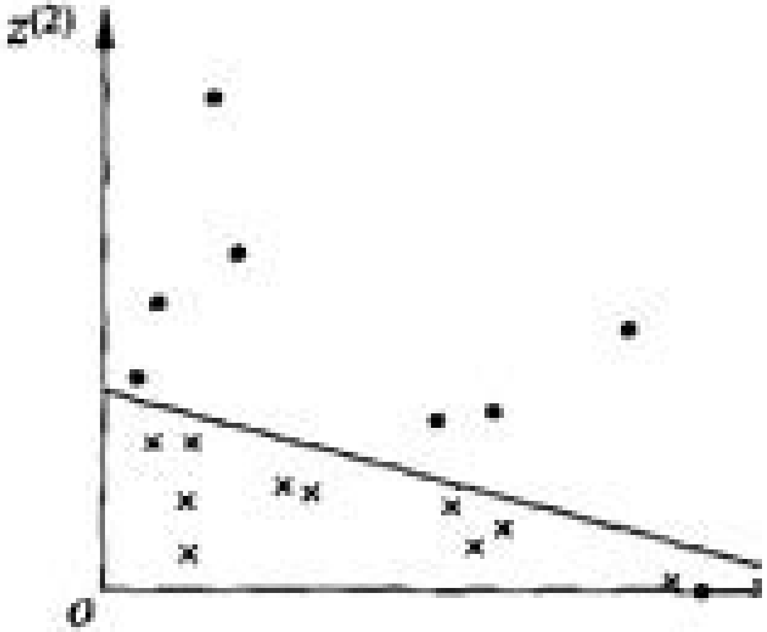
核技巧的想法是在学习和预测中只定义核函数$K(x, z)$，而不是显式的定义映射函数ϕ

- 非线性分类问题
- 进行非线形变换，将非线形问题变换为线性问题；

核技巧

基本想法： 通过一个非线性变换将输入空间（欧式空间R^n或离散集合）对应于一个特征空间(希尔伯特空间$\mathcal{H}$)，使得在输入空间R^n中的超曲面对应于特征空间$\mathcal{H}$中的一个超平面模型。





核函数定义

- 设$\mathcal{X}$是输入空间（欧氏空间R^n的子集或离散集合），$\mathcal{H}$是特征空间（希尔伯特空间），如果存在一个从$\mathcal{X}$到$\mathcal{H}$的映射

$$\phi(x): \mathcal{X} \rightarrow \mathcal{H}$$

使得对所有$x, z \in \mathcal{X}$，函数$K(x, z)$满足条件

$$K(x, z) = \phi(x) \cdot \phi(z)$$

则称$K(x, z)$为核函数，$\phi(x)$为映射函数，式中$\phi(x) \cdot \phi(z)$为$\phi(x)$和$\phi(z)$的内积。

对于给定的核$K(x,z)$，特征空间$\mathcal{H}$和映射函数$\phi(x)$的取法并不唯一，可以取不同的特征空间，即便是同一特征空间里也可以取不同的映射

注意这个例子里面$\phi(x)$实现了从低维空间到高维空间的映射。

$$K(x,z)=(x\cdot z)^2$$
$$\mathcal{X}=\mathbb{R}^2, x=(x^{(1)},x^{(2)})^T$$
$$\mathcal{H}=\mathbb{R}^3, \phi(x)=((x^{(1)})^2,\sqrt{2}x^{(1)}x^{(2)},(x^{(2)})^2)^T$$
$$\mathcal{H}=\mathbb{R}^4, \phi(x)=((x^{(1)})^2,x^{(1)}x^{(2)},x^{(1)}x^{(2)},(x^{(2)})^2)^T$$

提示：

- 理解下$\mathbb{R}^n$
- 理解下计算K要比通过ϕ计算K要容易
- 核函数的定义相当于给出了$\phi(x)\cdot\phi(z)$的结果，而没有显式的给出ϕ的定义，ϕ实现了从输入空间到特征空间的变换，所以说，学习是隐式的从特征空间中进行的，不需要显式的定义特征空间和映射函数，这样的技巧称为核技巧，通过线性分类学习方法和核函数解决非线性问题。

核具有再生性即满足

$$
K(\cdot,x)\cdot f=f(x)\\
K(\cdot,x)\cdot K(\cdot,z)=K(x,z)
$$

称为再生核

线性支持向量机

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

$$0 \leqslant \alpha_i \leqslant C, i = 1, 2, \ldots, N$$

其中(x_i, x_j)可以替换：

$$\min_{\alpha} \ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i$$

通常，直接计算$K(x,z)$比较容易，而通过$\phi(x)$和$\phi(z)$计算$K(x,z)$并不容易。

$$W(\alpha)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{N}\alpha_i$$

分类决策函数

$$f(x)=sign\left(\sum_{i=1}^{N_s}\alpha_i^* y_i \phi(x_i)\cdot\phi(x)+b^*\right)=sign\left(\sum_{i=1}^{N_s}\alpha_i^* y_i K(x_i,x)+b^*\right)$$

学习是隐式地在特征空间进行的，不需要显式的定义特征空间和映射函数。这样的技巧称为核技巧，核技巧不仅引用于支持向量机，而且应用于其他统计学习问题。

常用核函数:

1. 线性核

$$K(x_i, x_j) = x_i^T x_j$$

2. 多项式核函数

$$K\left(x_i, x_j\right) = \left(x_i^T x_j\right)^d$$

3. 高斯核函数

$$K\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)$$

4. 拉普拉斯核

$$K\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|}{\sigma}\right)$$

非线性支持向量分类机

构建最优化问题：

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i$$

$$s.t. \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$0 \leqslant \alpha_i \leqslant C, i = 1, 2, \ldots, N$$

求解得到$\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*)^T$

选择α^*的一个正分量计算

$$b^* = y_j - \sum_{i=1}^{N} \alpha_i^* y_i K(x_i, x_j)$$

构造决策函数

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i^* y_i K(x, x_i) + b^*\right)$$

学习算法：序列最小最优化

支持向量机的学习问题可以形式化为求解凸二次规划问题。

问题描述

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i$$

$$s.t. \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$0 \leqslant \alpha_i \leqslant C, i = 1, 2, \ldots, N$$

这个问题中，变量是α，一个变量α_i对应一个样本点(x_i, y_i)，变量总数等于N

SMO算法

整个SMO算法包括两**部分**：

1. 求解两个变量二次规划的解析方法
2. 选择变量的启发式方法

$$
\begin{aligned}
& \min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N} \alpha_i \\
s.t. \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& 0 \leqslant \alpha_i \leqslant C, i = 1, 2, \ldots, N
\end{aligned}
$$

↘️ Enjoy your machine learning!

https://github.com/wjssx/

E-mail: csr_dsp@sina.com