分类方法-丁兆云
——SVM

主要内容

- 1.了解SVM
- 2. 深入SVM
- 3. 非线性SVM

1. 了解SVM

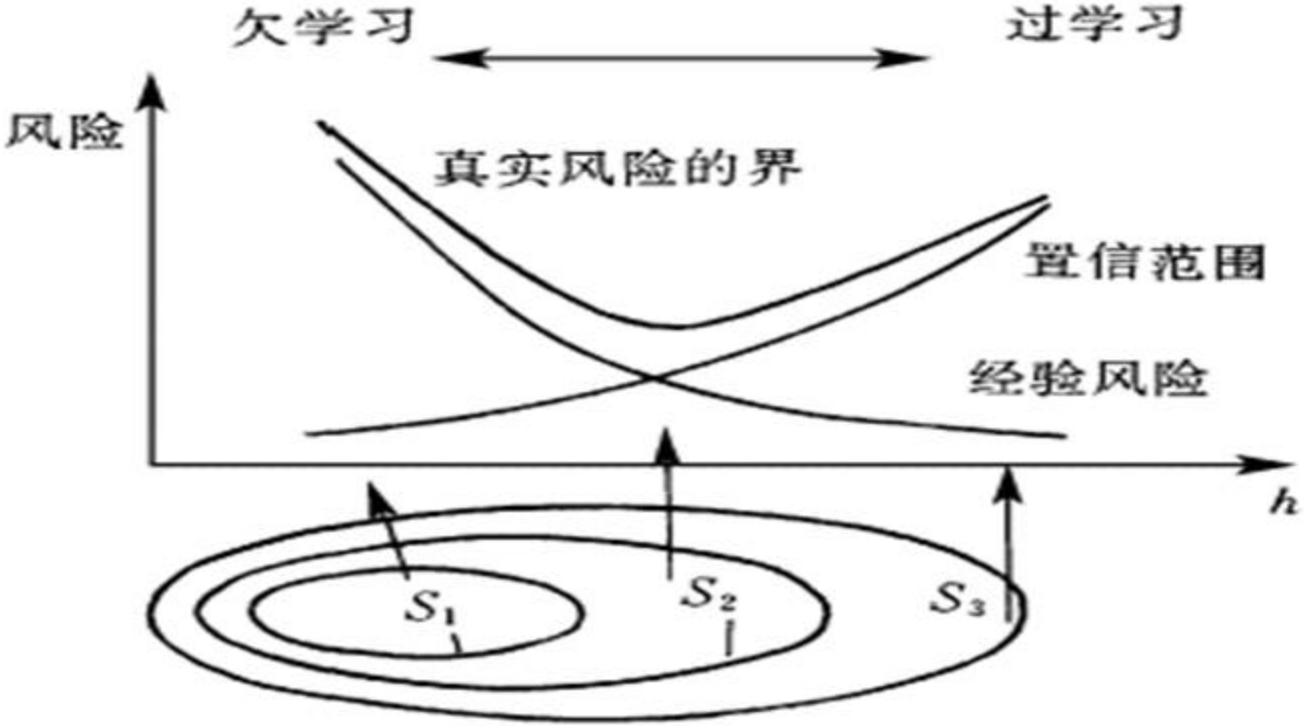
- 名称
  - 支持向量机
  - Support Vector Machine
- 简介
  - SVM是一种二分类模型，是特征空间上的间隔最大的线性分类学，其学习策略是间隔最大化，最终可转化为一个凸二次规划问题求解
- 来历
  - 第一篇论文由Vladimir Vapnik （弗拉基米尔·万普尼克）和他的同事于1992年发表

1. 了解SVM

- 传统的统计模式识别方法在进行机器学习时，强调经验风险最小化。而单纯的经验风险最小化会产生“过学习问题”，其推广能力较差。
- 推广能力是指：将学习机器(即预测函数，或称学习函数、学习模型)对未来输出进行正确预测的能力。

1. 了解SVM

- “过学习问题”：某些情况下，当训练误差过小反而会导致推广能力的下降。



1. 了解SVM

- 根据统计学习理论，学习机器的实际风险由经验风险值和置信范围值两部分组成。而基于经验风险最小化准则的学习方法只强调了训练样本的经验风险最小误差，没有最小化置信范围值，因此其推广能力较差。
- Vapnik 与1995年提出的支持向量机（Support Vector Machine, SVM）以训练误差作为优化问题的约束条件，以置信范围值最小化作为优化目标，即SVM是一种基于结构风险最小化准则的学习方法，其推广能力明显优于一些传统的学习方法。

1. 了解SVM

- 由于SVM 的求解最后转化成二次规划问题的求解，因此SVM 的解是全局唯一的最优解
- SVM在解决小样本、非线性及高维模式识别问题中表现出许多特有的优势，并能够推广应用到函数拟合等其他机器学习问题中.

1.1 分类标准起源：线性分类器

- 线性分类器的目标
  - 假设x表示数据点，y表示类别（-1,1）
  - 目标：找一个超平面把数据分成两类，超平面方程可表示为：

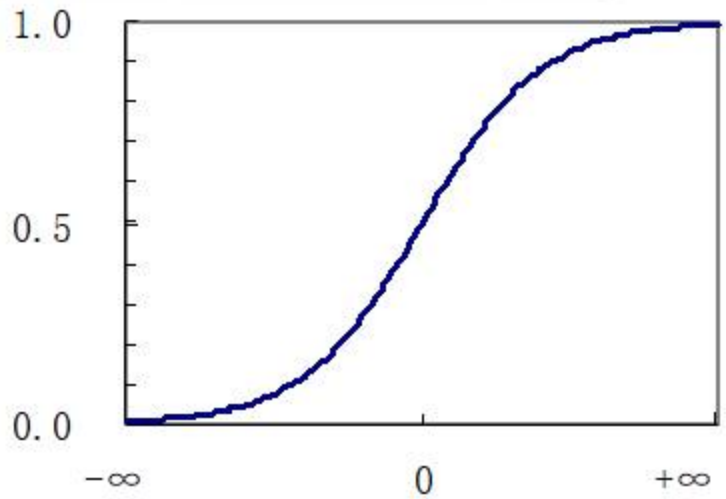
$$w^T x + b = 0$$

  - y的类别取1或者-1，而不是其他的，来源于Logistic回归

1.1 分类标准起源：Logistic 回归

- Logistic回归

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$



  - 将特征的线性组合作为自变量，自变量取值为负无穷到正无穷，使用Logistic函数（sigmoid函数）将自变量映射到(0,1)上
  - x是n维度特征向量，g是Logisitc函数
  - 若 $\theta^T x > 0$,则$h_\theta(x) > 0.5$,x属于y=1的类
  - 若 $\theta^T x \leq 0$,则$h_\theta(x) \leq 0.5$,x属于y=0的类

1.1 分类标准起源：Logistic 回归

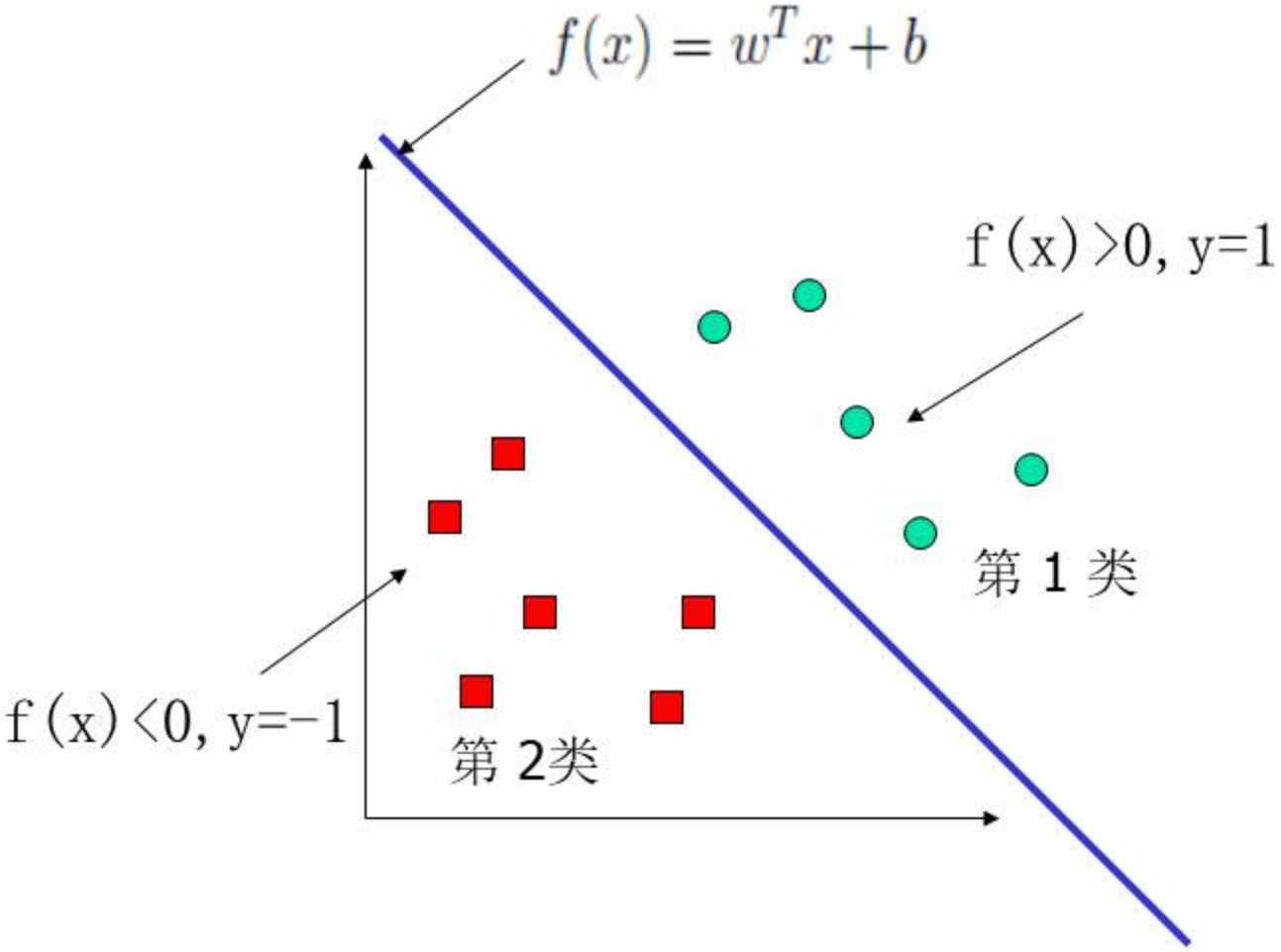
- Logistic变形
  - 将结果标签y=0和y=1替换为y=-1和y=1
  - 将$\theta^T x = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n (x_0 = 1)$中的$\theta_0$替换为b，将$\theta_1 x_1 + \cdots + \theta_n x_n$替换为$w^T x$，有：

$$\theta^T x = w^T x + b.$$

  - 也就是说线性分类函数跟Logistic回归的形式化表示 $h_\theta(x) = g(\theta^T x) = g(w^T x + b)$ 没区别
  - 将上述函数g(z)映射到y=1和y=-1上，映射关系为：

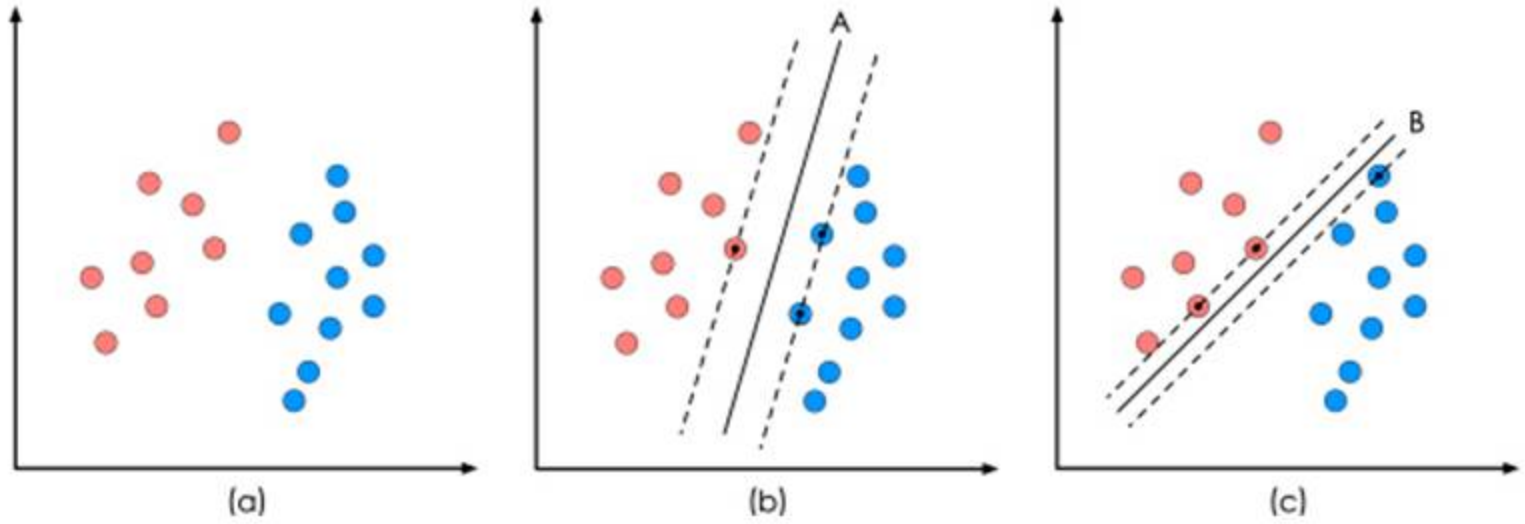
$$g(z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$

1.2 线性分类的一个例子



1.2线性分类的一个例子

- 许多决策边界可以分割这些数据点出为两类，我们选取哪一个?



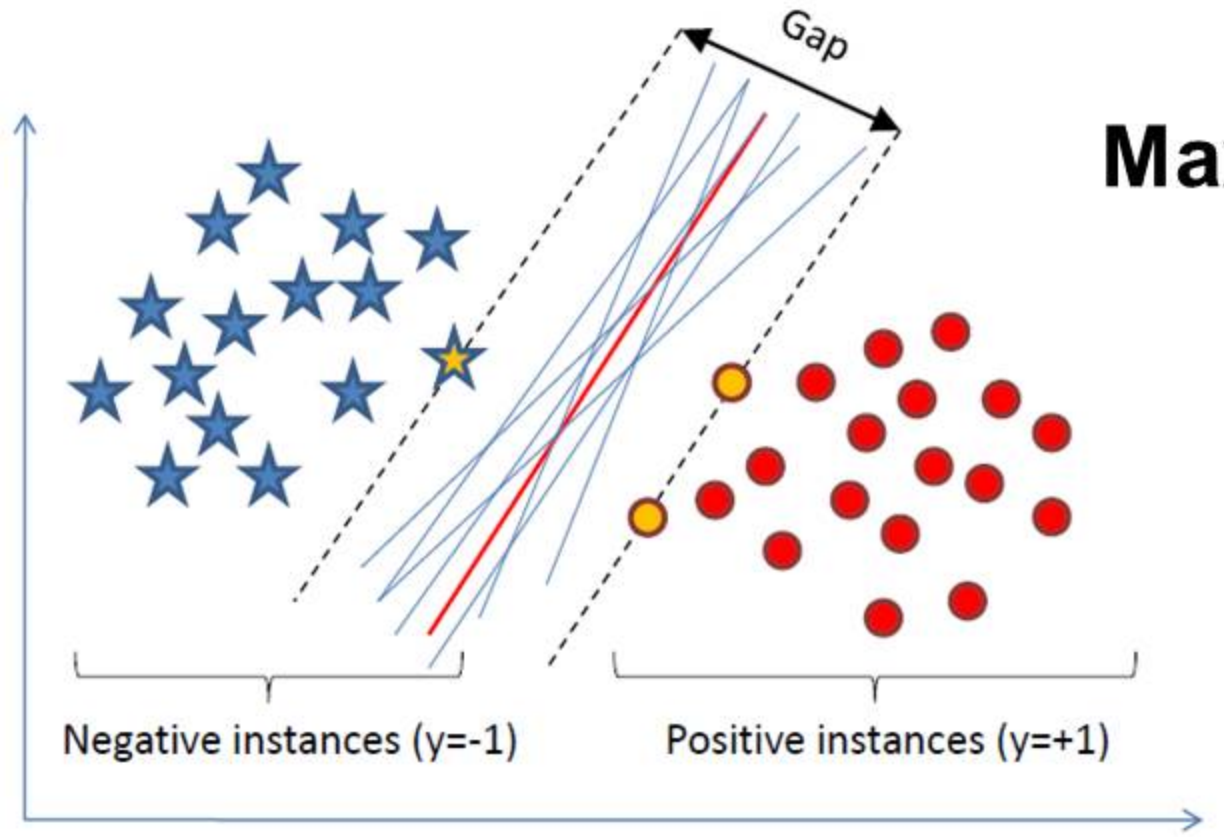
主要内容

- 1.了解SVM
- 2. 深入SVM
- 3. 非线性SVM

2.1最小间隔面推导

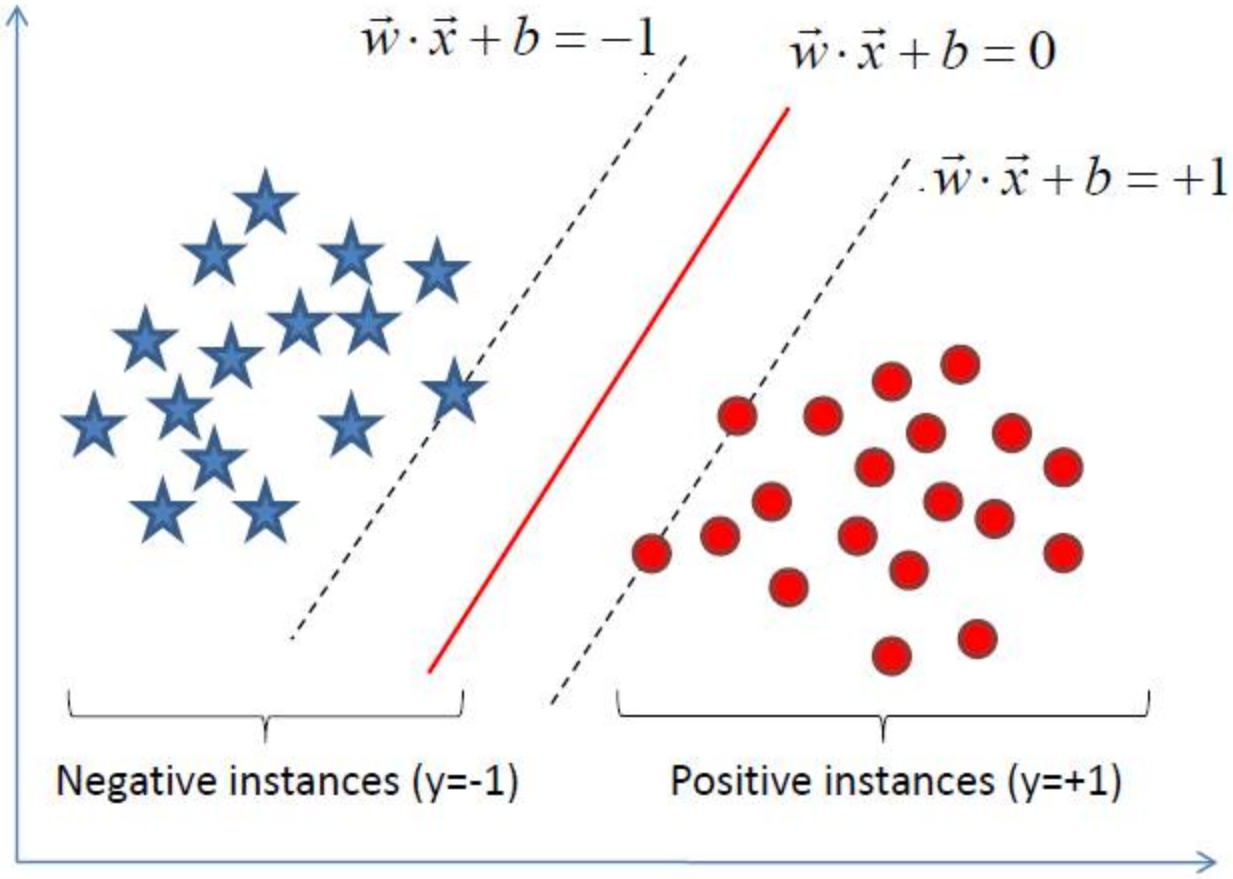
Given training data: $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \in R^n$

$y_1, y_2, \ldots, y_N \in \{-1, +1\}$

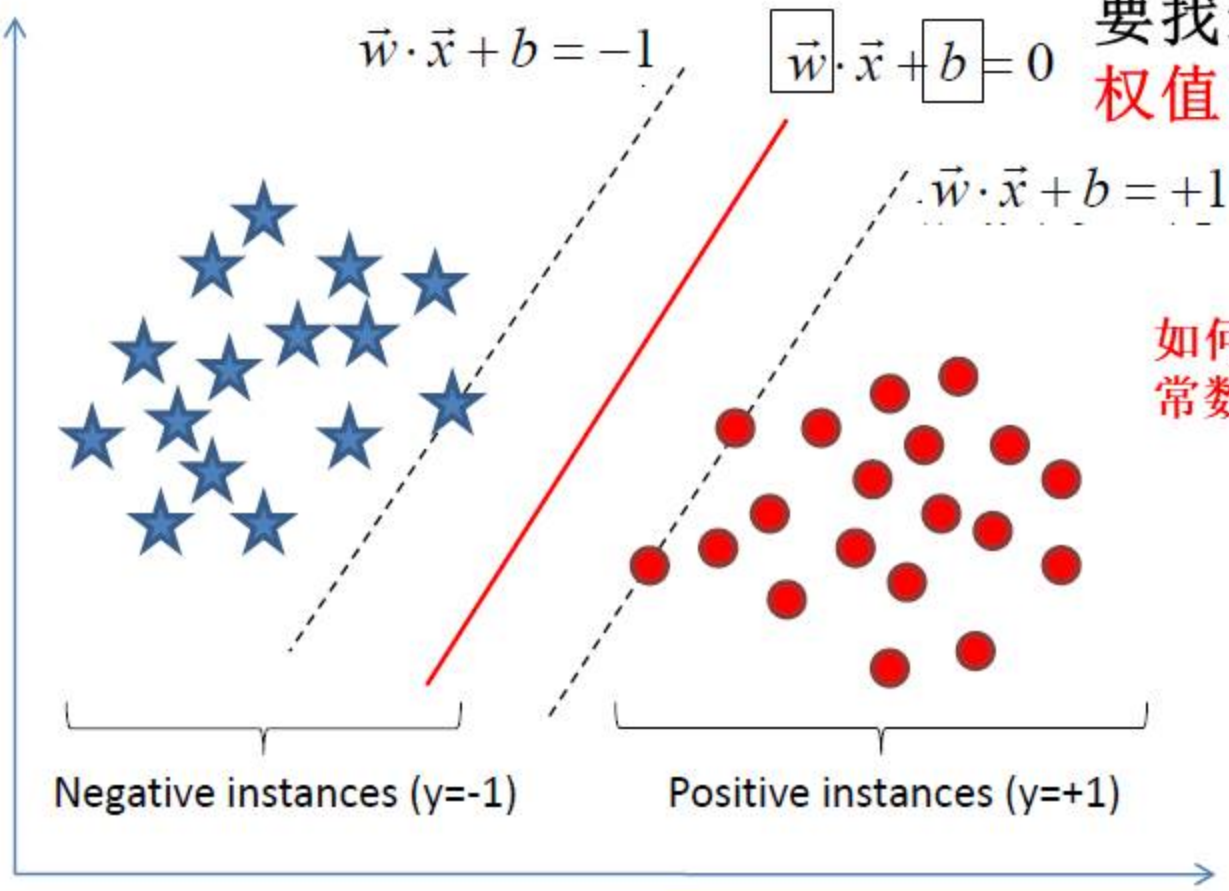


Maximize the Gap!

2.1最小间隔面推导


$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=0$
$\vec{w}\cdot\vec{x}+b=+1$
Negative instances (y=-1)
Positive instances (y=+1)

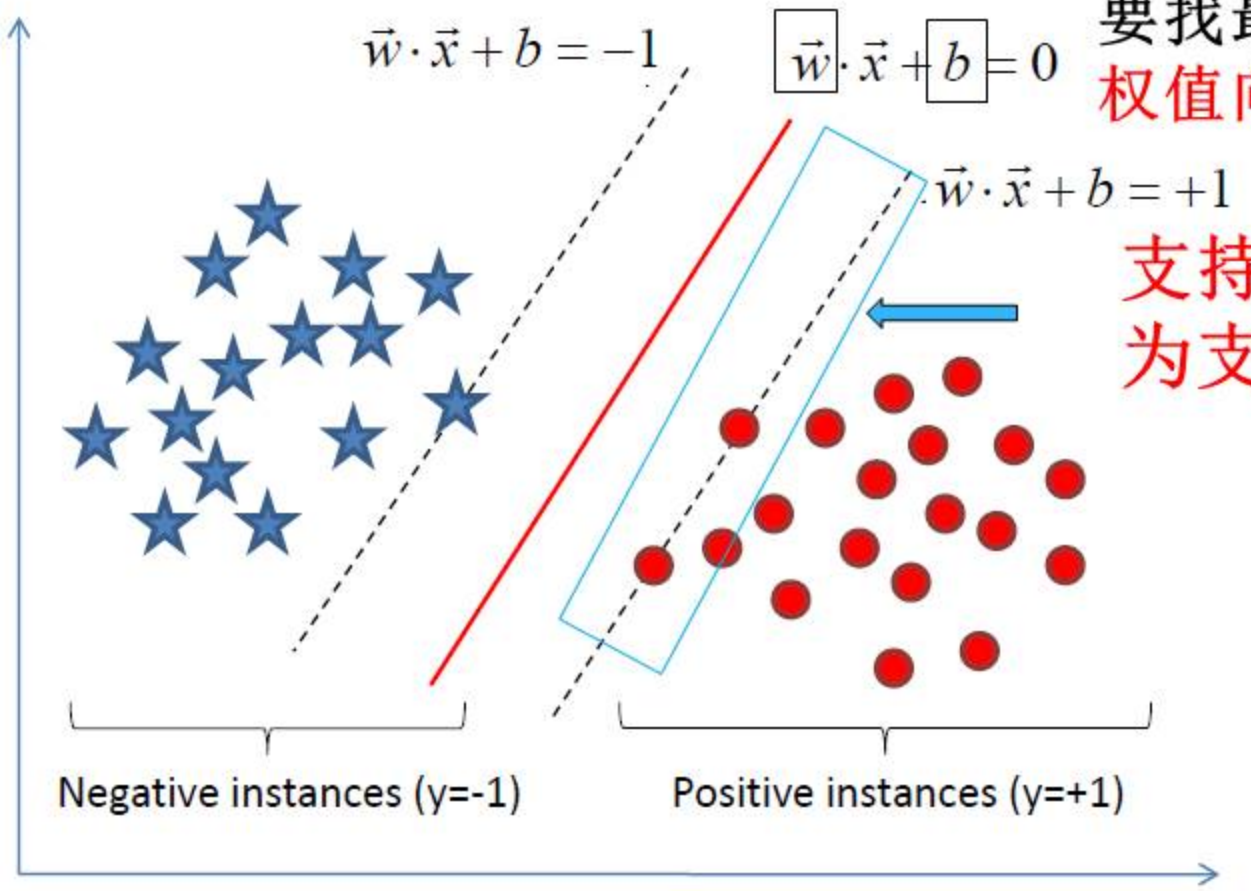
2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

如何确定权值向量w和常数b呢？

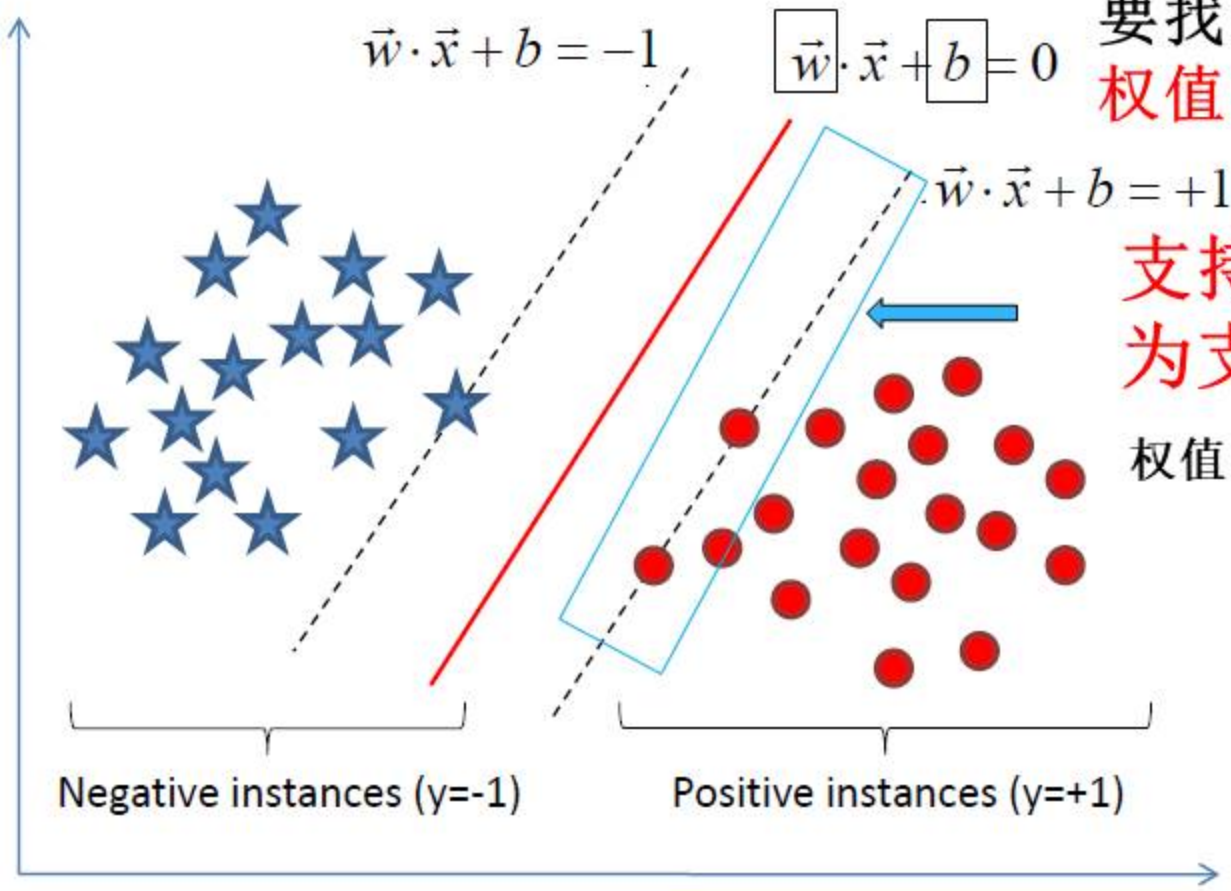
2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

2.1最小间隔面推导

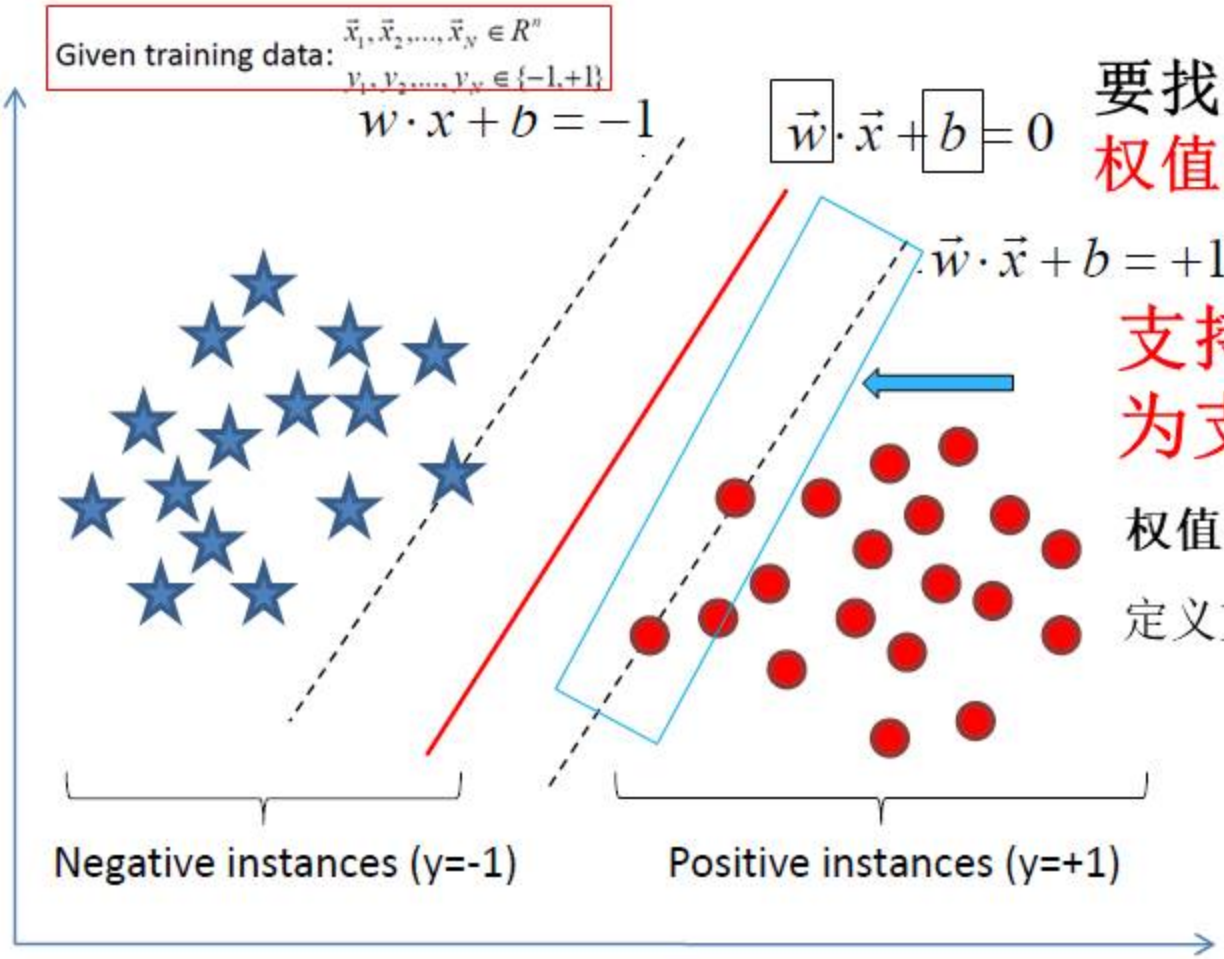


要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

权值向量w和常数b由支持点来决定

2.1最小间隔面推导



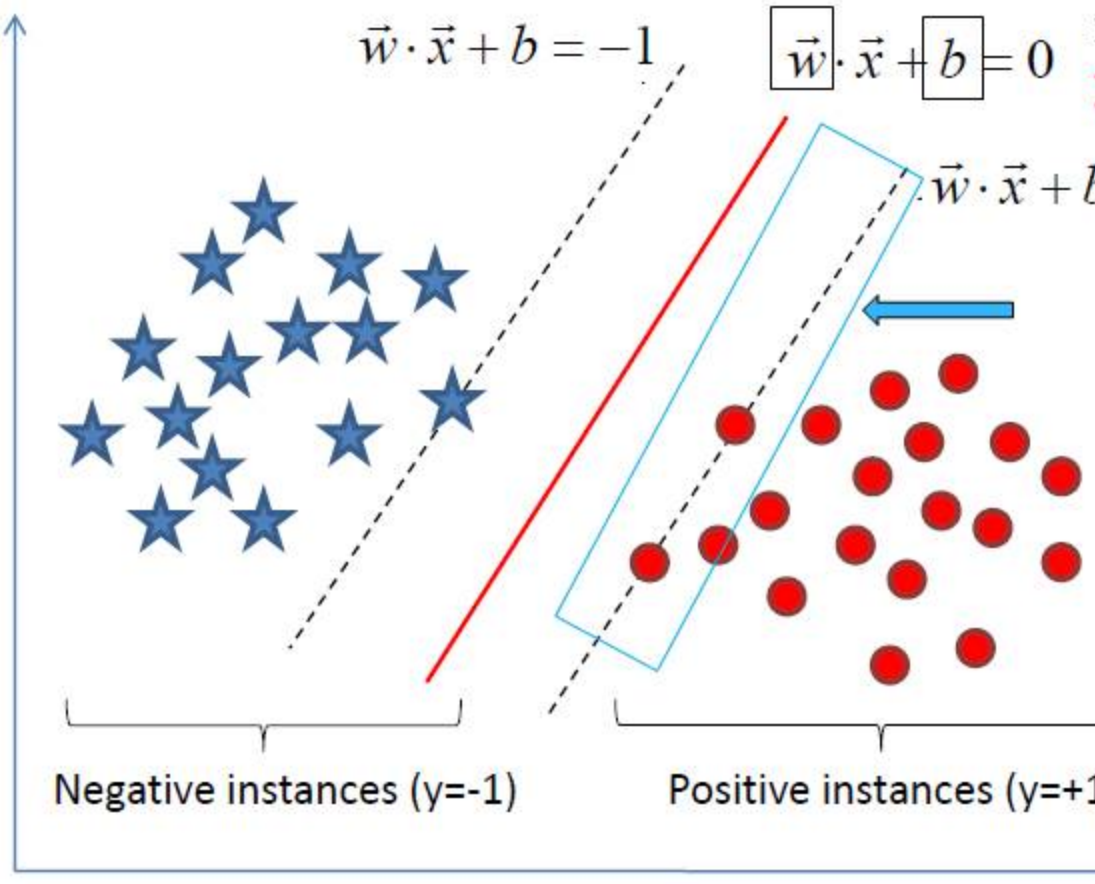
要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

权值向量w和常数b由支持点来决定

定义支持点为一个向量x_i

2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

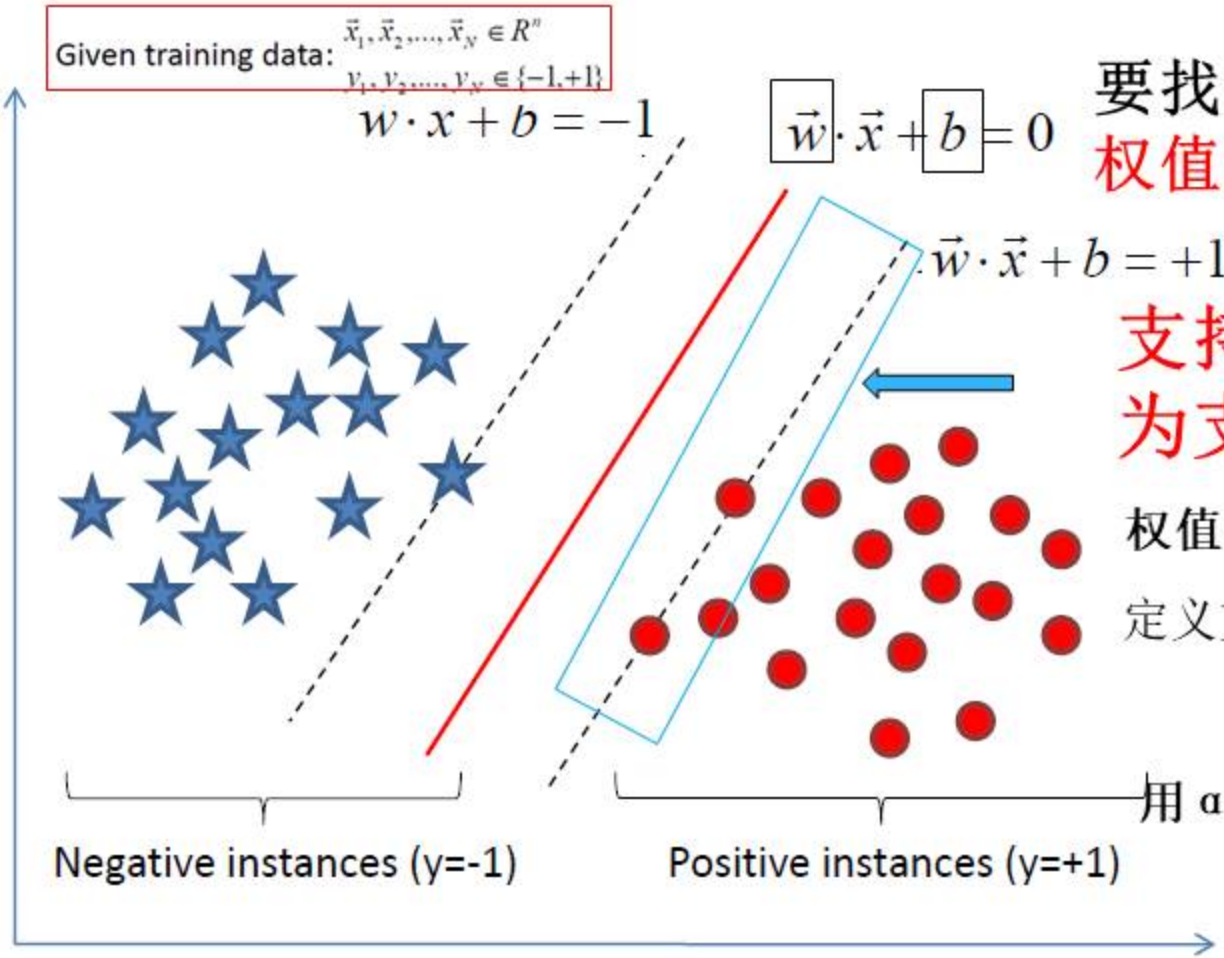
权值向量w和常数b由支持点来决定

定义支持点为一个向量x_i

$w=\alpha_1 x_1+\alpha_2 x_2+\cdots+\alpha_n x_n$

用$\alpha_1 x_1$表示数字系数和向量的乘积

2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

权值向量w和常数b由支持点来决定

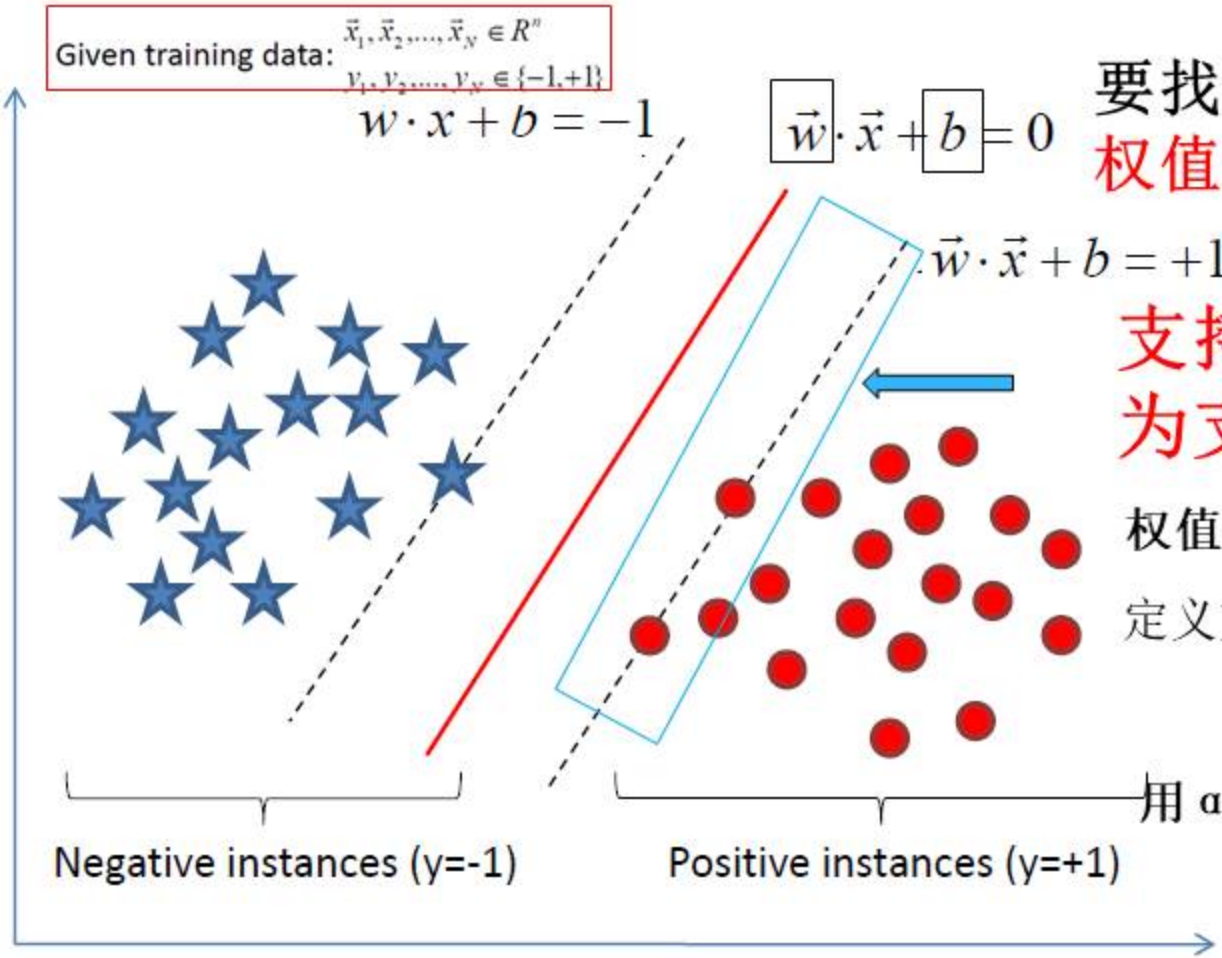
定义支持点为一个向量x_i

$w = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$

用$\alpha_1 x_1$表示数字系数和向量的乘积

支持面不仅跟支持点位置有关，还跟支持的类别y有关

2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

权值向量w和常数b由支持点来决定

定义支持点为一个向量x_i

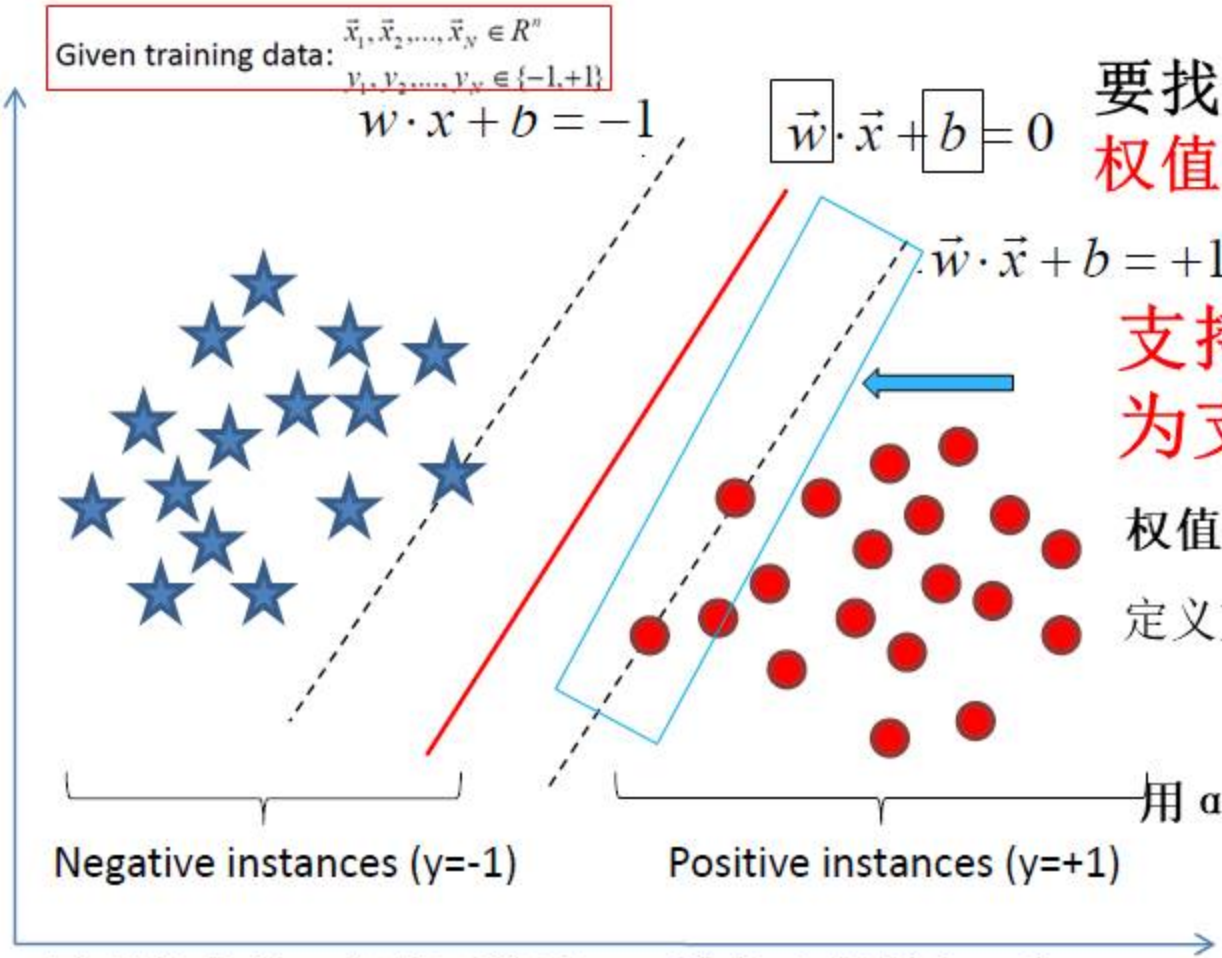
$w = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$

用$\alpha_1 x_1$表示数与系数和向量的乘积

支持面不仅跟支持点位置有关，还跟支持的类别y有关

$w = \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \cdots + \alpha_n y_n x_n$

2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

权值向量w和常数b由支持点来决定

定义支持点为一个向量x_i

$w = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$

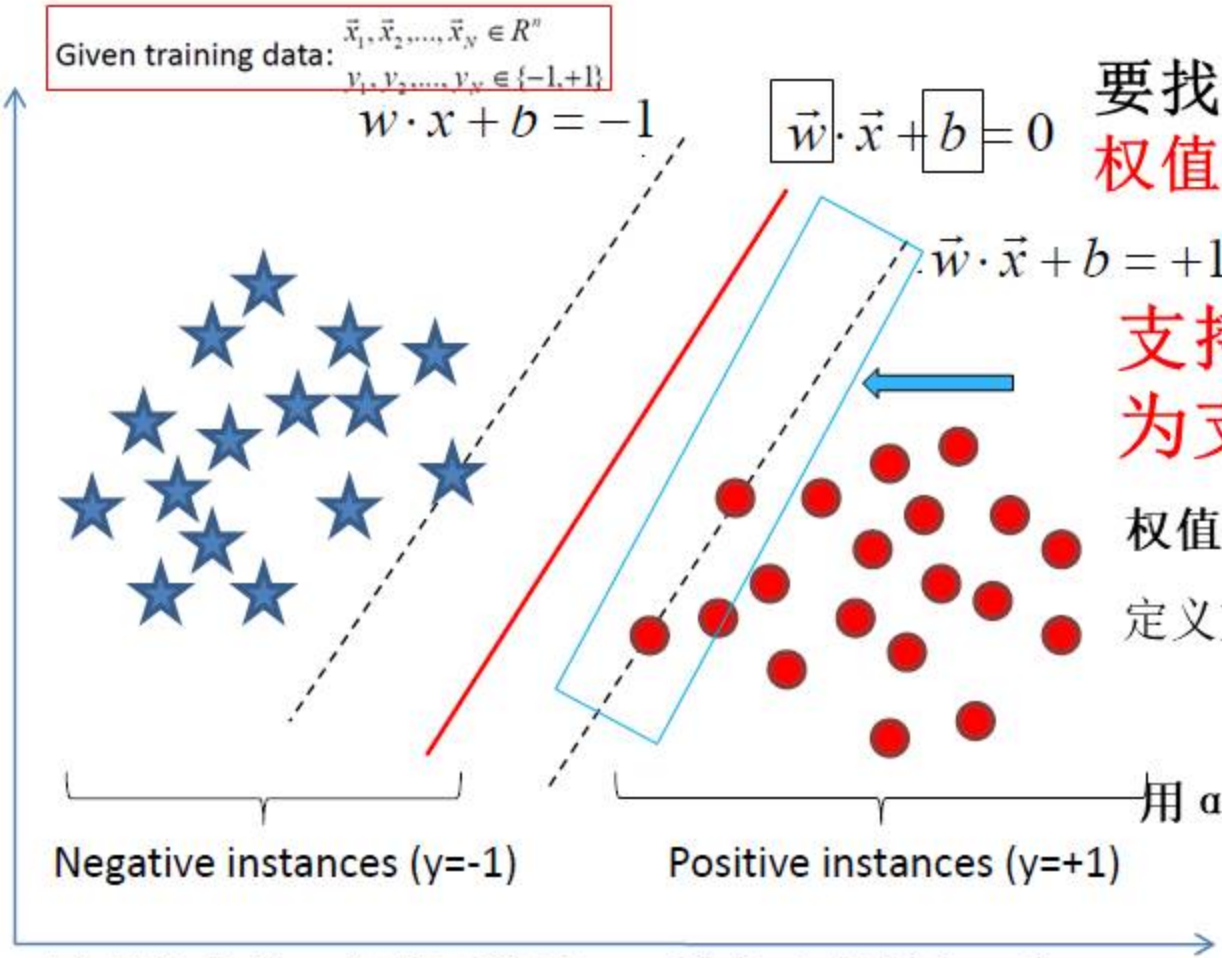
用$\alpha_1 x_1$表示数与系数和向量的乘积

支持面不仅跟支持点位置有关，还跟支持的类别y有关

$w = \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \cdots + \alpha_n y_n x_n$

只有很少的一部分不等于 0，这些点都落在H1和H2上，也正是这部分样本唯一的确定了分类函数，这些样本点称为支持（撑）向量。

2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

权值向量w和常数b由支持点来决定

定义支持点为一个向量x_i

$w = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$

用$\alpha_1 x_1$表示数与系数和向量的乘积

支持面不仅跟支持点位置有关，还跟支持的类别y有关

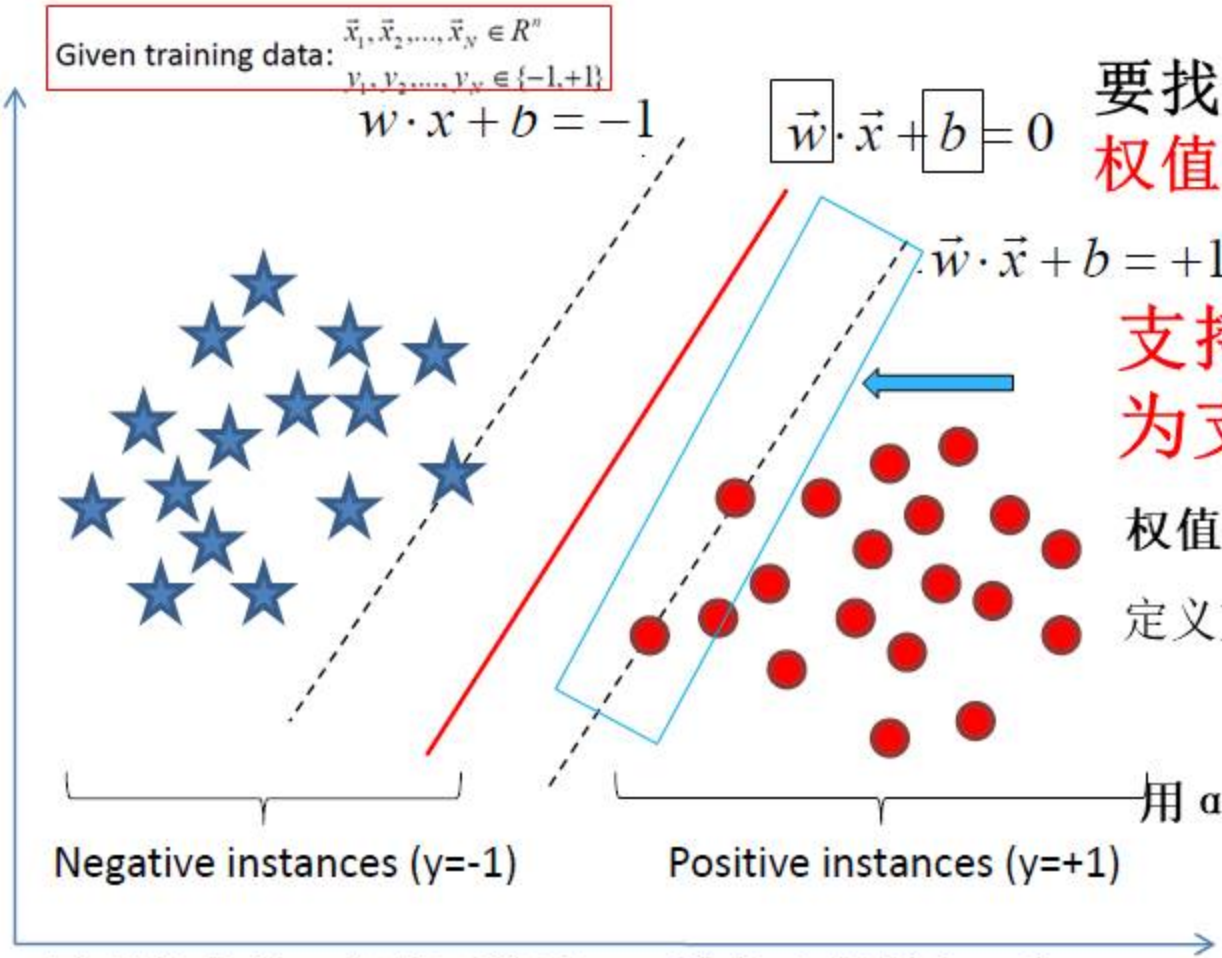
$w = \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \cdots + \alpha_n y_n x_n$

只有很少的一部分不等于 0，这些点都落在H1和H2上，也正是这部分样本唯一的确定了分类函数，这些样本点称为支持（撑）向量。

最小间隔面函数 $\vec{w} \cdot \vec{x} + b = 0 \Rightarrow$

$$g(x) = < w, x > + b = < \sum_{i=1}^{n} (\alpha_i y_i x_i), x > + b$$

2.1最小间隔面推导



要找最优分割面，只要确定权值向量w和常数b即可

支持面，上面的点被称为支持点

权值向量w和常数b由支持点来决定

定义支持点为一个向量x_i

$w = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n$

用$\alpha_1 x_1$表示数与系数和向量的乘积

支持面不仅跟支持点位置有关，还跟支持的类别y有关

$w = \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \cdots + \alpha_n y_n x_n$

只有很少的一部分不等于 0，这些点都落在H1和H2上，也正是这部分样本唯一的确定了分类函数，这些样本点称为支持（撑）向量。

最小间隔面函数 $\vec{w} \cdot \vec{x} + b = 0 \Rightarrow$

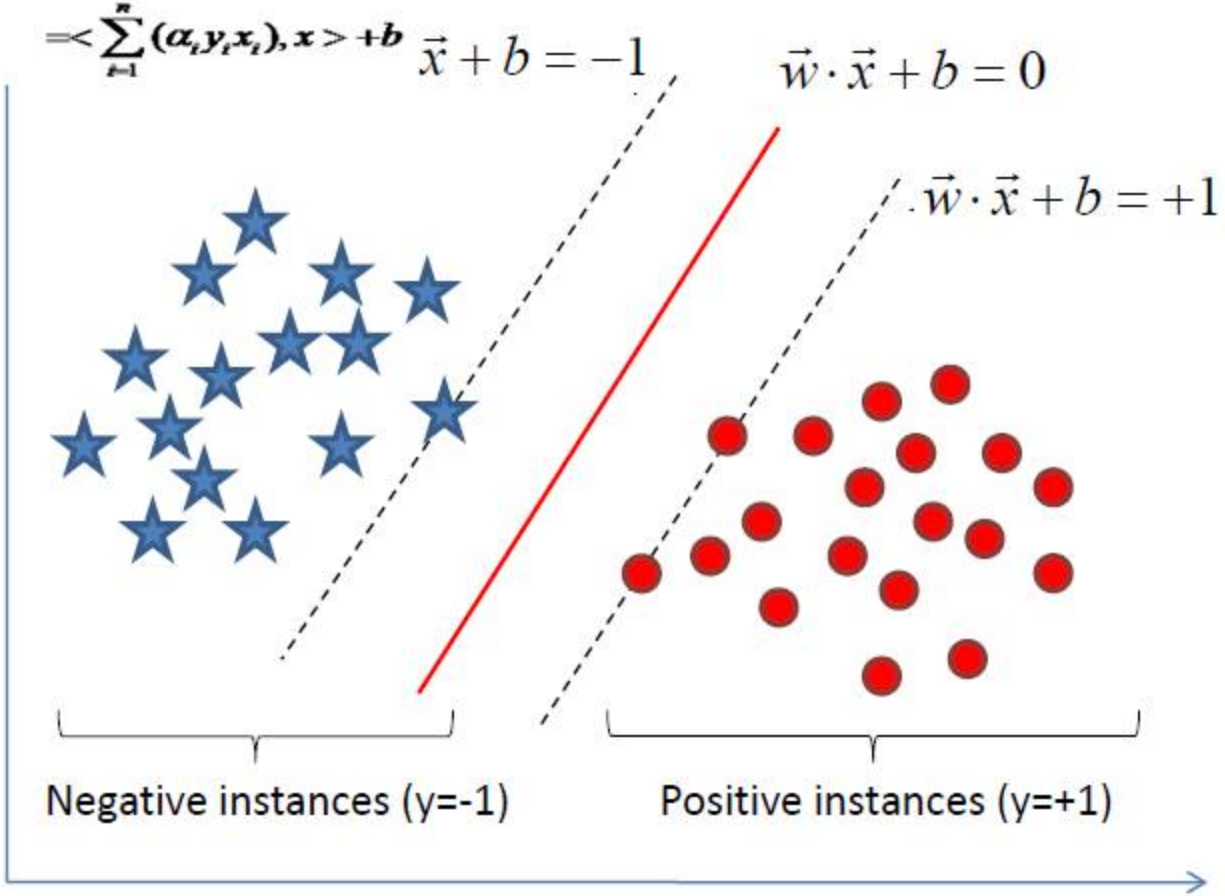
$$g(x) = < w, x > + b = < \sum_{i=1}^{n} (\alpha_i y_i x_i), x > + b$$

如何找支持点？

2.1最小间隔面推导

$g(x) = < w, x > + b$

$= < \sum_{i=1}^{n} (\alpha_i y_i x_i), x > + b$



应用几何知识，两根平行线的距离为常数项相减后除以法向量的模长

We know that

$$D = |b_1 - b_2| / \|\vec{w}\|$$

Therefore:

$$D = 2 / \|\vec{w}\|$$

最小化w跟最小化w平方是等价的，之所以这些做，是因为这样能够把问题变成二次规划问题，而二次规划问题是有通用的解法的

Since we want to maximize the gap,

we need to minimize $\|\vec{w}\|$

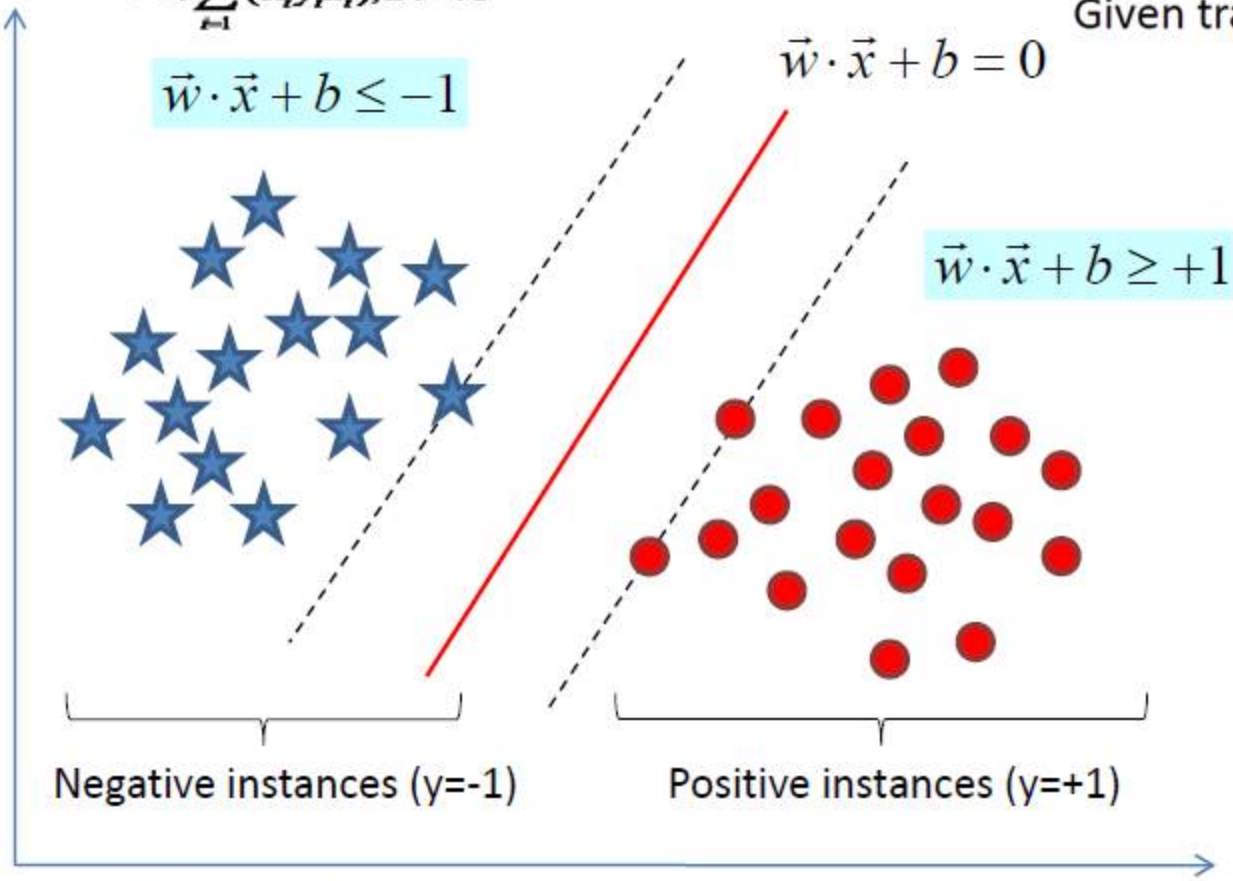
or equivalently minimize $\frac{1}{2}\|\vec{w}\|^2$

($\frac{1}{2}$ is convenient for taking derivative later on)

2.1最小间隔面推导

$$g(x) = < w, x > + b$$

$$= < \sum_{i=1}^{n} (\alpha_i y_i x_i), x > + b$$



Given training data: $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \in R^n$

$y_1, y_2, \ldots, y_N \in \{-1, +1\}$

Equivalently:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

In summary:

Want to minimize $\frac{1}{2}\|\vec{w}\|^2$ subject to $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$ for $i = 1, \ldots, N$

Then given a new instance x, the classifier is $f(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b)$

2.1最小间隔面推导

$$g(x) = < w, x > + b$$
$$= < \sum_{i=1}^{n} (\alpha_i y_i x_i), x > + b$$

"*primal formulation of linear SVMs*"

Minimize $\frac{1}{2}\sum_{i=1}^{n} w_i^2$ subject to $y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0$ for $i = 1,\ldots,N$

Objective function

Constraints

Gap(Margin): $D = 2/\|\vec{w}\|$

Problem Transformation:
max D -> min w -> min w^2 - > min 1/2(w^2)

2.1最小间隔面推导

$$g(x)=<w,x>+b$$
$$=<\sum_{i=1}^{n}(\alpha_i y_i x_i),x>+b$$

我们分类问题也被转化成一个带约束的最小值的问题：

$$\min \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad y_i[(wx_i)+b]-1\geq 0 \quad (i=1,2,\ldots,l)$$ (l是样本数)

在这个问题中，自变量就是w，而目标函数是w的二次函数，所有的约束条件都是w的线性函数这种规划问题有个很有名气的称呼——二次规划，而且可以更进一步的说，由于它的可行域是一个凸集，因此它是一个凸二次规划。凸二次规划让人喜欢的地方就在于，它有解，而且可以找到。

2.1最小间隔面推导

Minimize $\frac{1}{2}\sum_{i=1}^{n} w_i^2$ subject to $y_i(\vec{w}\cdot\vec{x}_i+b)-1\geq 0$ for $i = 1,\ldots,N$

Objective function　　Constraints

$$g(x)=<w,x>+b$$
$$=<\sum_{i=1}^{n}(\alpha_i y_i x_i),x>+b$$

- 用惩罚项来表达限制条件，从而能够转化带限制的优化问题为无限制的优化问题

$$\min_{w,b}\frac{1}{2}\|w\|^2+\text{penality term}$$

- 为训练数据中的每个数据样本，定义penality term如下：

$$\begin{Bmatrix} 0 & y_i(w^T x_i+b)\geq 1 \\ \infty & otherwise \end{Bmatrix} = \max_{\alpha_i\geq 0}\alpha_i(1-y_i(w^T x_i+b))$$

- 从而可以重写SVM的优化问题：

$$\min\{\frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\max_{\alpha_i\geq 0}\alpha_i(1-y_i(w^T x_i+b))\}$$

拉格朗日乘子

$$=\min_{w,b}\max_{\{\alpha_i\geq 0\}}\{\frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\alpha_i(1-y_i(w^T x_i+b))\}$$

2.1最小间隔面推导

- 通过交换“max”和“min”，形式化为对偶问题(1):

$$g(x)=<w,x>+b$$

$$\min_{w,b}\max_{\{\alpha_i\geq 0\}}\{\frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\alpha_i(1-y_i(w^Tx_i+b))\} \qquad =<\sum_{i=1}^{n}(\alpha_i y_i x_i),x>+b$$

$$=\max_{\{\alpha_i\geq 0\}}\ \min_{w,b}\underbrace{\{\frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\alpha_i(1-y_i(w^Tx_i+b))\}}_{L(w,b,\alpha)}$$

- 求解该对偶问题，在每一个固定的拉格朗日乘子 α_i 下，求解满足最小化 $J(w,b;\alpha)$ 的w和b

$$\frac{\partial L}{\partial \mathbf{w}}L(\mathbf{w},b,\alpha)=\mathbf{w}^\top-\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i^\top=0 \qquad (1)$$

$$\frac{\partial L}{\partial \mathbf{b}}L(\mathbf{w},b,\alpha)=-\sum_{i=1}^{n}\alpha_i y_i=0 \qquad (2)$$

2.1最小间隔面推导

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w}^{\top} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^{\top} = 0 \quad (1)$$

$$\frac{\partial L}{\partial \mathbf{b}} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (2)$$

$$g(x) = < w, x > + b$$

支持点理论解释

$$= < \sum_{i=1}^{n} (\alpha_i y_i x_i), x > + b$$

- 基于上式（1）可以得到 $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

2.1最小间隔面推导

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w}^{\top} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^{\top} = 0 \quad (1)$$

$$\frac{\partial L}{\partial \mathbf{b}} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (2)$$

$$g(x) = < w, x > + b$$

支持点理论解释

$$= < \sum_{i=1}^{n} (\alpha_i y_i x_i), x > + b$$

- 基于上式（1）可以得到 $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

- 所以 $L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)$

$$\min_{w,b} \max_{\{\alpha_i \geq 0\}} \{ \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b)) \}$$

$$= \max_{\{\alpha_i \geq 0\}} \underbrace{\min_{w,b} \{ \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b)) \}}_{L(w,b,\alpha)}$$

2.1最小间隔面推导

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w}^{\top} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^{\top} = 0 \quad (1)$$

$$\frac{\partial L}{\partial \mathbf{b}} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (2)$$

$$g(x) = <w, x> + b$$

支持点理论解释

$$= <\sum_{i=1}^{n} (\alpha_i y_i x_i), x> + b$$

- 基于上式（1）可以得到 $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

- 所以 $L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)$

$$\min_{w,b} \max_{\{\alpha_i \geq 0\}} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}$$

$$= \max_{\{\alpha_i \geq 0\}} \underbrace{\min_{w,b} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}}_{L(w,b,\alpha)}$$

- 原始的优化问题（1）变成如下问题（2）：

$$\max_{\{\alpha_i \geq 0\}} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)\}$$

$$s.t.\ \alpha_i \geq 0, \forall i,$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\min_{w,b} \max_{\{\alpha_i \geq 0\}} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}$$

$$= \max_{\{\alpha_i \geq 0\}} \underbrace{\min_{w,b} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}}_{L(w,b,\alpha)}$$

2.1最小间隔面推导

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w}^{\top} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^{\top} = 0 \quad (1)$$

$$\frac{\partial L}{\partial \mathbf{b}} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (2)$$

$$g(x) = <w, x> + b$$

支持点理论解释

$$= <\sum_{i=1}^{n} (\alpha_i y_i x_i), x> + b$$

- 基于上式（1）可以得到 $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

- 所以 $L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)$

$$\min_{w,b} \max_{\{\alpha_i \geq 0\}} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}$$

$$= \max_{\{\alpha_i \geq 0\}} \underbrace{\min_{w,b} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}}_{L(w,b,\alpha)}$$

- 原始的优化问题（1）变成如下问题（2）：

$$\max_{\{\alpha_i \geq 0\}} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)\}$$

$$s.t.\ \alpha_i \geq 0, \forall i,$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\min_{w,b} \max_{\{\alpha_i \geq 0\}} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}$$

$$= \max_{\{\alpha_i \geq 0\}} \underbrace{\min_{w,b} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}}_{L(w,b,\alpha)}$$

●仍然是一个二次规划问题，可以求解到全局最优解。（方法：SMO）

2.1最小间隔面推导

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w}^{\top} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^{\top} = 0 \quad (1)$$

$$\frac{\partial L}{\partial \mathbf{b}} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (2)$$

$$g(x) = <w, x> + b$$

支持点理论解释

$$= <\sum_{i=1}^{n} (\alpha_i y_i x_i), x> + b$$

- 基于上式（1）可以得到 $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

- 所以 $L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)$

$$\min_{w,b} \max_{\{\alpha_i \geq 0\}} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}$$

$$= \max_{\{\alpha_i \geq 0\}} \underbrace{\min_{w,b} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}}_{L(w,b,\alpha)}$$

- 原始的优化问题（1）变成如下问题（2）：

$$\max_{\{\alpha_i \geq 0\}} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)\}$$

$$s.t.\ \alpha_i \geq 0, \forall i,$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\min_{w,b} \max_{\{\alpha_i \geq 0\}} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}$$

$$= \max_{\{\alpha_i \geq 0\}} \underbrace{\min_{w,b} \{\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b))\}}_{L(w,b,\alpha)}$$

- 得到问题的解 $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

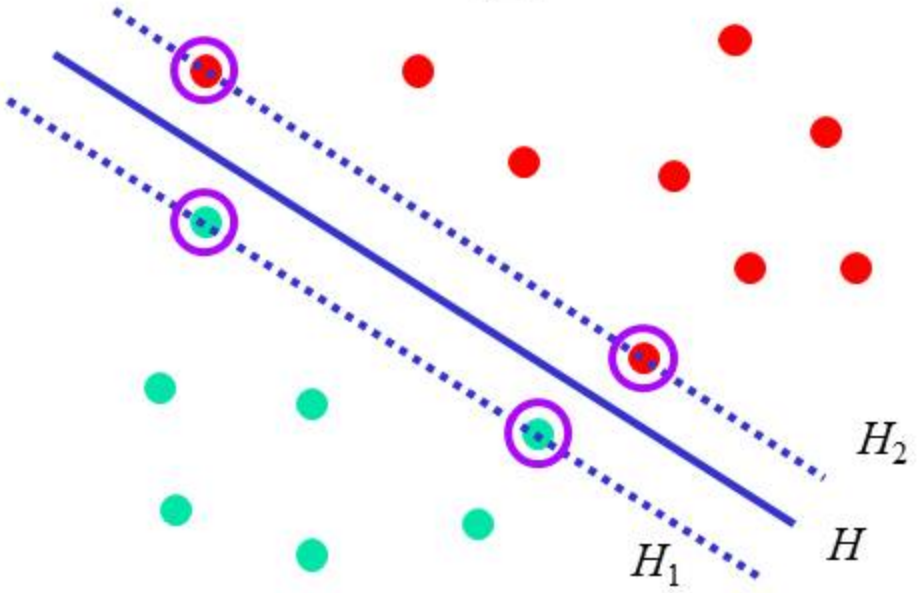
$$b = y_i - \mathbf{w}^{\top} \mathbf{x}_i$$

- 仍然是一个二次规划问题，可以求解到全局最优解。（方法：SMO）
- 一般来说，解仅含有部分非零
- 所谓支持向量，即对应非零向量

2.2 SVM目标函数求解：对偶问题求解

- 发现
  - 所有非支持向量对应的系数α都等于0

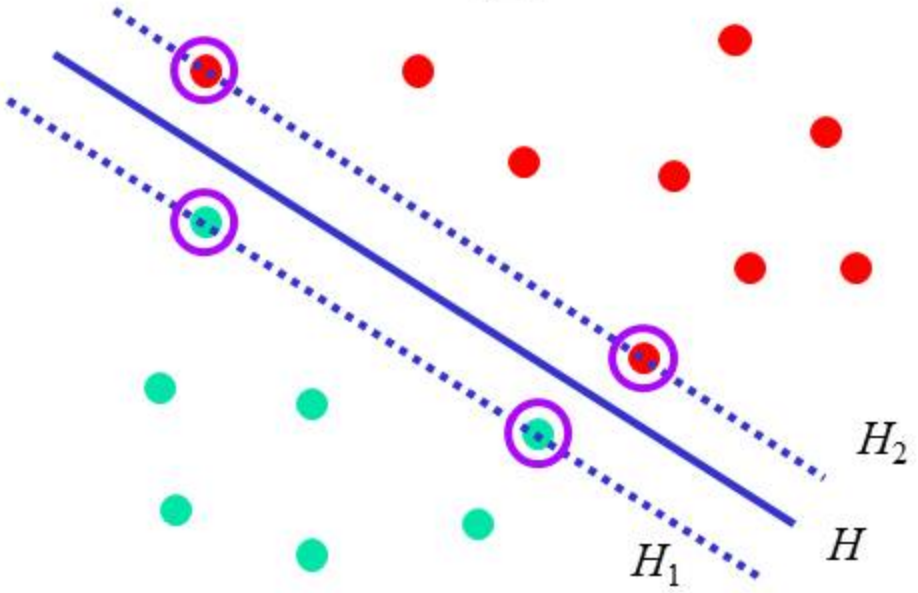
$$\mathcal{L}(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i(w^Tx_i+b)-1)$$



2.2 SVM目标函数求解：对偶问题求解

- 发现
  - 所有非支持向量对应的系数α都等于0

$$\mathcal{L}(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i(w^Tx_i+b)-1)$$



支持向量机解的稀疏性：训练完成后，大部分的训练样本都不需保留，最终模型仅与支持向量有关.

2.2 SVM目标函数求解：稀疏性理论解释

- 最终模型：

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\top}\boldsymbol{x} + b = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i^{\top}\boldsymbol{x} + b$$

- KKT条件：

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\boldsymbol{x}_i) \geq 1, \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0. \end{cases}$$

$$y_i f(\boldsymbol{x}_i) > 1 \quad \Rightarrow \quad \alpha_i = 0$$

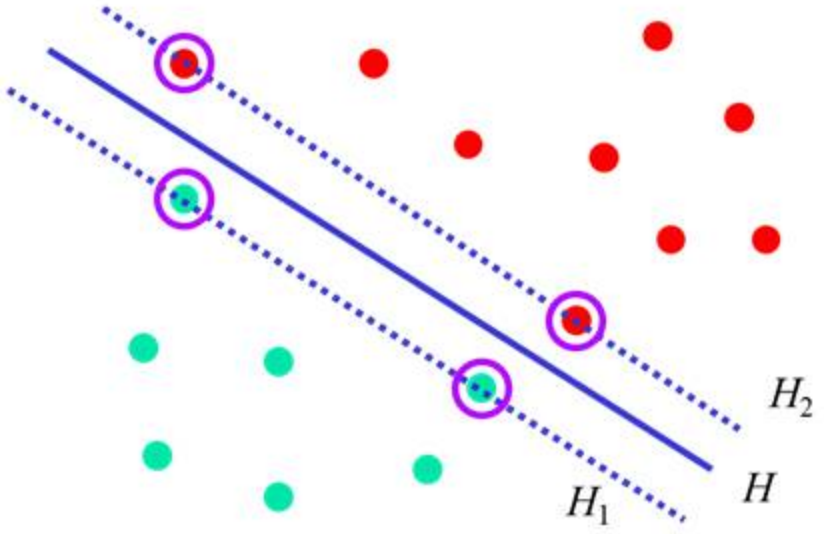
支持向量机解的稀疏性：训练完成后，大部分的训练样本都不需保留，最终模型仅与支持向量有关.

设置

Svm分类器中，若支持向量太多，说明训练的模型

A 过拟合

B 欠拟合



$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x} + b$$

提交

2.3 SVM目标函数求解：对偶问题求解

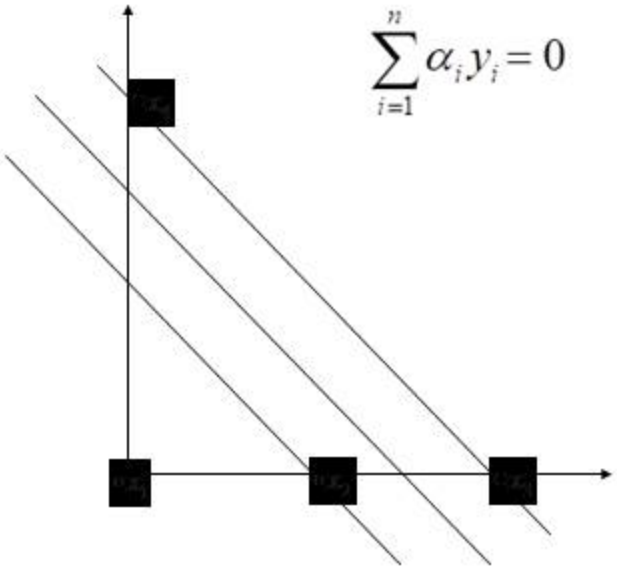
- x的分类
  - 预测x的分类时，就是将x代入到 $f(x) = w^T x + b$ 中，算出结果，根据其正负号判断其类别
  - 根据前面推导 $$w^* = \sum_{i=1}^{n} \alpha_i y_i x_i$$
  - 可得分类函数为 $$f(x) = (\sum_{i=1}^{n} \alpha_i y_i x_i)^T x + b$$ $$= \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + b$$
  - 可见新点x的类别预测，实际上只需要计算x与训练数据点的内积即可

2.4 一个例子

$$\max_{\{\alpha_i \geq 0\}} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)\}$$

$$s.t.\ \alpha_i \geq 0, \forall i,$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$



$\mathbf{x_1}$ **=(0, 0),** $\mathbf{y_1}$ **= +1**

$\mathbf{x_2}$ **=(1, 0),** $\mathbf{y_2}$ **= +1**

$\mathbf{x_3}$ **=(2, 0),** $\mathbf{y_3}$ **= -1**

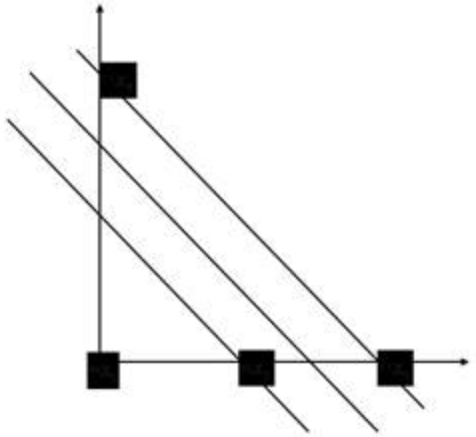
$\mathbf{x_4}$ **=(0, 2),** $\mathbf{y_4}$ **= -1**

$$Q(\alpha) = (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2}(\alpha_2^2 - 4\alpha_2\alpha_3 + 4\alpha_3^2 + 4\alpha_4^2)$$

可调用Matlab中的二次规划程序，求得α_1，α_2，α_3，α_4的值，进而求得w和b的值。

2.4 一个例子

$$\begin{cases} \alpha_1 = 0 \\ \alpha_2 = 1 \\ \alpha_3 = 3/4 \\ \alpha_4 = 1/4 \end{cases}$$



$$w = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{3}{4}\begin{bmatrix} 2 \\ 0 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

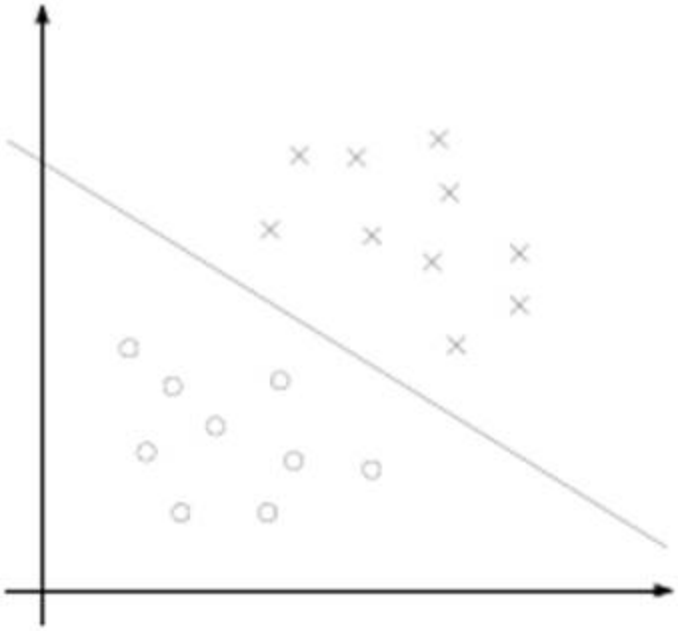
$$b = -\frac{1}{2}\left[-\frac{1}{2}, -\frac{1}{2}\right]\begin{bmatrix} 3 \\ 0 \end{bmatrix} = \frac{3}{4}$$

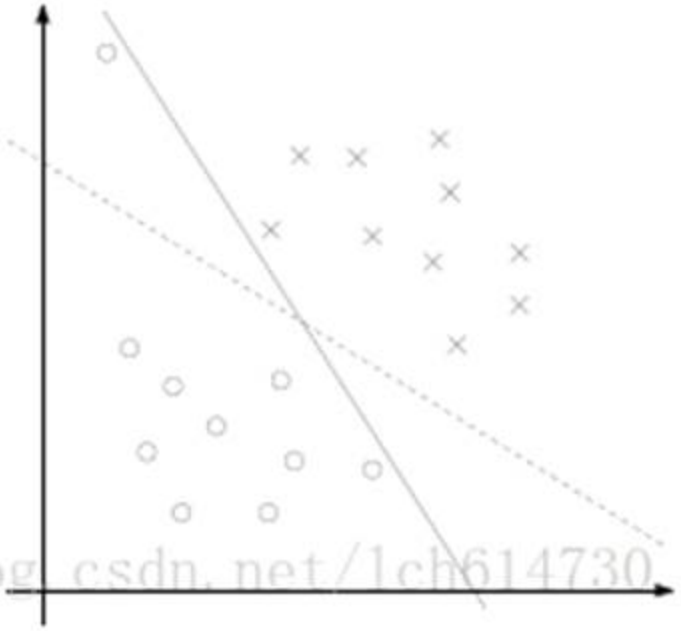
$$g(x) = 3 - 2x_1 - 2x_2 = 0$$

主要内容

- 1.了解SVM
- 2. 深入SVM
- 3. 非线性SVM

3.1基于软裕量的C-SVM

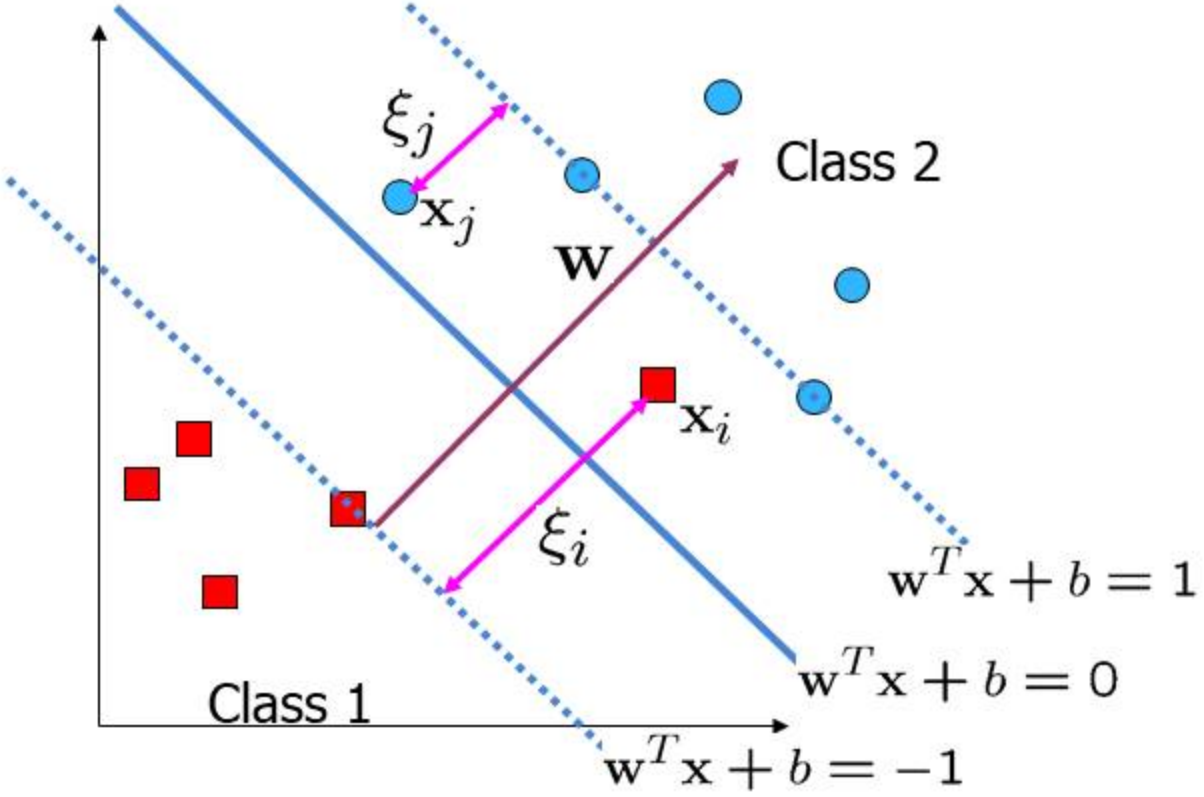




3.1基于软裕量的C-SVM

- 概述:
  - 经典SVM的基本假设是样本之间线性可分。但这在实践中常常并不合理——线性不可分是更普遍的现象。
  - 为了解决线性不可分的情况，Cortes和Vapnik于1995年提出通过修改和扩展上述裕量最大化问题来予以解决。
  - 这种方法的核心思想是：不再像经典SVM那样要求所有的训练样本均能被正确划分，而是允许一定数量的训练样本被分错，也即训练过程容忍一定程度的分类误差。这样，训练过程挑选的是一个尽可能能够正确划分训练样本的超平面，而训练的目标仍然是最大化该超平面与分类正确的、最近的正/负例训练样本之间的距离（仍然称之为裕量——软裕量）。

3.1基于软裕量的C-SVM



3.1基于软裕量的C-SVM

- 形式化:
  - 具体的，这种所谓软裕量方法引入了松弛变量$\xi_i(\geq 0)$，用以表征或者说度量分类器针对训练样本$\mathbf{x}_i$的差错程度。则最优划分平面求解的优化问题转化为:

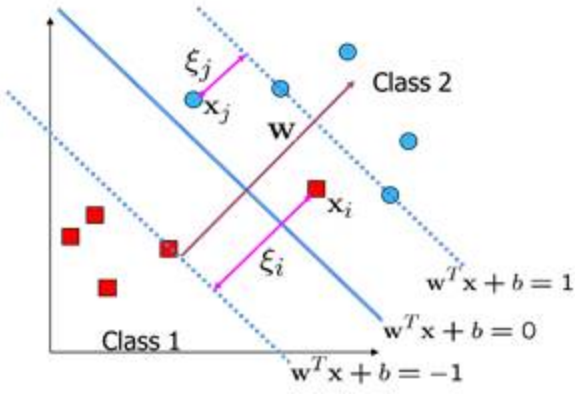
$$\min_{(\mathrm{w},b)}\left\{\frac{1}{2}\|\mathrm{w}\|^2+C\sum_{i=1}^{n}\xi_i\right\}$$

$$s.t.\quad \left(y_i\left(\mathrm{w}\mathrm{x}_i+b\right)\geq 1-\xi_i\right)\mathrm{and}\left(\xi_i\geq 0\right),\quad \forall i=1,\ldots,n$$

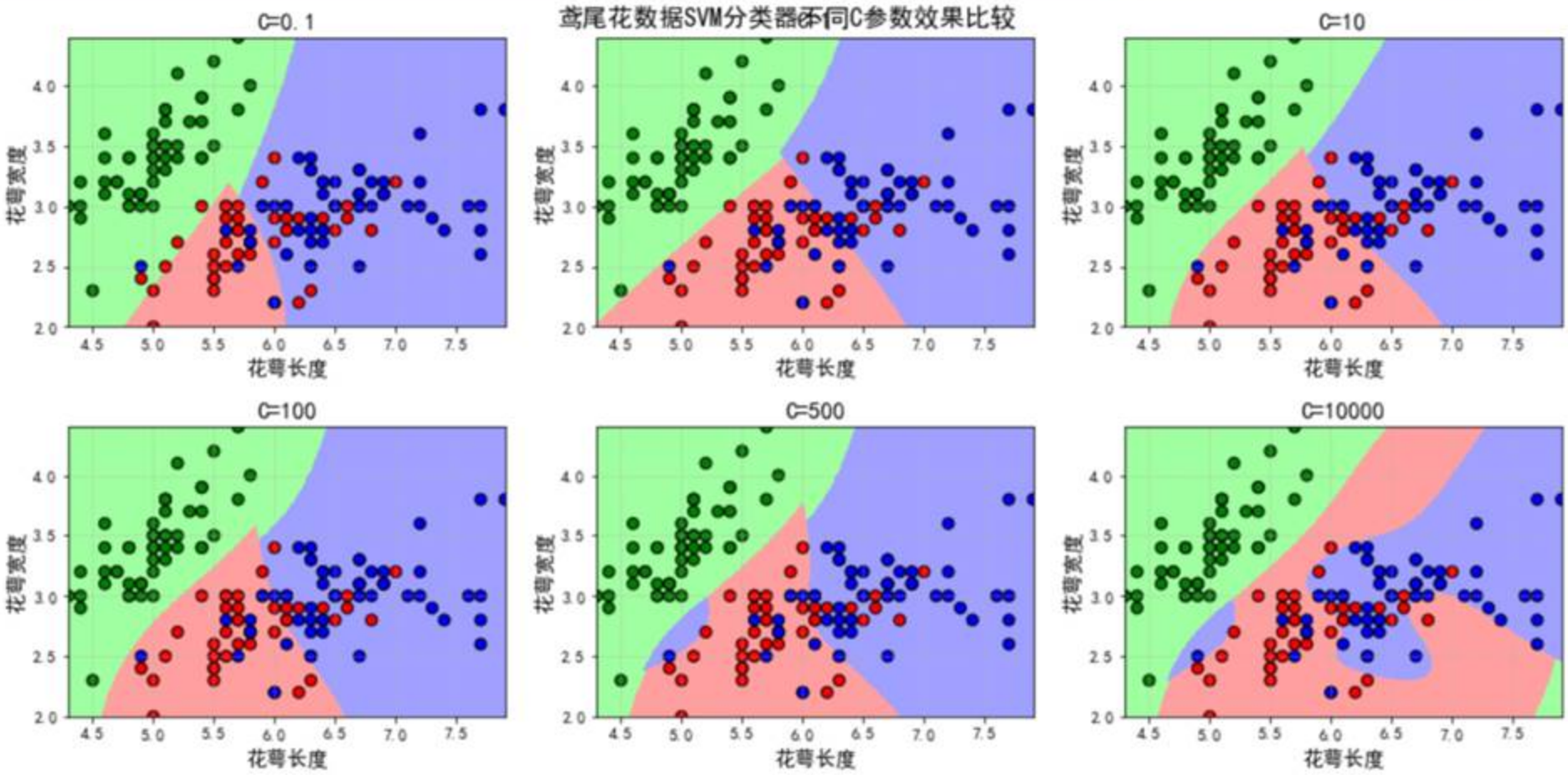
其中C为预先选定的调和参数。

  - 同样运用Lagrange数乘法，通过引入非负的Lagrange乘子矢量$\boldsymbol{\alpha}$和$\boldsymbol{\beta}$，可得到此时优化问题的原始形式如下：

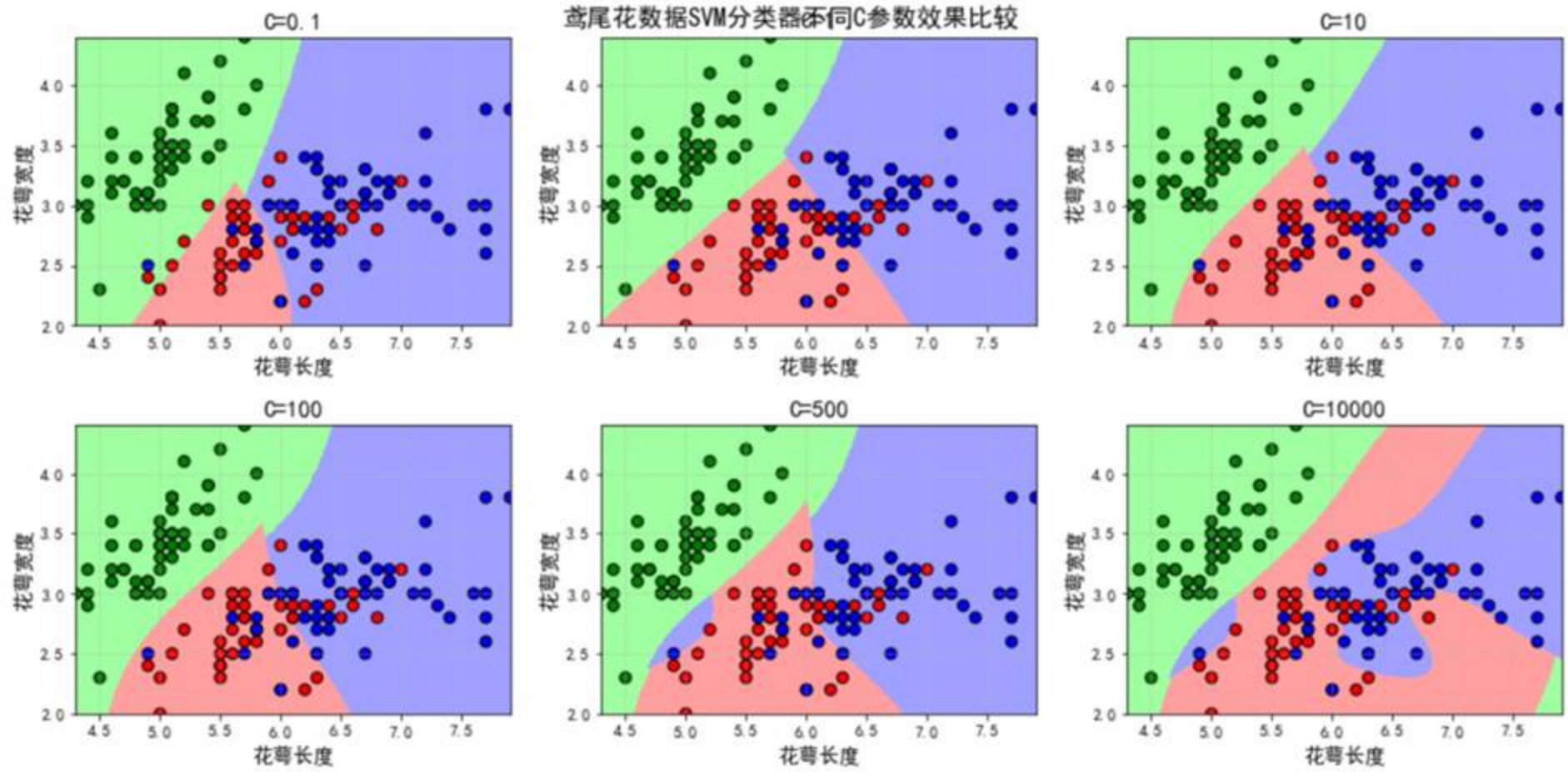
$$\min_{(\mathrm{w},b)}\max_{\boldsymbol{\alpha},\,\boldsymbol{\beta}}\left\{\frac{1}{2}\|\mathrm{w}\|^2+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i\left[y_i(\mathrm{w}\cdot\mathrm{x}_i+b)-1+\xi_i\right]-\sum_{i=1}^{n}\beta_i\xi_i\right\}$$



3.1不同的C的影响


鸢尾花数据SVM分类器不同C参数效果比较
C=0.1
C=10
C=100
C=500
C=10000
花萼宽度
花萼长度
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

3.1不同的C的影响



$$\min_{(\mathrm{w},b)}\left\{\frac{1}{2}\|\mathrm{w}\|^2+C\sum_{i=1}^{n}\xi_i\right\}$$

$$s.t.\quad \left(y_i\left(\mathrm{w}x_i+b\right)\geq 1-\xi_i\right)\mathrm{and}\left(\xi_i\geq 0\right),\quad \forall i=1,\ldots,n$$

C值太小时，使得松弛变量过大，**导致样本误分较多，但模型泛化能力强**

C值太大时，使得松弛变量过小，**导致过分拟合训练数据，导致过拟合**

单选题 1分

设置

基于软裕量的C-SVM，如果c过大，将会

A 松弛变量过大，导致样本误分较多，但模型泛化能力强

B 松弛变量过小，导致过分拟合训练数据，导致过拟合

$$\min_{(w,b)}\left\{\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i\right\}$$

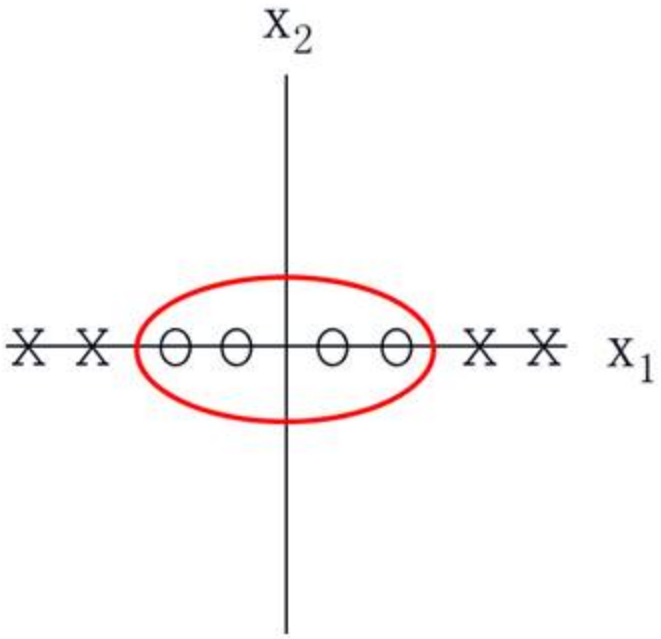
$$s.t.\quad \left(y_i\left(wx_i+b\right)\geq 1-\xi_i\right)\text{and}\left(\xi_i\geq 0\right),\quad \forall i=1,\ldots,n$$

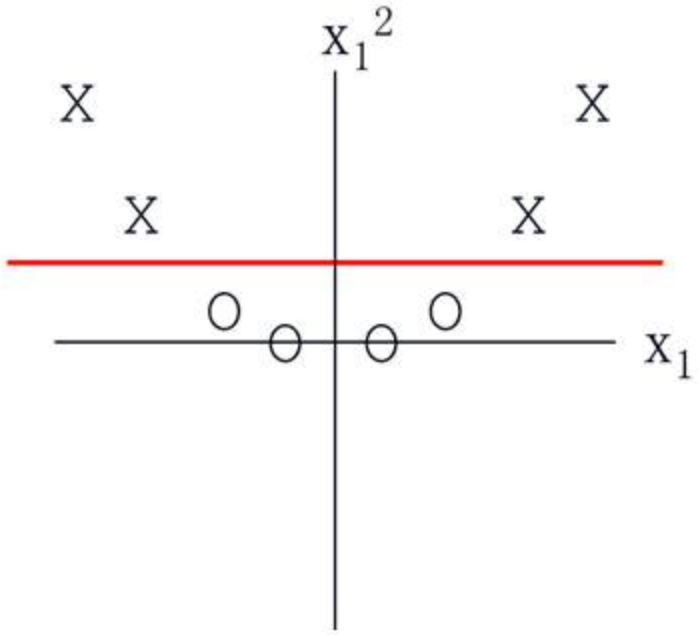
提交

3.2非线性SVM与核变换

- 概述:
  - 上面讨论的都是训练样本(大致)线性可分的情形，这时分类器为线性函数，即分类超平面。
  - 但现实情况中（见下面两图所示的情形），训练样本往往并非线性可分的，也即任何超平面都无法较好的分开两类训练样本，或者说使用任何超平面带来的、对训练样本的分类误差都是不可容忍的。

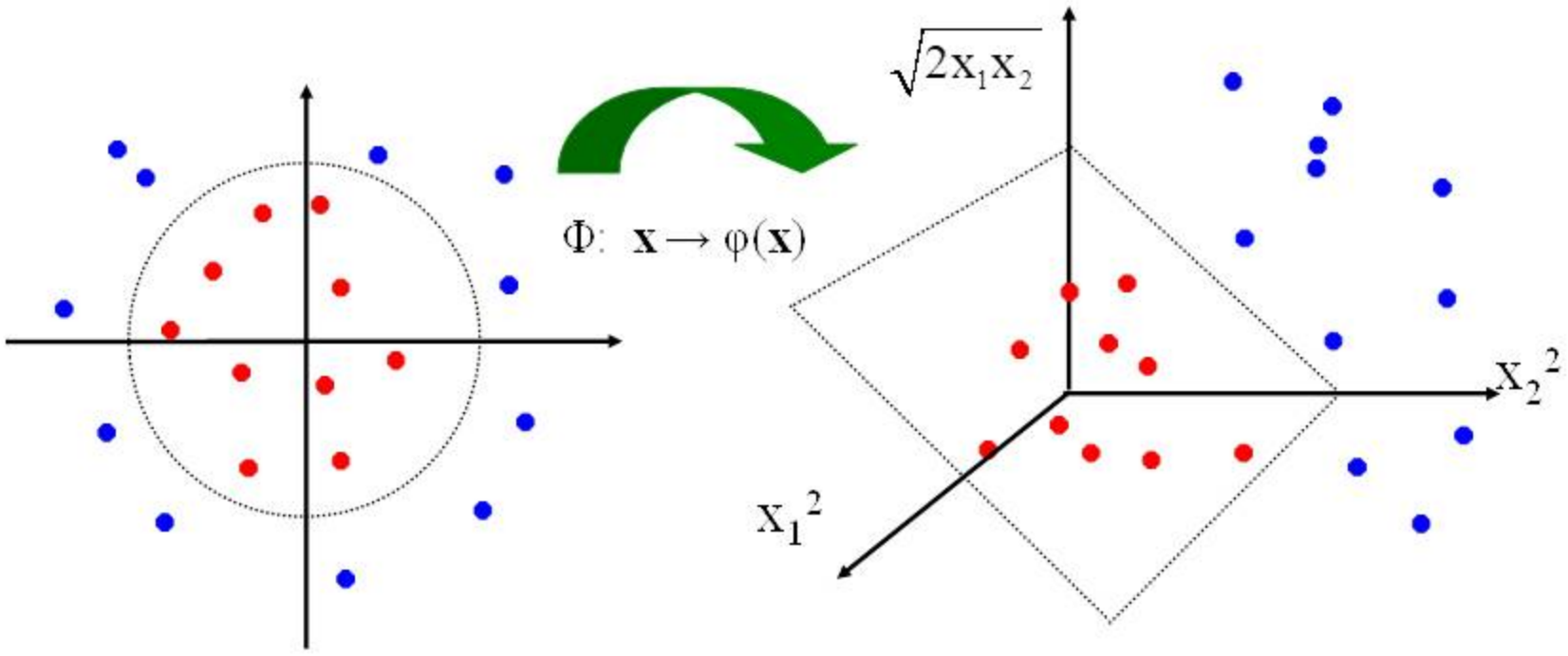
3.2非线性SVM与核变换





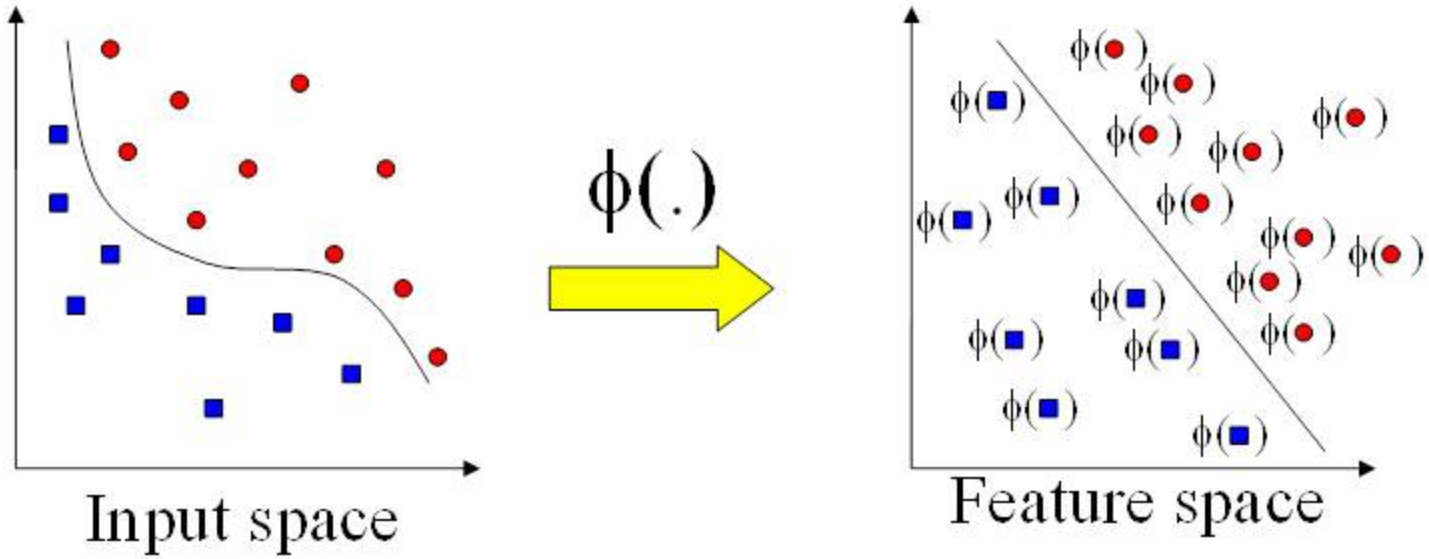
3.2非线性SVM与核变换

- 概述:



3.2非线性SVM与核变换

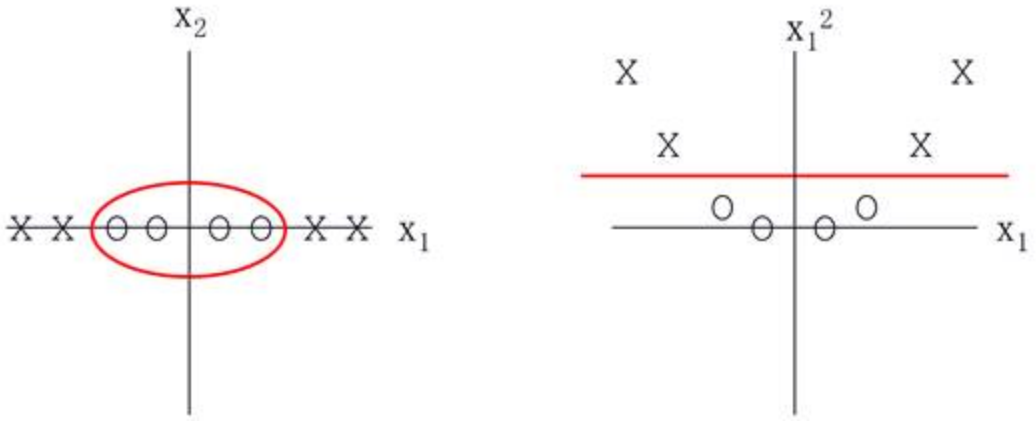
- 概述:



3.2非线性SVM与核变换

- **概述:**
  - **由于特征维度的提高一般总是能提升样本之间的可区分性，所以可以考虑将原始样本特征描述映射至某个高维空间中，使得映射后的样本之间线性可分。**
  - 形式化的，记此映射为 $\Phi: \mathrm{x} \to \varphi(\mathrm{x})$，其中 $\mathrm{x} \in R^p$，而 $z=\varphi(\mathrm{x}) \in R^q$，且通常有 $q \gg p$。这样，原始的训练样本集合**D**被映射为线性可分的高维空间中的集合：

$$\mathrm{D}' = \left\{ (\mathrm{z}_i, y_i) \mid \mathrm{z}_i \in R^q, y_i \in \{-1, 1\} \right\}_{i=1}^{n}$$



3.2非线性SVM与核变换

- **概述:**
  - **在R^q空间中，因为D'是线性可分的，所以其判别函数和最优分类平面求解的对偶形式分别为:**

$$f(\mathrm{z}) = \mathrm{sign}(\mathrm{w}\mathrm{z} + b - 1) = \mathrm{sign}(\sum_{i=1}^{n} \alpha_i y_i \mathrm{z}_i^T \mathrm{z} + b - 1)$$

$$\max_{\alpha \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathrm{z}_i^T \mathrm{z}_j \right\}$$

  - 观察以上两个式子可见：无论判别函数还是对偶形式中的目标函数都只涉及到高维空间中两个矢量之间的内积，而并不需要知道它们的具体坐标。 $\mathrm{z}_i^T \mathrm{z}_j = K(\mathrm{x}_i, \mathrm{x}_j)$

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

3.2非线性SVM与核变换

- **常用核函数:**
  - **高斯RBF (Radial Basis Function)核:**

$$K(\mathrm{x}_i,\mathrm{x}_j)=\exp\left(-\gamma\left\|\mathrm{x}_i-\mathrm{x}_j\right\|^2\right)=\exp\left(-\left\|\mathrm{x}_i-\mathrm{x}_j\right\|^2/2\sigma^2\right)$$

  - 齐次多项式(homogeneous polynomial)核:

$$K(\mathrm{x}_i,\mathrm{x}_j)=\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)^d$$

  - 非齐次多项式(inhomogeneous polynomial)核:

$$K(\mathrm{x}_i,\mathrm{x}_j)=\left(\mathbf{x}_i\cdot\mathbf{x}_j+1\right)^d$$

  - Sigmoid核:

$$K(\mathrm{x}_i,\mathrm{x}_j)=\tanh\left(\kappa\mathbf{x}_i\cdot\mathbf{x}_j+\theta\right)^d$$

3.3多分类问题

- 如何将SVM的二分类转换成多分类问题?
  - 一对一
    - 对N 类训练数据两两组合,构建$C^2_N = N(N-1)/2$个支持向量机。最后分类的时候采取“投票”的方式决定分类结果。
  - 一对其余
    - 对N分类问题构建N个支持向量机,每个支持向量机负责区分本类数据和非本类数据。最后结果由输出离分界面距离$w \cdot x + b$最大的那个支持向量机决定。

单选题 1分

设置

svm是否对噪音点敏感

A 是

B 否

提交

svm编程实践

- https://scikit-learn.org/stable/modules/svm.html#svm-classification
- 同学们可以尝试利用python读入iris，来完成svm，分析其分类效果

第12次课后作业

- 第十二次课后作业-在educoder平台上完成作业
- https://www.educoder.net/shixuns/bfyloih4/challenges
- https://www.educoder.net/shixuns/m63hopav/challenges
- https://www.educoder.net/shixuns/b6yi97f2/challenges
- https://www.educoder.net/shixuns/ya8h7utx/challenges

提交作业截至时间：2020年3月29日