统计学习方法

第7章 支持向量机

同济大学软件学院
章世杰
2019.01

第7章 支持向量机

原始问题:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i,$$

$$s.t.\ \ y_i(w\cdot x_i+b)\geq 1-\xi_i,\ \ i=1,2,\cdots,N,$$

$$\xi_i\geq 0,\ \ i=1,2,\cdots,N.$$

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

输出: 最大间隔分离超平面$w^*\cdot x+b^*=0$和分类决策函数

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

对偶问题:

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

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

$$0 \le \alpha_i \le C, \ \ i = 1,2,\cdots,N.$$

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

输出: 最优解$\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*)^T$, 选择$\alpha^*$的一个正分量$0 < \alpha_j^* < C$, 得到最大间隔分离超平面

$\sum_{i=1}^{N}\alpha_i^* y_i K(x \cdot x_i) + y_j - \sum_{i=1}^{N}\alpha_i^* y_i K(x_i, x_j) = 0$

和分类决策函数

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

7.1 线性可分支持向量机（硬间隔最大化）

- **线性可分支持向量机的定义**

给定的线性可分训练数据集, 通过间隔最大化或等价地求解相应的凸二次规划问题学习得到的分离超平面为

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

以及相应的分类决策函数

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

称为线性可分支持向量机.

7.1 线性可分支持向量机（硬间隔最大化）

- **函数间隔的定义**

对于给定的训练数据集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

$$\widehat{\gamma_i} = y_i(w \cdot x_i + b).$$

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

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

7.1 线性可分支持向量机（硬间隔最大化）

- **几何间隔的定义**

对于给定的训练数据集T和超平面(w, b), 定义超平面(w, b)关于样本点(x_i, y_i)的几何间隔为

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

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

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

7.1 线性可分支持向量机（硬间隔最大化）

- **最大化间隔原理**

支持向量机学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面. 对线性可分的训练数据集而言, 线性可分分离超平面有无穷多个(等价于感知机), 但是几何间隔最大的分离超平面是唯一的. 这里的间隔最大化又称为硬间隔最大化.

间隔最大化的直观解释是: 对训练数据集找到几何间隔最大的超平面意味着以充分大的确信度对训练数据进行分类, 也就是说, 不仅将正负实例点分开, 而且对最难分的实例点(离超平面最近的点)也有足够大的确信度将它们分开.

7.1 线性可分支持向量机（硬间隔最大化）

- **最大化间隔的数学表示**

数学形式, 即

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

$$s.t.\ \ y_i(w \cdot x_i + b) \geq \hat{\gamma},\ \ i = 1,2,\cdots,N.$$

不妨令$\hat{\gamma} = 1$, 又注意到最大化$\frac{1}{||w||}$和最小化$\frac{1}{2}||w||^2$是等价的, 于是得到下面的线性可分支持向量机学习的最优化问题

$$\min_{w,b}\ \frac{1}{2}||w||^2,$$

$$s.t.\ \ y_i(w \cdot x_i + b) \geq 1,\ \ i = 1,2,\cdots,N.$$

7.1 线性可分支持向量机（硬间隔最大化）

- **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, $x_i \in X = R^n$, $y_i \in Y = \{-1, +1\}$, $i = 1,2, \cdots, N$;

输出: 最大间隔分离超平面$w^* \cdot x + b^* = 0$和分类决策函数$f(x) = sign(w^* \cdot x + b^*)$.

(1)构造并求解约束最优化问题:

$$\min_{w,b} \ \frac{1}{2} ||w||^2,$$

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

求得最优解w^*, b^*.

(2)由此得到分离超平面:

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

和分类决策函数

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

7.1 线性可分支持向量机（硬间隔最大化）

- **最大间隔分离超平面的存在唯一性**

若训练数据集T线性可分, 则可将训练数据集中的样本完全正确分开的最大间隔分离超平面存在且唯一.
证明: (1)存在性由训练数据集线性可分, 且目标函数非负, 即有下界, 可得.

(2)下面证明必要性. 反正而设存在两个最优解(w_1^*, b_1^*)和(w_2^*, b_2^*). 其中, $||w_1^*|| = ||w_2^*||$是显然的.

取$w = \frac{w_1^*+w_2^*}{2}$进行不等式夹逼, 即得$w_1^*$和$w_2^*$线性相关. 结合$||w_1^*|| = ||w_2^*||$知, $w_1^* = \pm w_2^*$. 注意到, 训练数据集中既有正类点又有负类点所以最优解(w^*, b^*)必满足$w^* \neq 0$. 从而, $w_1^* \neq -w_2^*$, 否则, $w = 0$. 进而, $w_1^* = w_2^*$. 由此可以把两个最优解(w_1^*, b_1^*)和(w_2^*, b_2^*)分别写成 (w^*, b_1^*)和(w^*, b_2^*).
下面只需证明$b_1^* = b_2^*$. 令x_1', x_2'是集合$\{x_i | y_i = +1\}$中分别对应于(w^*, b_1^*)和(w^*, b_2^*)使得问题的不等式等号成立的点. 由定义可得, $w^* x_1' + b_1^* \leq w^* x_2' + b_1^*$, $w^* x_2' + b_2^* \leq w^* x_1' + b_2^*$, 且$w^* x_1' + b_1^* = 1 = w^* x_2' + b_2^*$. 所以, b_1^*-$b_2^* = w^*(x_1' - x_2') = 0$, 即$b_1^* = b_2^*$. 由$w_1^* = w_2^*$和$b_1^* = b_2^*$可知, 两个最优解$(w_1^*, b_1^*)$和$(w_2^*, b_2^*)$是相同的, 解的唯一性得证.

7.1 线性可分支持向量机（硬间隔最大化）

- **支持向量和间隔边界**

在线性可分情况下，训练数据集的样本点中与分离超平面距离最近的样本点的实例称为支持向量，支持向量是使约束条件式等号成立的点，即

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

对$y_i = \pm 1$的正(负)例点，支持向量在超平面

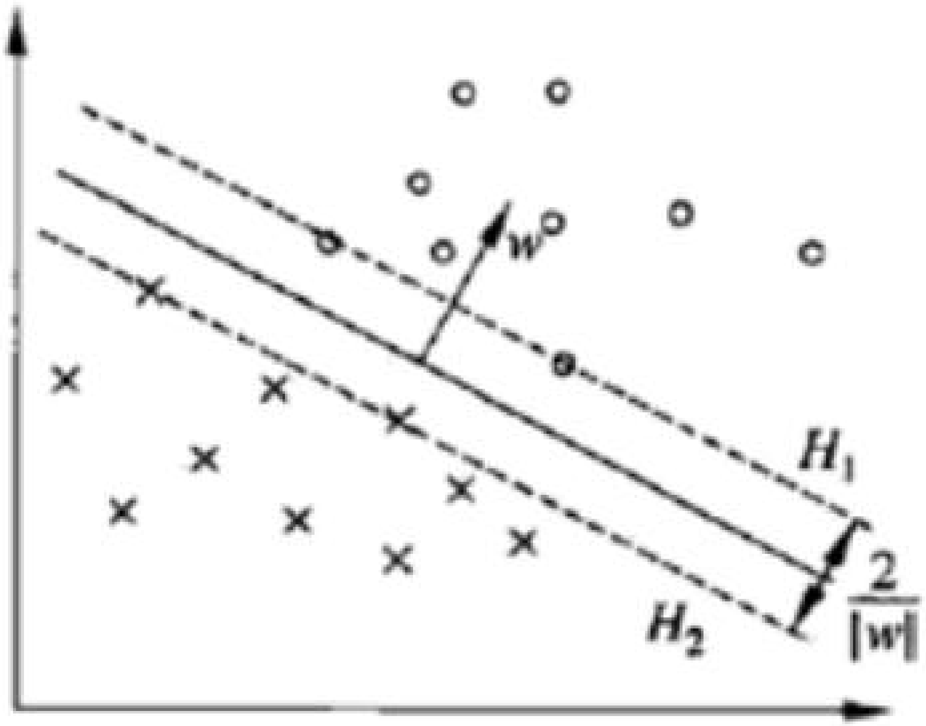
$$H_{1,2}: w \cdot x + b = \pm 1$$

上.

7.1 线性可分支持向量机（硬间隔最大化）

- **支持向量和间隔边界图解**

如下图所示，$H_{1,2}$称为间隔边界，$H_{1,2}$上的点称为支持向量.



7.1 线性可分支持向量机（硬间隔最大化）

- **学习的对偶算法**

引进拉格朗日乘子$\alpha_i \geq 0, i = 1,2,\cdots,N$, 定义拉格朗日函数:

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

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

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:

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

即

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

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

$$\alpha_i \geq 0, \ i = 1,2,\cdots,N.$$

7.1 线性可分支持向量机（硬间隔最大化）

- **线性可分支持向量机对偶学习算法**

根据KKT条件，即得

$$\nabla_w L(w^*, b^*, \alpha^*) = w^* - \sum_{i=1}^{N} \alpha_i^* y_i x_i = 0,$$

$$\nabla_b L(w^*, b^*, \alpha^*) = -\sum_{i=1}^{N} \alpha_i^* y_i = 0,$$

$$\alpha_i^*(y_i(w^* \cdot x_i + b^*) - 1) = 0, i = 1,2,\cdots,N,$$

$$y_i(w^* \cdot x_i + b^*) - 1 \geq 0, i = 1,2,\cdots,N,$$

$$\alpha_i^* \geq 0, i = 1,2,\cdots,N.$$

选择α^*的一个正分量$\alpha_j^* > 0$，得到$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$，$b^* = y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i, x_j)$.

最后得到最大间隔分离超平面

$$\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i, x_j) = 0,$$

和分类决策函数

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i, x_j)\right).$$

7.2 线性支持向量机（软间隔最大化）

- **线性支持向量机的定义**

对于给定的线性不可分的训练数据集, 通过求解凸二次规划问题, 即软间隔最大化问题, 得到的分离超平面为

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

以及相应的分类决策函数

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

称为线性支持向量机.

7.2 线性支持向量机（软间隔最大化）

- **线性支持向量机的原始问题**

原始问题:

$$\min_{w,b,\xi} \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i,$$

$$s.t. \ \ y_i(w\cdot x_i + b) \geq 1-\xi_i, \ \ i=1,2,\cdots,N,$$

$$\xi_i \geq 0, \ \ i=1,2,\cdots,N.$$

7.2 线性支持向量机（软间隔最大化）

- **线性支持向量机的对偶问题**

对偶问题:

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

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

$$0 \le \alpha_i \le C, \ i = 1,2,\cdots,N.$$

7.2 线性支持向量机（软间隔最大化）

- **合页损失函数**

函数

$$L\big(y(w \cdot x + b)\big) = [1 - y(w \cdot x + b)]_+,$$

称为合页损失函数.

线性支持向量机的原始最优化问题等价于最优化问题:

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

其中, 目标函数的第1项是经验损失或经验风险, 第2项是系数为λ的w的L_2范数, 是正则化项.

7.2 线性支持向量机（软间隔最大化）

- **合页损失函数图解**

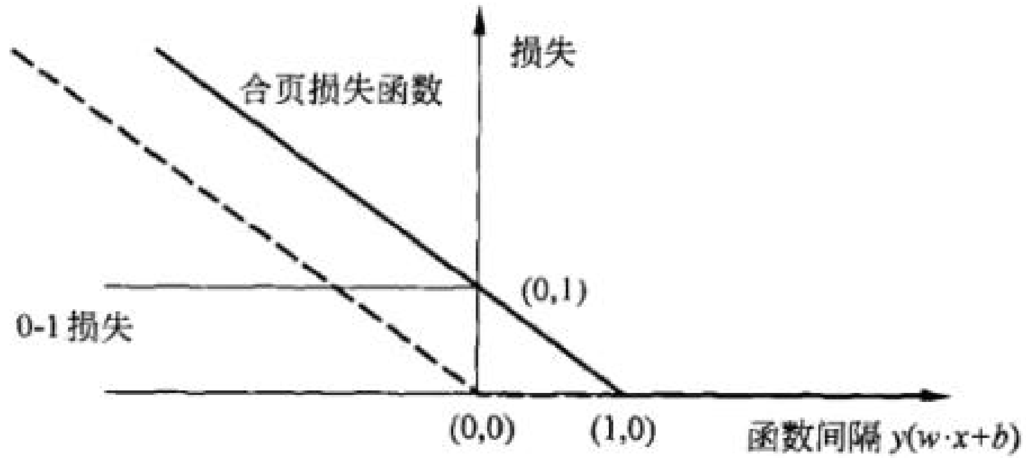


图 7.6　合页损失函数

7.3 非线性支持向量机与核函数

- **非线性分类问题**

非线性分类问题是指通过利用非线性模型才能很好地进行分类地问题.

用线性分类方法求解非线性分类问题分为两步: 首先使用一个变换将原空间的数据映射到新空间; 然后在新空间里用线性分类学习方法从训练数据中学习分类模型, 核技巧就属于这样的方法.

核技巧应用到支持向量机, 其基本想法就是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H), 使得在输入空间R^n中的超曲面模型对应于特征空间H中的超平面模型(支持向量机), 这样, 分类问题的学习任务通过在特征空间中求解线性支持向量机就可以完成.

7.3 非线性支持向量机与核函数

- **核函数的定义**

设X是输入空间(欧式空间R^n的子集或离散集合), 又设H为特征空间(希尔伯特空间), 如果存在一个从X到H的映射

$$\phi(x): X \rightarrow H$$

使得对所有$x, z \in X$, 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.

7.3 非线性支持向量机与核函数

- **核函数的另一定义**

设$X \subset R^n$, $K(x,z)$是定义在$X \times X$上的对称函数, 如果对任意$x_i \in X, i = 1,2,\cdots,m$, $K(x,z)$对应的$Gram$矩阵

$$K = [K(x_i, x_j)]_{m\times m}$$

是半正定矩阵, 则称$K(x,z)$为核函数.

7.3 非线性支持向量机与核函数

- **非线性支持向量机学习算法**

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

输出: 分类决策函数

(1)选取适当的核函数$K(x, z)$和适当的参数C, 构造并求解最优化问题

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

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

$$0 \le \alpha_i \le C, \ i = 1,2, \cdots, N.$$

求得最优解$\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*)^T$.

(2)选择α^*的一个正分量$0 < \alpha_j^* < C$, 计算$b^* = y_j - \sum_{i=1}^{N}\alpha_i^* y_i K(x_i, x_j)$.

(3)构造决策函数$f(x) = sign\left(\sum_{i=1}^{N}\alpha_i^* y_i K(x \cdot x_i) + y_j - \sum_{i=1}^{N}\alpha_i^* y_i K\left(x_i, x_j\right)\right)$.

7.4 序列最小最优化（SMO）算法

- **SMO算法的基本思路**

如果所有变量的解都满足此最优化问题的KKT条件, 那么这个最优化问题的解就得到了. 因为KKT条件是该最优化问题的充分必要条件. 否则, 选择两个变量, 固定其他变量, 针对这两个变量构建一个二次规划问题. 这个二次规划问题关于这两个变量的解应该更接近原始二次规划问题的解, 因为这会使得原始二次规划问题的目标函数值变得更小. 重要的是, 这时子问题可以通过解析方法求解, 这样就可以大大提高整个算法的计算速度. 子问题有两个变量, 一个是违反KKT条件最严重的那一个, 另一个由约束条件自动确定. 如此SMO算法将原问题不断分解为子问题并对子问题求解, 进而达到求解原问题的目的.

整个SMO算法包括两个部分: 求解两个变量二次规划的解析方法和选择变量的启发式方法.

7.4 序列最小最优化（SMO）算法

- **两个变量二次规划的求解方法**

不失一般性, 假设选择的两个变量是α_1, α_2, 其他变量$\alpha_i (i = 3,4,\cdots,N)$是固定的. 于是SMO的最优化问题的子问题可以写成:

$$\min_{\alpha_1,\alpha_2} W(\alpha_1,\alpha_2) = \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_1y_2K_{12}\alpha_1\alpha_2 - (\alpha_1+\alpha_2) + y_1\alpha_1\sum_{i=3}^{N} y_i\alpha_iK_{i1} + y_2\alpha_2\sum_{i=3}^{N} y_i\alpha_iK_{i2},$$

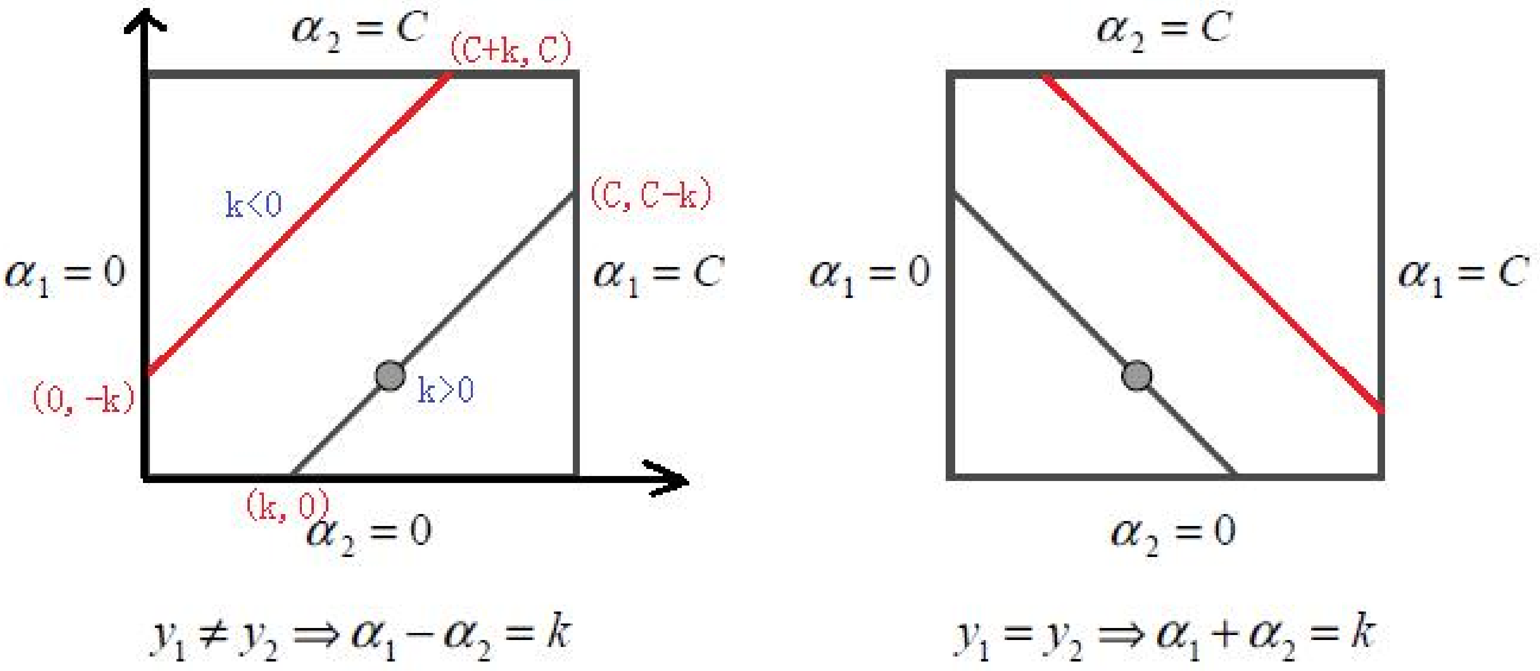
$$s.t.\ \ \alpha_1y_1 + \alpha_2y_2 = -\sum_{i=3}^{N} y_i\alpha_i = \varsigma,$$

$$0 \le \alpha_i \le C,\ \ i = 1,2,$$

其中, $K_{ij} = K\left(x_i, x_j\right), i, j = 1,2,\cdots,N, \varsigma$是常数, 目标函数式中省略了不含$\alpha_1, \alpha_2$的常数项.

7.4 序列最小最优化（SMO）算法

- **两个变量二次规划的约束条件图解**



7.4 序列最小最优化（SMO）算法

- **两个变量二次规划的解**

假设两个变量二次规划问题的初始解为$\alpha_1^{old}, \alpha_2^{old}$，最优解为$\alpha_1^{new}, \alpha_2^{new}$，并且假设在沿着约束方向未经剪辑时$\alpha_2$的最优解为$\alpha_2^{new,unc}$.

当$y_1 \neq y_2$时，$L = \max\left(0, \alpha_2^{old} - \alpha_1^{old}\right)$，$H = \min(C, C + \alpha_2^{old} - \alpha_1^{old})$; 当$y_1 = y_2$时，$L = \max\left(0, \alpha_2^{old} + \alpha_1^{old} - C\right)$，$H = \min(C, \alpha_2^{old} + \alpha_1^{old})$. 最优值$\alpha_2^{new}$的取值范围必须满足条件$L \leq \alpha_2^{new} \leq H$.

令$E_i = \left(\sum_{j=1}^{N} \alpha_j y_j K\left(x_j, x_i\right) + b\right) - y_i, i = 1,2$. 则$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{K_{11} + K_{22} - 2K_{12}}$, 进而,

$$\alpha_2^{new} = \begin{cases} H, \alpha_2^{new,unc} > H, \\ \alpha_2^{new,unc}, L \leq \alpha_2^{new,unc} \leq H, \\ L, \alpha_2^{new,unc} < L. \end{cases}$$

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2\left(\alpha_2^{old} - \alpha_2^{new}\right).$$

7.4 序列最小最优化（SMO）算法

- **变量的选择**

SMO算法称选择第一个变量为外层循环, 这个变量需要选择在训练集中违反KKT条件最严重的样本点. 首先遍历所有在间隔边界上的支持向量点, 检验它们是否满足KKT条件. 如果这些样本都满足KKT条件, 那么遍历整个训练集, 检验它们是否满足KKT条件.

SMO算法称选择第二个变量为内层循环, 假设我们在外层循环已经找到了α_1, 第二个变量α_2的选择标准是使得$|E_1 - E_2|$最大. 因为α_1已定, E_1也确定了. 如果E_1是正的, 那么选择最小的E_i作为E_2; 如果E_1是负的, 那么选择最大的E_i作为E_2. 可以将所有的E_i保存下来加快迭代.如果内层循环找到的点不能让目标函数有足够的下降, 可以遍历训练数据集; 若仍找不到合适的α_2, 则放弃第一个α_1, 再通过外层循环寻求另外的α_1.

7.4 序列最小最优化（SMO）算法

- **SMO算法**

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

输出: 近似解$\hat{\alpha}$.

(1)取初值$\alpha^{(0)} = 0$, 令$k = 0$;

(2)选取优化变量$\alpha_1^{(k)}, \alpha_2^{(k)}$, 解析求解两个变量的最优化问题, 求得最优解$\alpha_1^{(k+1)}, \alpha_2^{(k+1)}$, 更新$\alpha$为$\alpha^{(k+1)}$;

(3)若在精度ε范围内满足停机条件

$$\sum_{i=1}^{N} \alpha_i y_i = 0,$$
$$0 \le \alpha_i \le C,\ i = 1,2,\cdots,N,$$

$$y_i \cdot g(x_i) = \begin{cases} \ge 1, \{x_i | \alpha_i = 0\}, \\ = 1, \{x_i | 0 < \alpha_i < C\}, \\ \le 1, \{x_i | \alpha_i = C\}. \end{cases}$$

其中, $g(x_i) = \sum_{j=1}^{N} \alpha_j y_j K(x_j, x_i) + b$, 则转(4); 否则令$k = k + 1$, 转(2);

(4)取$\hat{\alpha} = \alpha^{(k+1)}$.