3_支持向量机

3_1_线性可分支持向量机

训练数据集

$$T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$$

其中，$\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^n, y_i \in \mathcal{Y} = \{+1, -1\}, i = 1, 2, \cdots, N$，$\mathbf{x}_i$为第$i$个特征向量（实例），$y_i$为第$\mathbf{x}_i$的类标记，当$y_i = +1$时，称$\mathbf{x}_i$为正例；当$y_i = -1$时，称$\mathbf{x}_i$为负例，$(\mathbf{x}_i, y_i)$称为样本点。

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

$$\mathbf{w}^* \cdot \mathbf{x} + b^* = 0$$

以及相应的分类决策函数

$$f(\mathbf{x}) = sign\left(\mathbf{w}^* \cdot \mathbf{x} + b^*\right)$$

称为线型可分支持向量机。其中，$\mathbf{w}^*$和b^*为感知机模型参数，$\mathbf{w}^* \in \mathbb{R}^n$叫做权值或权值向量，$b^* \in R$叫偏置，$\mathbf{w}^* \cdot \mathbf{x}$表示$\mathbf{w}^*$和$\mathbf{x}$的内积。$sign$是符号函数。

超平面$(\mathbf{w}, b)$关于样本点$(\mathbf{x}_i, y_i)$的函数间隔为

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

超平面$(\mathbf{w}, b)$关于训练集T的函数间隔

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

即超平面$(\mathbf{w}, b)$关于训练集T中所有样本点$(\mathbf{x}_i, y_i)$的函数间隔的最小值。

超平面$(\mathbf{w}, b)$关于样本点$(\mathbf{x}_i, y_i)$的几何间隔为

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

超平面$(\mathbf{w}, b)$关于训练集T的几何间隔

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

即超平面$(\mathbf{w}, b)$关于训练集T中所有样本点$(\mathbf{x}_i, y_i)$的几何间隔的最小值。

函数间隔和几何间隔的关系

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

最大间隔分离超平面等价为求解

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

等价的

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

等价的

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

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$T = \{ (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N) \}$，其中$\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^n, y_i \in \mathcal{Y} = \{+1, -1\}, i = 1, 2, \cdots, N$
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

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

   求得最优解$\mathbf{w}^*, b^*$。
2. 得到分离超平面

$$\mathbf{w}^* \cdot \mathbf{x} + b^* = 0$$

   以及分类决策函数

$$f(\mathbf{x}) = sign \left(\mathbf{w}^* \cdot \mathbf{x} + b^* \right)$$

（硬间隔）支持向量：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，即使约束条件等号成立的样本点

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 = 0$$

对$y_i = +1$的正例点，支持向量在超平面

$$H_1 : \mathbf{w} \cdot \mathbf{x} + b = 1$$

对$y_i = -1$的正例点，支持向量在超平面

$$H_1 : \mathbf{w} \cdot \mathbf{x} + b = -1$$

H_1和H_2称为间隔边界。
H_1和H_2之间的距离称为间隔，且$|H_1 H_2| = \frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$。

最优化问题的求解：

1. 引入拉格朗日乘子$\alpha_i \geq 0, i = 1, 2, \cdots, N$构建拉格朗日函数

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

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

2. 求$\min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)$:

$$
\begin{aligned}
\nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) &= \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \\
\nabla_b L(\mathbf{w}, b, \alpha) &= -\sum_{i=1}^{N} \alpha_i y_i = 0
\end{aligned}
$$

得

$$
\begin{aligned}
\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \\
\sum_{i=1}^{N} \alpha_i y_i = 0
\end{aligned}
$$

代入拉格朗日函数，得

$$
\begin{aligned}
L(\mathbf{w}, b, \alpha) &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) - \sum_{i=1}^{N} \alpha_i y_i \left[\left(\left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j\right) \cdot \mathbf{x}_i + b\right)\right] + \sum_{i=1}^{N} \alpha_i \\
&= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i \\
&= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) + \sum_{i=1}^{N} \alpha_i
\end{aligned}
$$

即

$$
\min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) + \sum_{i=1}^{N} \alpha_i
$$

3.求$\max_{\alpha} \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)$:

$$
\begin{aligned}
& \max_{\alpha} -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) + \sum_{i=1}^{N} \alpha_i \\
& s.t. \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& \alpha_i \geq 0, \quad i = 1, 2, \cdots, N
\end{aligned}
$$

等价的

$$
\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) - \sum_{i=1}^{N} \alpha_i
$$

线性可分支持向量机（硬间隔支持向量机）学习算法：
输入：线性可分训练数据集$T=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_N,y_N)\}$，其中$\mathbf{x}_i\in\mathcal{X}=\mathbb{R}^n, y_i\in\mathcal{Y}=\{+1,-1\}, i=1,2,\cdots,N$
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

$$\begin{aligned}&\min_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)-\sum_{i=1}^{N}\alpha_i\\&s.t.\sum_{i=1}^{N}\alpha_i y_i=0\\&\alpha_i\geq 0,\quad i=1,2,\cdots,N\end{aligned}$$

   求得最优解$\alpha^*=\left(\alpha_1^*,\alpha_1^*,\cdots,\alpha_N^*\right)$。
2. 计算

$$\mathbf{w}^*=\sum_{i=1}^{N}\alpha_i^* y_i\mathbf{x}_i$$

   并选择α^*的一个正分量$\alpha_j^*>0$，计算

$$b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)$$

3. 得到分离超平面

$$\mathbf{w}^*\cdot\mathbf{x}+b^*=0$$

   以及分类决策函数

$$f(\mathbf{x})=sign\left(\mathbf{w}^*\cdot\mathbf{x}+b^*\right)$$

3_2_线性支持向量机

线性支持向量机（软间隔支持向量机）：给定线性不可分训练数据集，通过求解凸二次规划问题

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

学习得到分离超平面为

$$\mathbf{w}^*\cdot\mathbf{x}+b^*=0$$

以及相应的分类决策函数

$$f(\mathbf{x})=sign\left(\mathbf{w}^*\cdot\mathbf{x}+b^*\right)$$

称为线型支持向量机。

最优化问题的求解：

1. 引入拉格朗日乘子$\alpha_i \geq 0, \mu_i \geq 0, i = 1, 2, \cdots, N$构建拉格朗日函数

$$
\begin{aligned}
L(\mathbf{w}, b, \xi, \alpha, \mu) &= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i\left[-y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) + 1 - \xi_i\right] + \sum_{i=1}^{N}\mu_i\left(-\xi_i\right) \\
&= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{N}\mu_i\xi_i
\end{aligned}
$$

其中，$\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_N)^T$以及$\mu = (\mu_1, \mu_2, \cdots, \mu_N)^T$为拉格朗日乘子向量。

2. 求$\min_{\mathbf{w},b} L(\mathbf{w}, b, \xi, \alpha, \mu)$:

$$
\begin{aligned}
&\nabla_{\mathbf{w}} L(\mathbf{w}, b, \xi, \alpha, \mu) = \mathbf{w} - \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i = 0 \\
&\nabla_b L(\mathbf{w}, b, \xi, \alpha, \mu) = -\sum_{i=1}^{N}\alpha_i y_i = 0 \\
&\nabla_{\xi_i} L(\mathbf{w}, b, \xi, \alpha, \mu) = C - \alpha_i - \mu_i = 0
\end{aligned}
$$

得

$$
\begin{aligned}
&\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i \\
&\sum_{i=1}^{N}\alpha_i y_i = 0 \\
&C - \alpha_i - \mu_i = 0
\end{aligned}
$$

代入拉格朗日函数，得

$$
\begin{aligned}
L(\mathbf{w}, b, \xi, \alpha, \mu) &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right) + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i y_i\left[\left(\left(\sum_{j=1}^{N}\alpha_j y_j x_j\right)\cdot\mathbf{x}_i + b\right] + \right. \\
&= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right) - \sum_{i=1}^{N}\alpha_i y_i b + \sum_{i=1}^{N}\alpha_i + \sum_{i=1}^{N}\xi_i\left(C - \alpha_i - \mu_i\right) \\
&= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right) + \sum_{i=1}^{N}\alpha_i
\end{aligned}
$$

即

$$
\min_{\mathbf{w},b,\xi} L(\mathbf{w}, b, \xi, \alpha, \mu) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right) + \sum_{i=1}^{N}\alpha_i
$$

3.求$\max_{\alpha}\min_{\mathbf{w},b,\xi} L(\mathbf{w}, b, \xi, \alpha, \mu)$:

$$
\begin{aligned}
&\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right) + \sum_{i=1}^{N}\alpha_i \\
&s.t. \sum_{i=1}^{N}\alpha_i y_i = 0 \\
&C - \alpha_i - \mu_i = 0 \\
&\alpha_i \geq 0 \\
&\mu_i \geq 0, \quad i = 1, 2, \cdots, N
\end{aligned}
$$

线性支持向量机（软间隔支持向量机）学习算法：
输入：训练数据集$T=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_N,y_N)\}$，其中$\mathbf{x}_i\in\mathcal{X}=\mathbb{R}^n, y_i\in\mathcal{Y}=\{+1,-1\}, i=1,2,\cdots,N$
输出：最大间隔分离超平面和分类决策函数

1. 选择惩罚参数$C\geq 0$，构建并求解约束最优化问题

$$
\begin{aligned}
&\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)-\sum_{i=1}^{N}\alpha_i\\
&s.t. \sum_{i=1}^{N}\alpha_i y_i=0\\
&0\leq\alpha_i\leq C,\quad i=1,2,\cdots,N
\end{aligned}
$$

   求得最优解$\alpha^*=\left(\alpha_1^*,\alpha_1^*,\cdots,\alpha_N^*\right)$。
2. 计算

$$
\mathbf{w}^*=\sum_{i=1}^{N}\alpha_i^* y_i\mathbf{x}_i
$$

   并选择α^*的一个分量$0<\alpha_j^*<C$，计算

$$
b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)
$$

3. 得到分离超平面

$$
\mathbf{w}^*\cdot\mathbf{x}+b^*=0
$$

   以及分类决策函数

$$
f(\mathbf{x})=sign\left(\mathbf{w}^*\cdot\mathbf{x}+b^*\right)
$$

（软间隔）支持向量：线性不可分情况下，最优化问题的解$\alpha^*=\left(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*\right)^T$中对应于$\alpha_i^*>0$的样本点$(x_i,y_i)$的实例$x_i$。

实例x_i的几何间隔

$$
\gamma_i=\frac{y_i\left(w\cdot x_i+b\right)}{\|w\|}=\frac{|1-\xi_i|}{\|w\|}
$$

且$\frac{1}{2}|H_1H_2|=\frac{1}{\|w\|}$
则实例x_i到间隔边界的距离

$$
\frac{\xi_i}{\|w\|}
$$

$$
\xi_i\geq 0\Leftrightarrow\begin{cases}\xi_i=0, x_i\text{在间隔边界上;}\\ 0<\xi_i<1, x_i\text{在间隔边界与分离超平面之间;}\\ \xi_i=1, x_i\text{在分离超平面上;}\\ \xi_i>1, x_i\text{在分离超平面误分类一侧;}\end{cases}
$$

线性支持向量机（软间隔）的合页损失函数

$$L\left(y\left(\mathbf{w}\cdot\mathbf{x}+b\right)\right)=\left[1-y\left(\mathbf{w}\cdot\mathbf{x}+b\right)\right]_{+}$$

其中，“+”为取正函数

$$[z]_{+}=\begin{cases} z, z>0 \\ 0, z\leq 0\end{cases}$$

3_3_非线性支持向量

核函数
设$\mathcal{X}$是输入空间（欧氏空间$\mathbb{R}^n$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

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

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

$$K\left(\mathbf{x},\mathbf{z}\right)=\phi\left(\mathbf{x}\right)\cdot\phi\left(\mathbf{z}\right)$$

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

常用核函数：

1. 多项式核函数

$$K\left(\mathbf{x},\mathbf{z}\right)=\left(\mathbf{x}\cdot\mathbf{z}+1\right)^{p}$$

2. 高斯核函数

$$K\left(\mathbf{x},\mathbf{z}\right)=\exp\left(-\frac{\|\mathbf{x}-\mathbf{z}\|^{2}}{2\sigma^{2}}\right)$$

非线性支持向量机：从非线性分类训练集，通过核函数与软间隔最大化，学习得到分类决策函数

$$f\left(\mathbf{x}\right)=sign\left(\sum_{i=1}^{N}\alpha_i^{*}y_iK\left(\mathbf{x},\mathbf{x}_i\right)+b^{*}\right)$$

称为非线性支持向量机，$K\left(\mathbf{x},\mathbf{z}\right)$是正定核函数。

非线性支持向量机学习算法：
输入：训练数据集$T=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_N,y_N)\}$，其中$\mathbf{x}_i\in\mathcal{X}=\mathbb{R}^n, y_i\in\mathcal{Y}=\{+1,-1\}, i=1,2,\cdots,N$
输出：分类决策函数

1. 选择适当的核函数$K(\mathbf{x},\mathbf{z})$和惩罚参数$C\geq 0$，构建并求解约束最优化问题

$$
\begin{aligned}
&\min_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K\left(\mathbf{x}_i,\mathbf{x}_j\right)-\sum_{i=1}^{N}\alpha_i\\
&s.t.\sum_{i=1}^{N}\alpha_i y_i=0\\
&0\leq\alpha_i\leq C,\quad i=1,2,\cdots,N
\end{aligned}
$$

   求得最优解$\alpha^*=\left(\alpha_1^*,\alpha_1^*,\cdots,\alpha_N^*\right)$。
2. 计算

$$
\mathbf{w}^*=\sum_{i=1}^{N}\alpha_i^* y_i\mathbf{x}_i
$$

   并选择α^*的一个分量$0<\alpha_j^*<C$，计算

$$
b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i K\left(\mathbf{x}_i,\mathbf{x}_j\right)
$$

3. 得到分离超平面

$$
\mathbf{w}^*\cdot\mathbf{x}+b^*=0
$$

   以及分类决策函数

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

3_4_序列最小最优化算法

序列最小最优化（sequential minimal optimization，SMO）算法 要解如下凸二次规划的对偶问题：

$$
\begin{aligned}
&\min_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K\left(\mathbf{x}_i,\mathbf{x}_j\right)-\sum_{i=1}^{N}\alpha_i\\
&s.t.\sum_{i=1}^{N}\alpha_i y_i=0\\
&0\leq\alpha_i\leq C,\quad i=1,2,\cdots,N
\end{aligned}
$$

选择α_1, α_2两个变量，其他变量$\alpha_i \ (i = 3, 4, \cdots, N)$是固定的，SMO的最优化问题的子问题

$$
\begin{aligned}
\min_{\alpha_1,\alpha_2} W(\alpha_1, \alpha_2) = & \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_1 y_2 K_{12}\alpha_1\alpha_2 \\
& - (\alpha_1 + \alpha_2) + y_1\alpha_1 \sum_{i=3}^{N} y_i\alpha_i K_{i1} + y_2\alpha_2 \sum_{i=3}^{N} y_i\alpha_i K_{i2} \\
s.t. \quad \alpha_1 + \alpha_2 = & - \sum_{i=3}^{N} \alpha_i y_i = \varsigma \\
0 \le \alpha_i \le C, & \quad i = 1, 2
\end{aligned}
$$

其中，$K_{ij} = K\left(\mathbf{x}_i, \mathbf{x}_j\right), i, j = 1, 2, \cdots, N, \varsigma$是常数，且省略了不含$\alpha_1, \alpha_2$的常数项。

设凸二次规划的对偶问题的初始可行解为$\alpha_1^{old}, \alpha_2^{old}$，最优解为$\alpha_1^{new}, \alpha_2^{new}$，且在沿着约束方向未经剪辑时$\alpha_2$的最优解为$\alpha_2^{new,unc}$。

由于α_2^{new}需要满足$0 \le \alpha_i \le C$，所以最优解α_2^{new}的取值范围需满足

$$
L \le \alpha_2^{new} \le H
$$

其中，L与H是α_2^{new}所在的对角线段断点的界。
如果$y_1 \neq y_2$，则

$$
L = \max\left(0, \alpha_2^{old} - \alpha_1^{old}\right), H = \min\left(C, C + \alpha_2^{old} - \alpha_1^{old}\right)
$$

如果$y_1 = y_2$，则

$$
L = \max\left(0, \alpha_2^{old} + \alpha_1^{old} - C\right), H = \min\left(C, \alpha_2^{old} + \alpha_1^{old}\right)
$$

记

$$
g(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b
$$

令

$$
\begin{aligned}
E_i &= g(\mathbf{x}_i) - y_i = \left(\sum_{j=1}^{N} \alpha_j y_j K\left(\mathbf{x}_j, \mathbf{x}_i\right) + b\right) - y_i, \quad i = 1, 2 \\
v_i &= \sum_{j=3}^{N} \alpha_j y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) = g(\mathbf{x}_i) - \sum_{j=1}^{2} \alpha_j y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) - b, \quad i = 1, 2
\end{aligned}
$$

则

$$
\begin{aligned}
W(\alpha_1, \alpha_2) = & \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_1 y_2 K_{12}\alpha_1\alpha_2 \\
& - (\alpha_1 + \alpha_2) + y_1 v_1 \alpha_1 + y_2 v_2 \alpha_2
\end{aligned}
$$

由于$\alpha_1 y_1 = \varsigma, y_i^2 = 1$，可将$\alpha_1$表示为

$$\alpha_1 = (\varsigma - y_2\alpha_2)\, y_1$$

代入，得

$$\begin{aligned} W(\alpha_2) &= \frac{1}{2}K_{11}[(\varsigma - y_2\alpha_2)\, y_1]^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_1 y_2 K_{12}(\varsigma - y_2\alpha_2)\, y_1\alpha_2 \\ &\quad - [(\varsigma - y_2\alpha_2)\, y_1 + \alpha_2] + y_1 v_1 (\varsigma - y_2\alpha_2)\, y_1 + y_2 v_2 \alpha_2 \\ &= \frac{1}{2}K_{11}(\varsigma - y_2\alpha_2)^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_2 K_{12}(\varsigma - y_2\alpha_2)\,\alpha_2 \\ &\quad - (\varsigma - y_2\alpha_2)\, y_1 - \alpha_2 + v_1(\varsigma - y_2\alpha_2) + y_2 v_2\alpha_2 \end{aligned}$$

对α_2求导

$$\begin{aligned} \frac{\partial W}{\partial \alpha_2} &= K_{11}\alpha_2 + K_{22}\alpha_2 - 2K_{12}\alpha_2 \\ &\quad - K_{11}\varsigma y_2 + K_{12}\varsigma y_2 + y_1 y_2 - 1 - v_1 y_2 + y_2 v_2 \end{aligned}$$

令其为0，得

$$\begin{aligned} (K_{11} + K_{22} - 2K_{12})\,\alpha_2 &= y_2 (y_2 - y_1 + \varsigma K_{11} - \varsigma K_{12} + v_1 - v_2) \\ &= y_2 \left[y_2 - y_1 + \varsigma K_{11} - \varsigma K_{12} + \left(g(\mathbf{x}_1) - \sum_{j=1}^{2} \alpha_j y_j K_{1j} - b \right) - \left(g(\mathbf{x}_2) - \sum_{j=}^{2} \right.\right. \end{aligned}$$

将$\varsigma = \alpha_1^{old} y_1 + \alpha_2^{old} y_2$代入，得

$$\begin{aligned} (K_{11} + K_{22} - 2K_{12})\,\alpha_2^{new,unc} &= y_2 \left((K_{11} + K_{22} - 2K_{12})\,\alpha_2^{old} y_2 + y_2 - y_1 + g(x_1) - g(x_2) \right) \\ &= (K_{11} + K_{22} - 2K_{12})\,\alpha_2^{old} + y_2 (E_1 - E_2) \end{aligned}$$

令$\eta = K_{11} + K_{22} - 2K_{12}$代入，得

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2 (E_1 - E_2)}{\eta}$$

经剪辑后

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

由于$\varsigma = \alpha_1^{old} y_1 + \alpha_2^{old} y_2$及$\varsigma = \alpha_1^{new} y_1 + \alpha_2^{new} y_2$
则

$$\begin{aligned} \alpha_1^{old} y_1 + \alpha_2^{old} y_2 &= \alpha_1^{new} y_1 + \alpha_2^{new} y_2 \\ \alpha_1^{new} &= \alpha_1^{old} + y_1 y_2 \left(\alpha_2^{old} - \alpha_2^{new} \right) \end{aligned}$$

由分量$0 < \alpha_1^{new} < C$，则

$$b_1^{new} = y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i1} - \alpha_1^{new} y_1 K_{11} - \alpha_2^{new} y_2 K_{21}$$

由

$$\begin{aligned} E_1 &= g(\mathbf{x}_1) - y_1 = \left(\sum_{j=1}^{N} \alpha_j y_j K_{ij} + b \right) - y_1 \\ &= \sum_{i=3}^{N} \alpha_i y_i K_{i1} + \alpha_1^{old} y_1 K_{11} + \alpha_2^{old} y_2 K_{21} + b^{old} - y_1 \end{aligned}$$

则

$$y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i1} = -E_1 + \alpha_1^{old} y_1 K_{11} + \alpha_2^{old} y_2 K_{21} + b^{old}$$

代入，得

$$b_1^{new} = -E_1 + y_1 K_{11} \left(\alpha_1^{new} - \alpha_1^{old} \right) - y_2 K_{21} \left(\alpha_2^{new} - \alpha_2^{old} \right) + b^{old}$$

同理，得

$$b_2^{new} = -E_2 + y_1 K_{12} \left(\alpha_1^{new} - \alpha_1^{old} \right) - y_2 K_{22} \left(\alpha_2^{new} - \alpha_2^{old} \right) + b^{old}$$

如果$\alpha_1^{new}, \alpha_2^{new}$满足$0 < \alpha_i^{new} < C, i = 1, 2,$
则

$$b^{new} = b_1^{new} = b_2^{new}$$

否则

$$b^{new} = \frac{b_1^{new} + b_2^{new}}{2}$$

更新E_i

$$E_i^{new} = \sum_{S} y_j \alpha_j K_{(\mathbf{x}_i, \mathbf{x}_j)} + b^{new} - y_i$$

其中，S是所有支持向量x_j的集合。

SMO算法：
输入：训练数据集$T=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_N,y_N)\}$，其中$\mathbf{x}_i\in\mathcal{X}=\mathbb{R}^n, y_i\in\mathcal{Y}=\{+1,-1\}, i=1,2,\cdots,N$，精度$\varepsilon$；
输出：近似解$\hat{\alpha}$

1. 取初始值$\alpha^0=0$，令$k=0$；
2. 选取优化变量$\alpha_1^{(k)},\alpha_2^{(k)}$，求解

$$\begin{aligned}\min_{\alpha_1,\alpha_2} W(\alpha_1,\alpha_2)=&\frac{1}{2}K_{11}\alpha_1^2+\frac{1}{2}K_{22}\alpha_2^2+y_1y_2K_{12}\alpha_1\alpha_2\\&-(\alpha_1+\alpha_2)+y_1\alpha_1\sum_{i=3}^N y_i\alpha_iK_{i1}+y_2\alpha_2\sum_{i=3}^N y_i\alpha_iK_{i2}\\s.t.\quad \alpha_1+\alpha_2=&-\sum_{i=3}^N\alpha_iy_i=\varsigma\\0\leq\alpha_i\leq C,&\quad i=1,2\end{aligned}$$

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

$$\begin{aligned}&\sum_{i=1}^N\alpha_iy_i=0\\&0\leq\alpha_i\leq C, i=1,2,\cdots,N\end{aligned}$$

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

则转4.；否则令$k=k+1$，转2.；
4.取$\hat{\alpha}=\alpha^{(k+1)}$。

3_5_SVM应用

In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets,linear_model,model_selection,svm

def load_data_classfication():
    iris=datasets.load_iris()
    X_train=iris.data
    y_train=iris.target
    return model_selection.train_test_split(X_train, y_train,test_size=0.25,
        random_state=0,stratify=y_train)

def test_SVC_linear(*data):
    X_train,X_test,y_train,y_test=data
    cls=svm.SVC(kernel='linear')
    cls.fit(X_train,y_train)
    print('Coefficients:%s, intercept %s'%(cls.coef_,cls.intercept_))
    print('Score: %.2f' % cls.score(X_test, y_test))

def test_SVC_poly(*data):
    X_train,X_test,y_train,y_test=data
    fig=plt.figure()

    degrees=range(1,20)
    train_scores=[]
    test_scores=[]
    for degree in degrees:
        cls=svm.SVC(kernel='poly',degree=degree)
        cls.fit(X_train,y_train)
        train_scores.append(cls.score(X_train,y_train))
        test_scores.append(cls.score(X_test, y_test))
    ax=fig.add_subplot(1,3,1)
    ax.plot(degrees,train_scores,label="Training score ",marker='+' )
    ax.plot(degrees,test_scores,label= " Testing  score ",marker='o' )
    ax.set_title( "SVC_poly_degree ")
    ax.set_xlabel("p")
    ax.set_ylabel("score")
    ax.set_ylim(0,1.05)
    ax.legend(loc="best",framealpha=0.5)

    gammas=range(1,20)
    train_scores=[]
    test_scores=[]
    for gamma in gammas:
        cls=svm.SVC(kernel='poly',gamma=gamma,degree=3)
        cls.fit(X_train,y_train)
        train_scores.append(cls.score(X_train,y_train))
        test_scores.append(cls.score(X_test, y_test))
    ax=fig.add_subplot(1,3,2)
    ax.plot(gammas,train_scores,label="Training score ",marker='+' )
    ax.plot(gammas,test_scores,label= " Testing  score ",marker='o' )
    ax.set_title( "SVC_poly_gamma ")
    ax.set_xlabel(r"$\gamma$")
    ax.set_ylabel("score")
    ax.set_ylim(0,1.05)
    ax.legend(loc="best",framealpha=0.5)

    rs=range(0,20)
    train_scores=[]
    test_scores=[]
    for r in rs:
        cls=svm.SVC(kernel='poly',gamma=10,degree=3,coef0=r)
        cls.fit(X_train,y_train)
        train_scores.append(cls.score(X_train,y_train))
        test_scores.append(cls.score(X_test, y_test))
    ax=fig.add_subplot(1,3,3)
    ax.plot(rs,train_scores,label="Training score ",marker='+' )
    ax.plot(rs,test_scores,label= " Testing  score ",marker='o' )
    ax.set_title( "SVC_poly_r ")
    ax.set_xlabel(r"r")
```