SVM 详解

guoliqiang@pku.edu.cn

C-SVM

1. 线性分类器

1.1 定义

1. R^n 空间中的超平面。

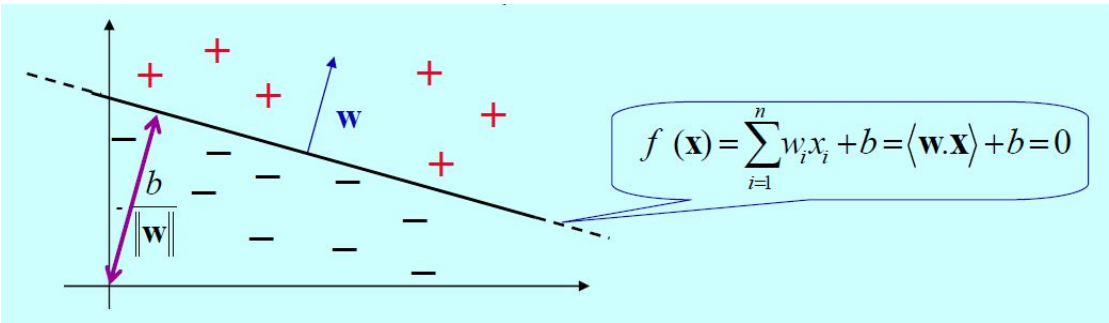
2. 由向量 $\boldsymbol{w}$ 和截距 b 定义。

3. 分类规则定义如下：

$$f(\mathbf{x})=\text{sgn}[\sum_{i=1}^{n}w_ix_i+b]=\begin{cases}+1 & if\sum_{i=1}^{n}w_ix_i+b>0\\ -1 & otherwise\end{cases} \quad (1)$$

$$\Rightarrow f(\boldsymbol{x})=\text{sgn}(<\boldsymbol{w}\cdot\boldsymbol{x}>+b)=\begin{cases}+1 & if<\boldsymbol{w}\cdot\boldsymbol{x}>+b>0\\ -1 & otherwise\end{cases}$$

注意：等于 0 的情况被分到了负例中。对于 $f(\boldsymbol{x})$ 的绝对值很小的情况，很难有好的处理方法；因为细微的变动（比如超平面稍微转一个小角度）就可能导致结果类别的改变。理想情况，期望 $f(\boldsymbol{x})$ 的值都是很大的正数或很小的负数，这样就能更加确信它是属于其中某一类别。



图[1]：线性分类器示意图

1 向量 $\boldsymbol{w}$ 的方向与超平面垂直。

2 $\frac{b}{\|\boldsymbol{w}\|}$ 相当于原点到超平面的距离；正负表示原点是在超平面的上侧还是下侧。

3 补充知识：点到直线的距离公式

设直线 $Ax+By+C=0$ 与点 (x_0,y_0)，则点到直线的距离为：

$$\frac{Ax_0+By_0+C}{\sqrt{A^2+B^2}} \quad (2)$$

2. 间隔

2.1 函数间隔（functional margin）

点 (x_i,y_i) 的函数间隔 γ_i w.r.t. $(\boldsymbol{w},b)$ 为：

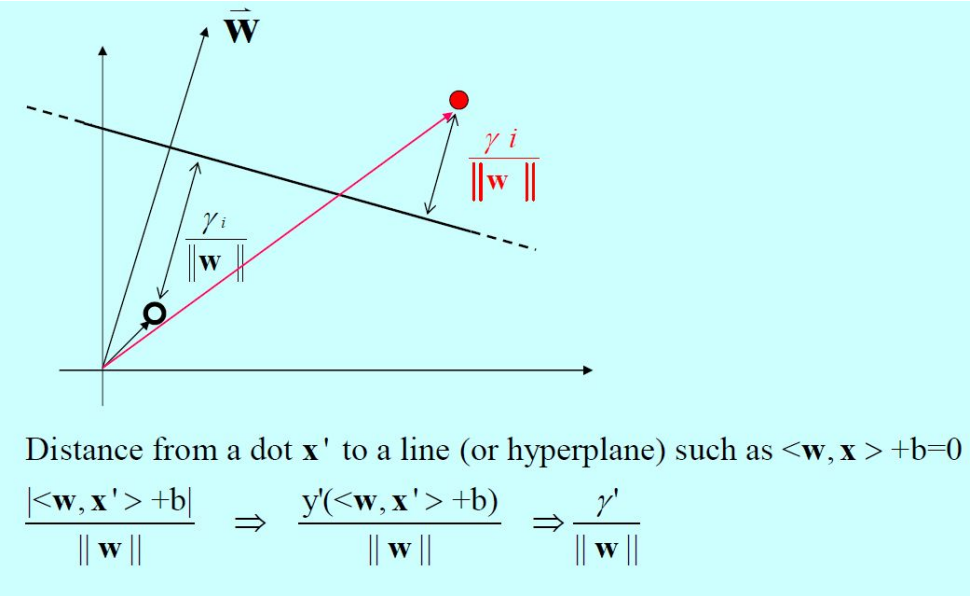
$$\gamma_i = y_i(<\mathbf{w}\cdot\mathbf{x_i}>+b) \text{ 其中 } y_i \in \{+1,-1\} \quad (3)$$

可以看到函数间隔为点到直线距离的分子，因为当 $<\mathbf{w}\cdot\mathbf{x_i}>+b<0$ 时 $y_i=-1$，乘以 y_i 起到了取绝对值的作用。

2.2 几何间隔（geometric margin）

$$\frac{\gamma_i}{\|\boldsymbol{w}\|} \quad (4)$$

其中 $\|\boldsymbol{w}\|=\sqrt{w_1^2+w_2^2+\ldots+w_n^2}$，$\|\boldsymbol{w}\|$ 是 2-范数。可以看到几何间隔就是点到超平面的距离。



图[2]：几何间隔实际含义

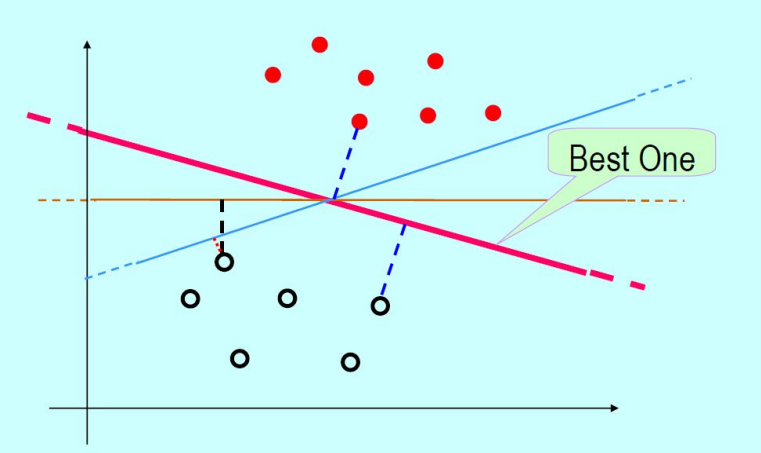
1. w.r.t.超平面 $(\boldsymbol{w},b)$，训练集 S 的函数间隔为：

$$\gamma_{S,\boldsymbol{w},b} = \min_{x_i \in S} \gamma_i \quad (5)$$

2. 训练集 S 的函数间隔为:

$$r_S = \max_{\boldsymbol{w},b} r_{S,\boldsymbol{w},b} \quad (6)$$

从后面可以看到 SVM 就是在寻找某个条件下的这个最大间隔 r_S，此时 $(\boldsymbol{w},b)$ 便为最大间隔超平面，此超平面可以最小化泛化误差(generalization error)，如下图：



图[3]:3 个不同但可以无误分开正负例的超平面

其中红颜色代表的超平面是最佳的，因为它不偏向于任何一方。（图中蓝颜色代表的距离相等）

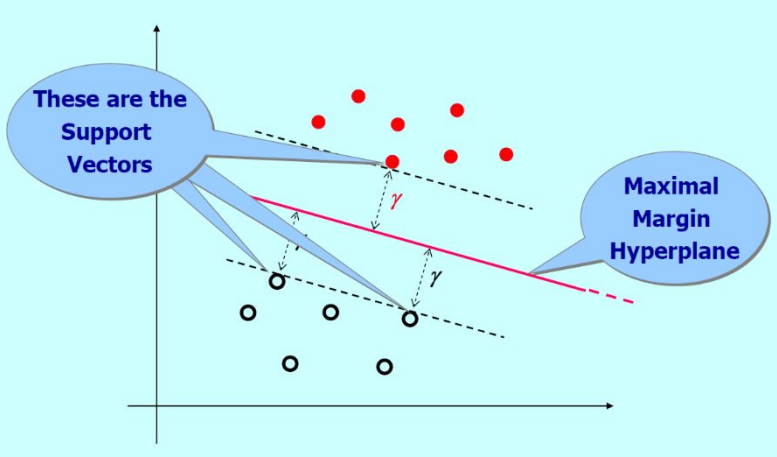


图 4：最大间隔超平面与支持向量点

可以看到最大间隔超平面的确定只需要图中的 3 个样例点便可以，其它的点对最大超平面的确定没有任何作用，（也就是，如果你能确定训练数据中哪些是这中类型的点，完全可以去掉其余的点，只用这些点训练模型；但不幸的是在超平面确定之前，这些点是无法确定的），这三个点便被称为“支持向量”。

2.3 函数间隔的问题

函数间隔：

$$\gamma_i = y_i(<\mathbf{w}\cdot\mathbf{x_i}>+b)\,(7)$$

由于可以随意成倍的改变 $(\boldsymbol{w},b)$ 而不改变其代表的超平面，比如 $<\mathbf{w}\cdot\mathbf{x_i}>+b=0$ 与 $<\lambda\times\mathbf{w}\cdot\mathbf{x_i}>+\lambda\times b=0$ 代表的超平面是一样的，但是点 $(\mathbf{x_i},y_i)$ 的函数间隔分别 γ_i 与 $\lambda\times\gamma_i$。为了计算方便，SVM 设定 $\gamma_i=1$，因为总能通过改变 λ 的值达到这一点，也就是 SVM 最终计算出的 $(\boldsymbol{w},b)$ 是在 $\gamma_i=1$ 情况下的。需要注意的是，几何间隔并不存在此问题。

$\gamma_i=1$ 情况下得到的最大间隔超平面 $(\boldsymbol{w}^*,b^*)$ 的几何间隔为：$\frac{1}{\|w^*\|}$，这样 SVM 的优化目标为：

$$\max \frac{1}{\| \boldsymbol{w} \|}$$
$$\Rightarrow \min \| \boldsymbol{w} \| \propto \min \frac{1}{2} < \boldsymbol{w} \cdot \boldsymbol{w} > \tag{8}$$

即寻找到几何间隔最大时的 $(\boldsymbol{w}, b)$。

3. 线性可分的 SVM

3.1 简介

$$\min_{\boldsymbol{w},b} \frac{1}{2} < \boldsymbol{w} \cdot \boldsymbol{w} >$$
$$s.t.\ y_i(< \boldsymbol{w} \cdot \boldsymbol{x}_i > + b) \geq 1 \tag{9}$$

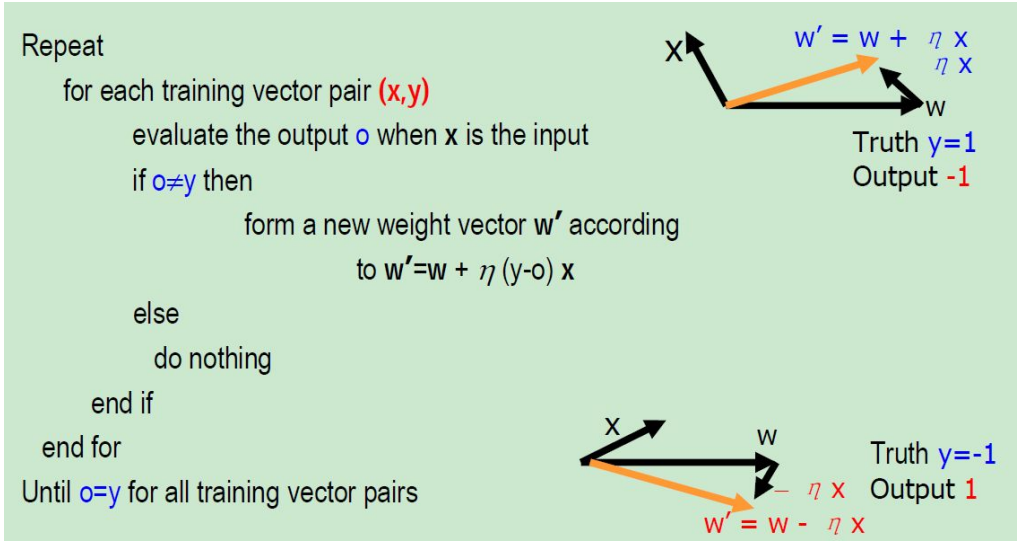
1. SVM 设定最大间隔分割超平面的在某个训练集合上的函数间隔为 1，这点可以从限制条件 $y_i(< \boldsymbol{w} \cdot \boldsymbol{x}_i > + b) \geq 1$ 得出。也就是，训练数据中点的函数间隔至少为 1，此时最大间隔分割超平面在训练集上的函数间隔为 1。

2. 公式 9 定义的 SVM 只有训练数据集合上确实存在这样一个超平面的情况下才会有解，因此称为线性可分的 SVM，然而现实情况中不总是存在有这样的超平面，此问题的解决方法可以从下文中看到。

3. 关于 $< \boldsymbol{w} \cdot \boldsymbol{x}_i > + b$

把 b 也看作一维，对应的 x_{n+1} 的值始终为 1， 这样相当于超平面和所有的点都会移动，直到超平面移动到到原点的位置。

这样，所有正例的 x 和<w, b>(下图中的 w)的夹角都会小于 90 度，所有负例的 x 和<w, b>的夹角都会大于 90 度。$< w, b > \bullet x = | < w, b > | \times |x| \times \cos$(夹角)，这样可以得出如下更新规则[1]：



3.2 凸函数与凸集

1. 对于 $\boldsymbol{w} \in R^n$，如果任意 $\boldsymbol{w}, \boldsymbol{u} \in R^n$，并且对于任意 $\theta \in (0,1)$ 有：

[1] http://stackoverflow.com/questions/1697243/perceptron-learning-algorithm-not-converging-to-0

$$f(\theta \boldsymbol{w}+(1-\theta)\boldsymbol{u}) \le \theta f(\boldsymbol{w})+(1-\theta)f(\boldsymbol{u}) \quad (10)$$

则 $f(\boldsymbol{w})$ 称为凸函数(下凸)。

2. 对于集合 $\Omega \in R^n$，如果 $\forall \boldsymbol{w},\boldsymbol{u} \in \Omega$ 并对于任何 $\theta \in (0,1)$，点 $(\theta \boldsymbol{w}+(1-\theta)\boldsymbol{u}) \in \Omega$，则称 $\Omega \in R^n$ 是凸的。

当目标函数和约束集合是凸的时候，目标函数的局部最小值 $\boldsymbol{w}^*$ 也是全局最小值，因为：对于任意 $\boldsymbol{u} \neq \boldsymbol{w}^*$，根据局部最小值的定义，存在充分接近 1 的 θ 使得：

$$f(\boldsymbol{w}^*) \le f(\theta \boldsymbol{w}^*+(1-\theta)f(\boldsymbol{u})) \quad (11)$$

根据凸集的性质可得：

$$\begin{aligned} & f(\boldsymbol{w}^*) \le f(\theta \boldsymbol{w}^*+(1-\theta)f(\boldsymbol{u})) \\ \Rightarrow\ & f(\boldsymbol{w}^*) \le \theta f(\boldsymbol{w}^*)+(1-\theta)f(\boldsymbol{u}) \\ \Rightarrow\ & (1-\theta)f(\boldsymbol{w}^*) \le (1-\theta)f(\boldsymbol{u}) \\ \Rightarrow\ & f(\boldsymbol{w}^*) \le f(\boldsymbol{u}) \end{aligned} \quad (12)$$

因此得出 $\boldsymbol{w}^*$ 是全局最小值。凸集和凸函数的性质保证了 SVM 的最优化问题是一个可解的问题。

3.3 带有限制条件的最优化问题

3.3.1 简介

其一般形式为如下所示：

$$\begin{aligned} & \min f(\boldsymbol{w}) \qquad \boldsymbol{w} \in R^n \\ & s.t.\ g_i(\boldsymbol{w}) \le 0, i=1,...,k \\ & \quad h_i(\boldsymbol{w}) = 0, i=1,...,m \end{aligned} \quad (13)$$

1. 可以看到线性 SVM 的求解可以转换成此类问题。

2. 当目标函数是一次或二次函数，限制条件都是一次函数时，其就是中学期间学过的线性规划问题。求解此问题最著名的方法就是拉格朗日乘子法。（由于 SVM 中的超平面 $f(\boldsymbol{w})$ 属于凸函数，因此其极值与最值相同）

3.3.2 拉格朗日乘子法

拉格朗日乘子法是在一种在等式限制条件下寻找目标函数极值的有效策略，例如：

$$\begin{aligned} & \min f(\boldsymbol{w}) \qquad \boldsymbol{w} \in R^n \\ & s.t.\ h_i(\boldsymbol{w}) = 0, i=1,...,m \end{aligned} \quad (14)$$

拉格朗日函数为：

$$L(\boldsymbol{w},\lambda) = f(\boldsymbol{w})+\sum \lambda_i h_i(\boldsymbol{w}) \quad (15)$$

求解 $\min f(\boldsymbol{w})$ 转换为求解 $\min L(\boldsymbol{w},\lambda)$，通过其求解过程可以看到 $f(\boldsymbol{w})$ 的极值和 $L(\boldsymbol{w},\lambda)$ 的极值是相同的，求解 $\min L(\boldsymbol{w},\lambda)$ 过程如下：

$$\begin{aligned}\frac{\partial L(\boldsymbol{w},\lambda)}{\partial \boldsymbol{w}} &= \frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}} + \frac{\sum \lambda_i h_i(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0 \\ \frac{\partial L(\boldsymbol{w},\lambda)}{\partial \lambda_i} &= h_i(\boldsymbol{w}) = 0\end{aligned} \quad (16)$$

可以看到若 $L(\boldsymbol{w},\lambda)$ 在 $(\boldsymbol{w}^*,\lambda^*)$ 取得极值，则根据上式有：$h_i(\boldsymbol{w}^*)=0$ 这样：

$$L(\boldsymbol{w}^*,\lambda^*) = f(\boldsymbol{w}^*) + \sum \lambda^*{}_i h_i(\boldsymbol{w}^*) = f(w^*) \quad (17)$$

很容易得出 $L(\boldsymbol{w}^*,\lambda^*)$ 是 $L(\boldsymbol{w},\lambda)$ 的极值，至于此时的 $f(w^*)$ 是不是具有约束条件 $h_i(\boldsymbol{w})=0$ 的目标函数 $f(\boldsymbol{w})$ 的极值[2]。

简单的说就是在满足约束条件 $h_i(\boldsymbol{w})=0$ 的 $f(\boldsymbol{w})$ 的极小值点 $\boldsymbol{w}^*$， $f(\boldsymbol{w})$ 的梯度 $\frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}$ 一定可以由各个约束条件的梯度 $\frac{h_i(\boldsymbol{w})}{\partial \boldsymbol{w}}$ 线性组合获得，这也就是如下公式存在的理由。

$$\frac{\partial L(\boldsymbol{w},\lambda)}{\partial \boldsymbol{w}} = \frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}} + \frac{\sum \lambda_i h_i(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0 \quad (18)$$

3.3.3 KKT 条件

不幸的是最初的拉格朗日乘子法不能处理限制条件含有不等式的情况，KKT 条件使得具有不等式限制条件的最优化问题同样可以转换非约束条件函数求极值问题。

形式化描述如下：

$$\begin{aligned}&\min f(\boldsymbol{w}) \qquad \boldsymbol{w} \in R^n \\ &s.t.\ g_i(\boldsymbol{w}) \le 0, i = 1,...,k \\ &\quad h_i(\boldsymbol{w}) = 0, i = 1,...,m\end{aligned} \quad (19)$$

其拉格朗日乘子法公式为：

$$L(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu}) = f(\boldsymbol{w}) + \sum \lambda_i h_w(\boldsymbol{w}) + \sum \mu_i g_i(\boldsymbol{w}) \quad (18)$$

注： $\boldsymbol{w} \in \Omega$，并且 Ω 是凸集，函数 f 为凸函数。

KKT 条件将原问题最优化 $\boldsymbol{w}$ 转化为最优化 $\boldsymbol{\alpha}$。

一个点 $\boldsymbol{w}^*$ 是最优解的充要条件为，存在 $\boldsymbol{\lambda}^*$ 与 $\boldsymbol{\mu}^*$ 使得：

---

[2] http://en.wikipedia.org/wiki/Lagrange_multiplier

《模式识别》Theodoridis

《支持向量机导论》P73

1. $\frac{\partial L(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)}{\partial \boldsymbol{w}}=0$

2. $\frac{\partial L(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)}{\partial \lambda}=h_i(\boldsymbol{w}^*)=0$

3. $\boldsymbol{\mu}^*_i g_i(\boldsymbol{w}^*)=0$

4. $g^*_i(\boldsymbol{w}^*)\le 0$

5. $\boldsymbol{\mu}^*_i \ge 0$

从下文中可以看到，$\min f(\boldsymbol{w})$ 的最优解 $\boldsymbol{w}^*$ 等价于对偶问题 $\max_{\lambda\ge 0,\mu}\min_{w\in\Omega}(L(\boldsymbol{w},\lambda,\mu))$ 的最优解。

令 $h(\boldsymbol{\lambda},\boldsymbol{\mu})=\min_{\boldsymbol{w}\in\Omega}L(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})$ 设 $\boldsymbol{w}^*\in\Omega$ 是一个原问题的可行解，$\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*$ 是上述问题的一个可行解。则：

$$f(\boldsymbol{w}^*)\ge h(\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)$$

证明

对于 $\boldsymbol{w}\in\Omega$，从 $h(\boldsymbol{\lambda},\boldsymbol{\mu})$ 的定义出发，有：

$$h(\boldsymbol{\lambda},\boldsymbol{\mu})=\min_{\boldsymbol{u}\in\Omega}L(\boldsymbol{u},\boldsymbol{\lambda},\boldsymbol{\mu})\le L(\boldsymbol{w}^*,\boldsymbol{\lambda},\boldsymbol{\mu})=f(\boldsymbol{w}^*)+\lambda g(\boldsymbol{w}^*)+\mu h(\boldsymbol{w}^*)\le f(\boldsymbol{w}^*)$$

因为 w^* 的可行性意味着 $g(\boldsymbol{w}^*)\le 0; h(\boldsymbol{w}^*)=0$，同时 $(\boldsymbol{\lambda},\boldsymbol{\mu})$ 的可行性意味着 $\mu\ge 0$，

可见，对偶问题的上界由原问题给出：

$$\max_{\lambda,\mu\ge 0}\min_{w\in\Omega}(L(\boldsymbol{w},\lambda,\mu))\le \min f(w)_{g(w)\le 0,h(w)=0}$$

那么，如果 $f(\boldsymbol{w}^*)=h(\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)$ 其中 $\mu^*\ge 0$，并且 $g(\boldsymbol{w}^*)\le 0, h(\boldsymbol{w}^*)=0$，则 $\boldsymbol{w}^*$ 与 $\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*$ 分别是原问题和对偶问题的解，同时 $\mu^* g(\boldsymbol{w}^*)=0$

因为既然值是相等的，

$$h(\lambda,\mu)=\min_{u\in\Omega}L(u,\lambda,\mu)\le L(w,\lambda,\mu)=f(w)+\lambda g(w)+\mu h(w)\le f(w)$$

中的不等号全部可以替换为等号。

3.3.3.1 KKT 证明

3.3.3.1.1 必要性

1. $\frac{\partial L(w^*,\lambda^*,\mu^*)}{\partial w}=0$

可以理解为最小值点的梯度可以由限制函数在此点的梯度线性组合而得到。这个问题的证明请参考 [Nash 96]

2. $\frac{\partial L(w^*,\lambda^*,\mu^*)}{\partial \lambda}=h_i(w)=0$，显然成立。

3. $\mu^*_i g_i(\boldsymbol{w}^*)=0$

简单设 $m=0$，考虑公式 1 中由 $g_i(\boldsymbol{w}^*)$ 组成的那些梯度，最值得出现必然在可行域的边缘（或交叉点），这样，那些和这个边缘（交叉点）有关的 $g_i(\boldsymbol{w}^*)$ 其值必然是 0，这样 $\mu^*_i g_i(\boldsymbol{w}^*)=0$ 成立；对于那些和这个边缘（交叉点）无关的 $g_i(\boldsymbol{w}^*)$，必须令 $\mu^*_i=0$，因为公式 $\frac{\partial L(w^*,\lambda^*,\mu^*)}{\partial w}=0$ 中最小值点的梯度可以由限制函数在此点的梯度线性组合，“此点的梯度”只可能是那些“生成”此点的限制条件函数的梯度，还有一个重要原因此时 $L(\boldsymbol{w}^*,\boldsymbol{\lambda},\boldsymbol{\mu})$ 的值等于 $f(w^*)$ 的值，因此 $g_i(\boldsymbol{w}^*)\neq 0$ 时，必然 $\mu^*_i=0$。

4. $g^*_i(\boldsymbol{w}^*)\le 0$，显然成立。

5. $\mu^*_i\ge 0$，保证此点是最小值。

3.3.3.2 充分性

http://blog.pluskid.org/?p=702

3.3.4 基于KKT条件的公式推导

$\frac{\partial L(w^*,\lambda^*,\mu^*)}{\partial w}=0$ 与 $L(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})$ 凸函数说明 $L(w^*,\lambda^*,\mu^*)$ 是 $L(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})$ 最小值点，这样：

$$L(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)=f(\boldsymbol{w}^*)+\sum\lambda_i h_i(\boldsymbol{w}^*)+\sum\mu_i g_i(\boldsymbol{w}^*)$$

由于 $h_i(\boldsymbol{w}^*)=0$ $\mu_i g_i(\boldsymbol{w}^*)=0$ 因此

$$L(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)=f(\boldsymbol{w}^*)+\sum\lambda_i h_i(\boldsymbol{w}^*)+\sum\mu_i g_i(\boldsymbol{w}^*)=f(\boldsymbol{w}^*)$$

μ^* 是 $\max_{\mu\le 0}L(w,\lambda,\mu)$ 时得到的，此时由于 $\mu\ge 0$， $g_i(\boldsymbol{w})\le 0$ $h_i(\boldsymbol{w})=0$，所以 $\max_{\mu\ge 0}L(w,\lambda,\mu)$ 最大值时只能 $\mu_i g_i(\boldsymbol{w})=0$，于是：

$$\max_{\mu\ge 0}L(w,\lambda,\mu)=f(w)$$

进而：

$$\min f(w)\Leftrightarrow\min_{w,\lambda}\max_{\mu\ge 0}L(w,\lambda,\mu)$$

由于鞍点位置上：$\min_{w,\lambda}\max_{\mu\ge 0}L(w,\lambda,\mu)=\max_{\mu\ge 0}\min_{w,\lambda}L(w,\lambda,\mu)$，（最值只可能是鞍点），因此：

$$\min f(w)\Leftrightarrow\max_{\mu\ge 0}\min_{w,\lambda}L(w,\lambda,\mu)\Leftrightarrow\begin{cases}\max_{\mu\ge 0}L(w,\lambda,\mu)\\ s.t.\ \frac{\partial L(w,\lambda,\mu)}{\partial w}=0\ and\ \frac{\partial L(w,\lambda,\mu)}{\partial \lambda}=0\end{cases}$$

这样 SVM 求解公式：

$$\min_{\boldsymbol{w},b} \frac{1}{2} < \boldsymbol{w} \cdot \boldsymbol{w} >$$
$$s.t.\ y_i(< \boldsymbol{w} \cdot \boldsymbol{x_i} > +b) \geq 1$$

可以写成：

$$\max_{\alpha \geq 0} L(w,b,\alpha) = \frac{1}{2} < w \cdot w > - \sum \alpha_i [y_i(< w \cdot x_i > +b) - 1]$$
$$s.t.$$
$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum \alpha_i y_i x_i = 0 \Rightarrow w = \sum \alpha_i y_i x_i$$
$$\frac{\partial L(w,b,\alpha)}{\partial b} = -\sum \alpha_i y_i = 0 \Rightarrow \sum \alpha_i y_i = 0$$

将下面两项代入上述目标函数得到：

$$\begin{aligned}\max_{\alpha \geq 0} L(w,b,\alpha) &= \frac{1}{2} < w \cdot w > - \sum \alpha_i [y_i(< w \cdot x_i > +b) - 1] \\ &= \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j < x_i \cdot x_j > - \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j < x_i \cdot x_j > + \sum_{i=1}^{l} \alpha_i - b \sum_{i=1}^{l} \alpha_i y_i \\ &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j < x_i \cdot x_j >\end{aligned}$$

进而变为如下形式：

$$\max_{\alpha \geq 0} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j < x_i \cdot x_j >$$
$$s.t.$$
$$\sum \alpha_i y_i = 0$$

这样 SVM 的求解便是求解 α_i，在结果 $\alpha^* = < \alpha_1^*, \alpha_2^*, \alpha_3^*, ..., \alpha_l^* >$ 中，每个 $\alpha_i \geq 0$ 可以反映出当前样本对 SVM 分类器的影响权重，可以看到只有那些支持向量的点的值不为 0，其余点的 α_i 均等于 0.

这样求得 α_i 的值，依据 $\boldsymbol{w} = \sum \alpha_i y_i \boldsymbol{x}_i$ 便可以求得 w 的值，值 b 可以通过以下公式计算出：

我们肯定能分别找到一个支持向量的正例 x_+ 和负例 x_-，此时有：

$$< w \cdot x_+ > + b = +1$$
$$< w \cdot x_- > + b = -1$$

这样：

$$b = -\frac{1}{2}(< w \cdot x_+ > + < w \cdot x_- >)$$

这样便得到了 (w^*, b^*)，分类过程如下：

$$class(x) = sign[f(x)] = < w^* \cdot x > + b^*$$

$$\Leftrightarrow \sum_{i=1}^{l} \alpha_i y_i < x_i \cdot x > + b^*$$

$$\Leftrightarrow \sum_{i=1}^{l} \alpha_i y_i < x_i \cdot x > - \frac{1}{2}(< w \cdot x_+ > + < w \cdot x_- >)$$

$$\Leftrightarrow \sum_{i \in SV} \alpha_i y_i < x_i \cdot x > - \frac{1}{2}(< w \cdot x_+ > + < w \cdot x_- >)$$

其中 SV 代表支持向量点的集合。

可以看到超平面可以被那些支持向量的点表示出来。

3.3.4.1 有用中间推导公式：

$$\alpha_i^*[y_i(< w^* \cdot x_i > + b^*) - 1] = 0$$

此公式理解：

对于支持向量的点： $y_i(< w^* \cdot x_i > + b^*) = 1 \Leftrightarrow y_i(< w^* \cdot x_i > + b^*) - 1 = 0$ 对于非支持向量的点： $\alpha_i^* = 0$

由于支持向量点： x_i 满足：

$$y_i f(x_i) = y_i(< w^* \cdot x_i > + b^*) = y_i(\sum_{i=1}^{l} \alpha_i y_i < x_i \cdot x > + b^*) = 1$$

$$< w^* \cdot w^* > = (\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_i x_i) = \sum_{i,j} \alpha_i \alpha_j y_i y_i < x_i \cdot x_j >$$

$$= \sum_{i \in sv} \alpha_i y_i \sum_{j \in sv} \alpha_j y_i < x_i \cdot x_j >$$

$$= \sum_{i \in sv} \alpha_i (1 - y_j \times b)$$

$$= \sum_{i \in sv} \alpha_i - b \sum_{i \in sv} \alpha_i y_j$$

$$= \sum_{i \in sv} \alpha_i$$

其中：从前面得知 $\sum_{i \in sv} \alpha_i y_j = 0$ 这样，几何间隔 $\frac{1}{\| w^* \|}$ 可以改写为 $(\sum_{i \in sv} \alpha_i)^{-\frac{1}{2}}$

3.3.5 对偶目标函数与原始目标函数的差值

当 $\alpha = \alpha^{\wedge}$ 时，我们根据 $w = \sum \alpha_i y_i x_i$ 可以得到此时的 $w = w^{\wedge}$ ，因此：

$$
\begin{aligned}
&L(w^{\wedge},b,\alpha^{\wedge})-\frac{1}{2}\|w^{\wedge}\|^2\\
&=\frac{1}{2}<w\cdot w>-\sum\alpha_i[y_i(<w\cdot x_i>+b)-1]-\frac{1}{2}\|w^{\wedge}\|^2\\
&=-\sum\alpha_i[y_i(<w\cdot x_i>+b)-1]\\
&=\sum\alpha_i-\sum_{i,j}\alpha_i y_i y_j\alpha_j<x_i\cdot x_j>
\end{aligned}
$$

此差值称为可行间隙，可见只有在 α 取得最优值时，可行间隙才为 0.支持向量点的个数越少，其对应的超平面泛化能力越强。（有个误差公式 p88）

根据 $\alpha_i^*[y_i(<w^*\cdot x_i>+b^*)-1]=0$ 的计算过程，在软最大间隔 SVM 中有：

$$
\begin{aligned}
&\alpha_i[y_i(<x_i\cdot w>+b)-1+\xi_i]=0\\
&\xi_i(\alpha_i-C)=0
\end{aligned}
$$

因为此时有两个限制条件：

$$
y_i(<\boldsymbol{w}\cdot\boldsymbol{x_i}>+b)\geq 1-\xi_i \text{ 与 } 0\leq\alpha_i\leq C
$$

可以看出 $\alpha_i=C$ 的点是存在误差的点，当点直接跨到线另一侧时，ξ_i 是一个很大的数，这样的点很有可能是噪音点。当 $0\leq\alpha_i<C$ 时的点是那些完美支持向量的点。

因此，我们可以得出：

$$
\begin{aligned}
&\alpha_i=0\Leftrightarrow y_i f(x_i)\geq 1\\
&0<\alpha_i<C\Leftrightarrow y_i f(x_i)=1\\
&\alpha_i=C\Leftrightarrow y_i f(x_i)\leq 1
\end{aligned}
$$

换句话说，如果 α_i 不符合上述条件，那此时的解一定违反 KKT 条件。

软最大间隔 SVM 的可行间隙计算如下：

$$
\xi_i=\max(0,1-y_i(\sum_j y_j\alpha_j K(x_j,x_i)+b))
$$

$$
\begin{aligned}
&\frac{1}{2}<w\cdot w>+C\sum\xi_i-L(w,b,\xi,\alpha,r)\\
&=\sum_i\alpha_i[y_i(\sum y_j\alpha_j K(x_j,x_i)+b)-1+\xi_i]+\sum_i r_i\xi_i
\end{aligned}
$$

又因为：$\gamma_i=C-\alpha_i$，所以：

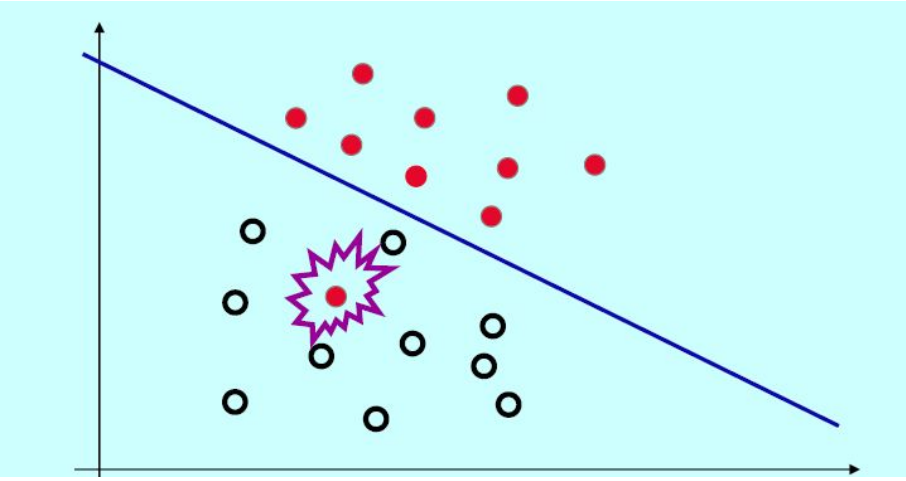
$$
\begin{aligned}
&\sum_i\alpha_i[y_i(\sum y_j\alpha_j K(x_j,x_i)+b)-1+\xi_i]+\sum_i r_i\xi_i\\
&=\sum_i\alpha_i[y_i(\sum_j y_j\alpha_j K(x_j,x_i))-1]+C\sum_i\xi_i\\
&=\sum_i\alpha_i-2W(\alpha)+C\sum_i\xi_i
\end{aligned}
$$

其中：

$$w(\alpha) = -\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j K(x_i \cdot x_j) + \sum_i \alpha_i$$

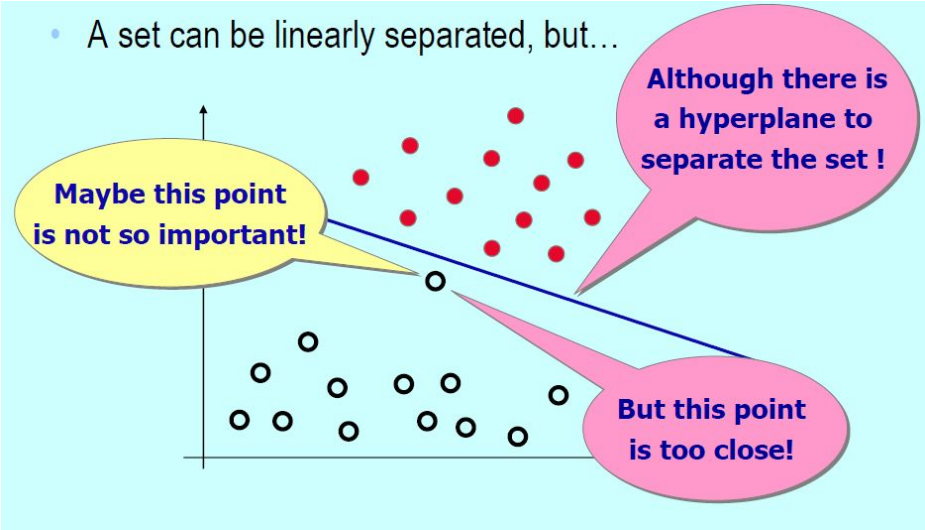
4. 线性不可分问题

二分类问题的不总是可以完全线性可分的，比如下图：



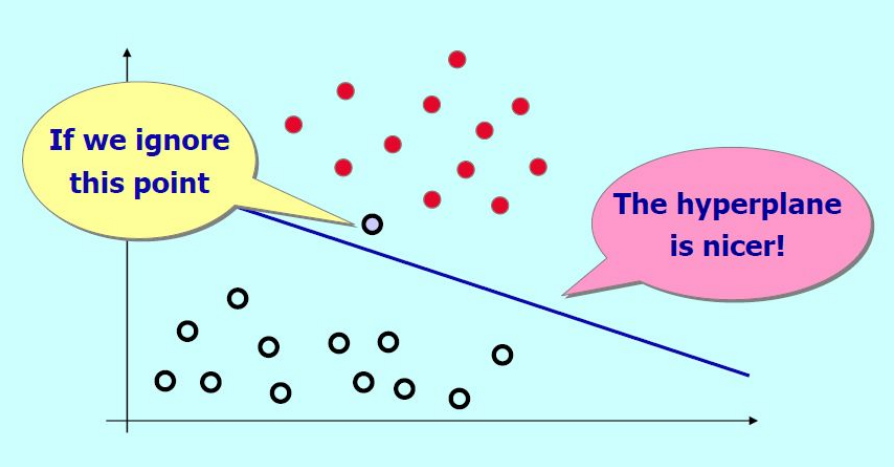
图[5] 线性不可分示意图

即使是线性可分的，可能由于噪音数据导致计算出一个很差的最大分割超平面，比如下图：



图[6]噪音数据对最大分割超平面的影响

如果我么可以这样选择最大分割超平面便会好很多，也就是忽略掉噪音数据，如下图所示：



图[7] 忽略噪音数据后计算出的最大分割超平面

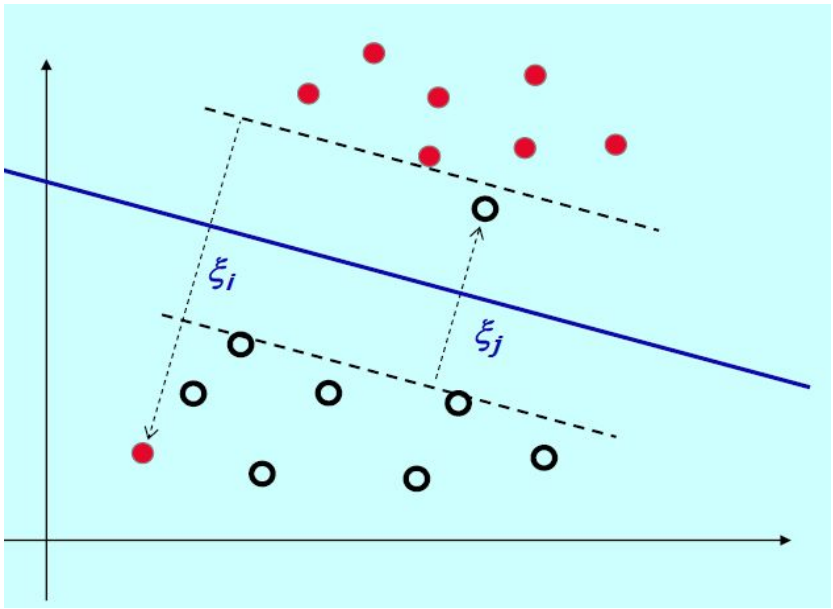
这样便有了软间隔 SVM，引入松弛变量（slack variable）ξ，限制条件有原始的：

$$y_i(<w\cdot x_i>+b)\geq 1$$

调整为：

$$y_i(<w\cdot x_i>+b)\geq 1-\xi_i$$
$$\xi_i\geq 0$$

其含义如下图所示：



图[8] 松弛变量的含义

这样软间隔 SVM 的最优化弓手由原始的：

$$\min_{w,b}\frac{1}{2}<\boldsymbol{w}\cdot\boldsymbol{w}>$$
$$s.t.\ y_i(<\boldsymbol{w}\cdot\boldsymbol{x}_i>+b)\geq 1$$

调整为：

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2} < \boldsymbol{w} \cdot \boldsymbol{w} > + C \sum_{i=1}^{l} \xi_i$$
$$s.t.\ y_i(< \boldsymbol{w} \cdot \boldsymbol{x_i} > + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

其中 C 是认为设定的数值，用来平衡最大间隔误差和训练误差。这样对偶问题由

$$\max_{\alpha \geq 0} L(w,b,\alpha) = \frac{1}{2} < w \cdot w > - \sum \alpha_i [y_i(< w \cdot x_i > + b) - 1]$$
$$s.t.$$
$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum \alpha_i y_i x_i = 0 \Rightarrow w = \sum \alpha_i y_i x_i$$
$$\frac{\partial L(w,b,\alpha)}{\partial b} = -\sum \alpha_i y_i = 0 \Rightarrow \sum \alpha_i y_i = 0$$

调整为：

$$\max_{\alpha \geq 0, \gamma \geq 0} L(w,b,\alpha,\xi) = \frac{1}{2} < w \cdot w > + C \sum \xi_i - \sum \alpha_i [y_i(< w \cdot x_i > + b) - (1 - \xi_i)] - \sum \gamma_i \xi_i$$
$$s.t.$$
$$\frac{\partial L(w,b,\alpha,\xi,\gamma)}{\partial w} = w - \sum \alpha_i y_i x_i = 0 \Rightarrow w = \sum \alpha_i y_i x_i$$
$$\frac{\partial L(w,b,\alpha,\xi,\gamma)}{\partial b} = -\sum \alpha_i y_i = 0 \Rightarrow \sum \alpha_i y_i = 0$$
$$\frac{\partial L(w,b,\alpha,\xi,\gamma)}{\partial \xi} = \sum C - \alpha_i - \gamma_i = 0 \Rightarrow C = \alpha_i + \gamma_i$$

代入上式得到：

$$\max_{\alpha \geq 0, \gamma \geq 0} L(w,b,\alpha,\xi)$$
$$= \frac{1}{2} < w \cdot w > + C \sum \xi_i - \sum \alpha_i [y_i(< w \cdot x_i > + b) - (1 - \xi_i)] - \sum \gamma_i \xi_i$$
$$= -\frac{1}{2} \sum_{i,j=1} \alpha_i \alpha_j y_i y_j < x_i \cdot x_j > + \sum_i \alpha_i$$

由于 $C = \alpha_i + \gamma_i$，且 $\alpha_i \geq 0$ $\gamma_i \geq 0$ 可以归结为 $0 \leq \alpha_i \leq C$ 所以上式

$$s.t.$$
$$\sum \alpha_i y_i = 0 \ and \ 0 \leq \alpha_i \leq C$$

因此最终得到：

$$\max_{\alpha\geq 0,} = -\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j < x_i \cdot x_j > + \sum_i \alpha_i$$

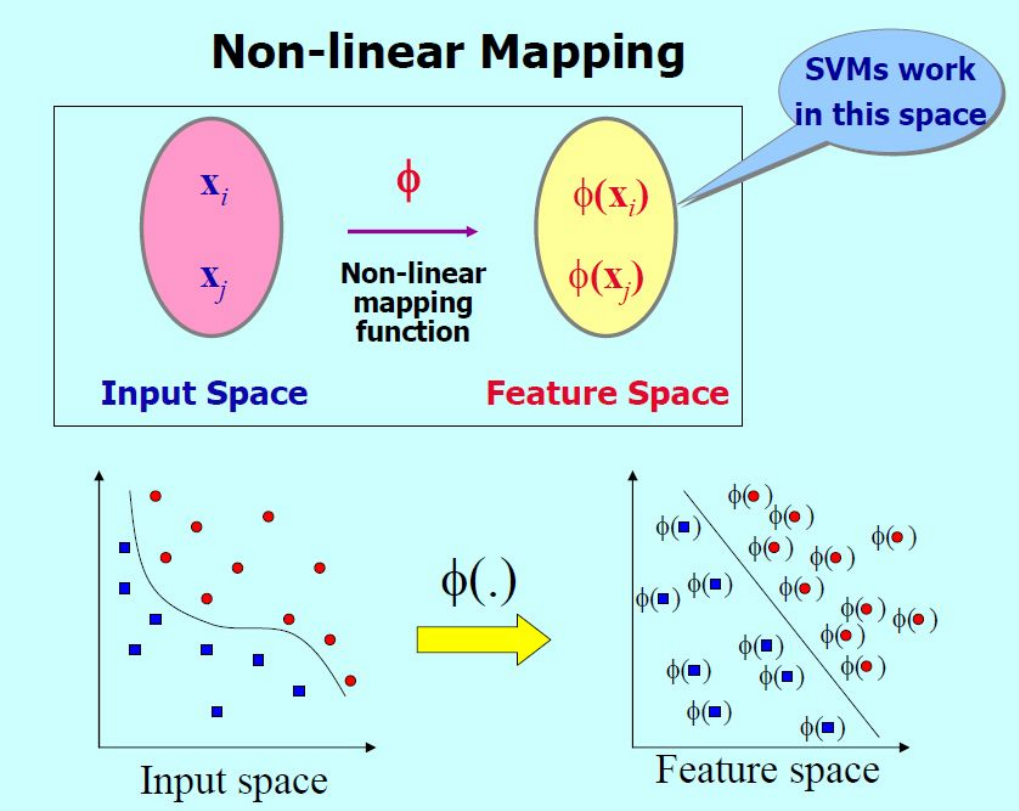
$$s.t.$$

$$\sum \alpha_i y_i = 0 \; and \; 0 \leq \alpha_i \leq C$$

和非软间隔 SVM 相比，公式只增加了一个限制条件 $0 \leq \alpha_i \leq C$ 软间隔 SVM 只是用来处理噪音数据，对于线性不可分的情况还需要另外一个工具，那就是核函数，其原理如下。

4.1 核函数

通过一个非线性映射函数 ϕ，将原空间中的样本 x 映射到一个新空间，在这个新空间它是线性可分的。如下所示：



图[9]映射函数含义

幸运的是，在计算过程中我们不需要显示的计算新的映射空间 x 的表示方法（其也是很难计算的，有时候新空间是无限维的），因为从计算公式中我们可以看到，其只需要计算 x_i, x_j 的内积 $< x_i \cdot x_j >$，这样我们如果直接能计算出 $\phi(x_i), \phi(x_j)$ 的内积 $< \phi(x_i) \cdot \phi(x_j) >$ 便可。此时计算公式调整为

$$\max_{\alpha\geq 0,} = -\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j < \phi(x_i) \cdot \phi(x_j) > + \sum_i \alpha_i$$

$$s.t.$$

$$\sum \alpha_i y_i = 0 \; and \; 0 \leq \alpha_i \leq C$$

引入核函数 $K(x_i,x_j)$，另其

$$K(x_i,x_j)=<\phi(x_i)\cdot\phi(x_j)>$$

注意：核函数必须满足条件 $K(x_i,x_j)=K(x_j,x_i)$

核函数具有如下 3 个好处：

1. 将线性不可分转换成线性可分。
2. 不用显式表示新空间。
3. SVM 可以隐式利用新空间。

4.1.1 常用的核函数

1. 多项式核函数

$$K(x_i,x_j)=(<x_i\cdot x_j>+c)^d$$

其中 c,d 是用户设定的参数

2. 高斯核函数（RBF 核函数）

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

其中 σ 是用户输入的参数

3. Sigmoid 核函数

$$K(x_i,x_j)=\tanh(\kappa<x_i\cdot x_j>+\Theta)$$

$$\tanh(a)=\frac{e^a-e^{-a}}{e^a+e^{-a}}$$

其中 κ 与 Θ 是用户输入的参数

这样 SVM 最终的计算公式如下：

训练：

$$\max_{\alpha\geq 0,}=-\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_i\alpha_i$$

$$s.t.$$

$$\sum\alpha_i y_i=0\ and\ 0\leq\alpha_i\leq C$$

预测:

$$f(x)=\text{sgn}[\sum \alpha_i y_i K(x_i,x)+b]$$

5. SMO 算法

5.1 简介

5.1.1 α 更新规则

每次迭代只优化两个点的最小子集。参考公式：

$$\max_{\alpha\geq 0,}=-\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_i \alpha_i$$

s.t.

$$\sum \alpha_i y_i=0 \; and \; 0\leq \alpha_i \leq C$$

其中 $\sum \alpha_i y_i=0 \; and \; 0\leq \alpha_i \leq C$ ，这两个条件在推理迭代更新中使用到了。

不妨设两个点分别为 α_1 与 α_2 ，由于需要满足 $\sum \alpha_i y_i=0$ ，所以：

$$\alpha_1^{\text{new}}\text{y}_1+\alpha_2^{\text{new}}\text{y}_2=\text{常数}=\alpha_1^{\text{old}}\text{y}_1+\alpha_2^{\text{old}}\text{y}_2$$

同时 $0\leq\alpha_1$，$\alpha_2\leq\text{C}$，这样就约束目标函数在一条直线上寻找最优解。我们不妨首先计算 α_2^{new} ，由于 $0\leq\alpha_2\leq\text{C}$ 与 $\alpha_1^{\text{new}}\text{y}_1+\alpha_2^{\text{new}}\text{y}_2=\text{常数}=\alpha_1^{\text{old}}\text{y}_1+\alpha_2^{\text{old}}\text{y}_2$， α_2^{new} 的取值范围可以缩小到：

$$\text{U}\leq\alpha_2^{\text{new}}\leq\text{V}$$

若 $\text{y}_1\neq\text{y}_2$ 则：

$$a_1^{new}y_1+a_2^{new}y_2=a_1^{old}y_1+a_2^{old}y_2$$

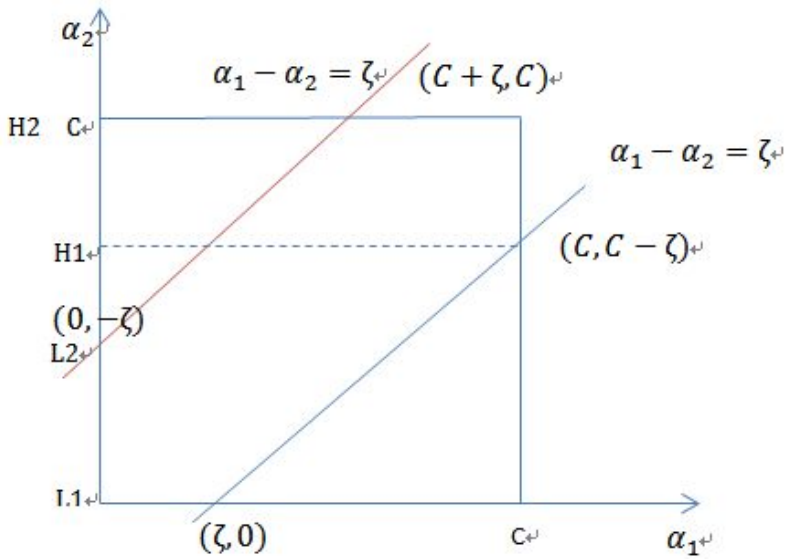
两边同乘以 y_2 得：

$$-a_1^{new}+a_2^{new}=-a_1^{old}+a_2^{old}$$
$$\Leftrightarrow$$
$$a_2^{new}=a_2^{old}-a_1^{old}+a_1^{new}$$
$$a_2^{new}-a_2^{old}+a_1^{old}=a_1^{new}$$
$$\Rightarrow$$
$$0\leq a_2^{new}-a_2^{old}+a_1^{old}\leq C_1$$
$$\Leftrightarrow$$
$$a_2^{old}-a_1^{old}\leq a_2^{new}\leq C_1+a_2^{old}-a_1^{old}$$

所以：

$$\text{U}=\max(0,\ \alpha_2^{\text{old}}-\alpha_1^{\text{old}})\ \text{V}=\min(\text{C}_2,\ \text{C}_1+\alpha_2^{\text{old}}-\alpha_1^{\text{old}})$$

参见下图：



若 $y_1=y_2$：

$$a_1^{new}y_1+a_2^{new}y_2=a_1^{old}y_1+a_2^{old}y_2$$

两边同乘以 y_2 得：

$$a_1^{new}+a_2^{new}=a_1^{old}+a_2^{old}$$
$$\Leftrightarrow$$
$$a_2^{new}=a_2^{old}+a_1^{old}-a_1^{new}$$
$$-a_2^{new}+a_2^{old}+a_1^{old}=a_1^{new}$$
$$\Rightarrow$$
$$0\le -a_2^{new}+a_2^{old}+a_1^{old}\le C_1$$
$$\Leftrightarrow$$
$$a_2^{old}+a_1^{old}-C_1\le a_2^{new}\le a_2^{old}+a_1^{old}$$

所以：

$$U=\max(0,\ \alpha_2^{old}+\alpha_1^{old}-C_1)$$

定义

$$E_i=f(x_i)-y_i=\sum a_j y_j k(x_j,x_i)-y_i \text{ 其中 } i=1 \text{ 或 } i=2$$

定义

$$\kappa=K(x_1,x_1)+K(x_2,x_2)-2K(x_1,x_2)=\|\phi(x_1)-\phi(x_2)\|^2$$

当 α_1，α_2 允许改变时，要使目标函数 $\max_{\alpha\ge 0,}=-\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_i\alpha_i$ 进一步最优，可以通过下面公式更新 α_1，α_2：

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

得到的 α_2^{new} 需要处于范围 $\mathrm{U} \le \alpha_2^{\mathrm{new}} \le \mathrm{V}$，因此需要做一步剪辑，如下：

$$\alpha_2^{new} = \begin{cases} V & if\, \alpha_2^{new} > V \\ a_2^{new} & if \mathrm{U} \le \alpha_2^{\mathrm{new}} \le \mathrm{V} \\ U & if\, \alpha_2^{\mathrm{new}} < \mathrm{U} \end{cases}$$

这样：

$$a_1^{new} = a_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})$$

这个更新步骤的证明如下：

定义

$$v_i = \sum_{j=3}^{t} y_j \alpha_j K(x_i, x_j) = f(x_i) - \sum_{j=1}^{2} y_j \alpha_j K(x_i, x_j) - b \text{其中} i = 1,2$$

目标公式

$$\max_{\alpha \ge 0,} = -\frac{1}{2} \sum_{i,j=1} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_i \alpha_i$$

中 $\alpha_{3\sim t}$ 都可以看做为常数，这样将 α_1，α_2 作为目标得到：

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

由于 $\sum \alpha_i^{old} \mathrm{y}_i = \sum \alpha_i^{new} y_i = 0$，这意味着： $\alpha_1 + s \times \alpha_2 = \text{常数} = \alpha_1^{\mathrm{old}} + \mathrm{s} \times \alpha_2^{\mathrm{old}} = \gamma$ 其中 $\mathrm{s} = \mathrm{y}_1 \mathrm{y}_2$，在计算出 α_2^{new} 这个方程可以用来计算 a_1^{new}，这样将 $\alpha_1 = \gamma - s \times \alpha_2$ 代入上式得到：

$$\begin{aligned} \mathrm{W}(\alpha_2) = {} & \gamma - s \times \alpha_2 - \alpha_2 - \frac{1}{2} K_{11} (\gamma - s \times \alpha_2)^2 - \frac{1}{2} K_{22} \alpha_2^2 - s \times K_{12} (\gamma - s \times \alpha_2) \alpha_2 \\ & - y_1 (\gamma - s \times \alpha_2) v_1 - y_2 \alpha_2 v_2 + \text{常数} \end{aligned}$$

由于驻点满足：

$$\frac{\partial \mathrm{W}(\alpha_2)}{\partial \alpha_2} = 0$$

因此：

$$\frac{\partial \mathrm{W}(\alpha_2)}{\partial \alpha_2} = 1 - s + s K_{11} (\gamma - s\alpha_2) - K_{22} \alpha_2 + K_{12} \alpha_2 - s K_{12} (\gamma - s\alpha_2) + y_2 v_1 - y_2 v_2 = 0$$

注意 y_1, y_2 的值只能是 1 与-1，公式转换的时候可以被利用到。

得出：

$$\begin{aligned} (K_{22} - 2K_{12} + K_{11}) \alpha_2 &= 1 - s + s K_{11} \gamma - s K_{12} \gamma + y_2 v_1 - y_2 v_2 \\ &= y_2 (y_2 - y_1 + y_1 K_{11} \gamma - y_1 K_{12} \gamma + v_1 - v_2) \end{aligned}$$

进而：

$$\alpha_2^{new}\kappa y_2 = y_2 - y_1 + \gamma y_1(K_{11} - K_{12}) + v_1 - v_2$$
$$= y_2 - y_1 + f(x_1) - \sum_{j=1}^{2} y_j \alpha^{old}{}_j K_{1j} + \gamma y_1 K_{11} - f(x_2) + \sum_{j=1}^{2} y_j \alpha^{old}{}_j K_{2j} - \gamma y_1 K_{12}$$
$$= y_2 - y_1 + f(x_1) - f(x_2) + y_2\alpha_2^{old}K_{11} - y_2\alpha_2^{old}K_{12} + y_2\alpha_2^{old}K_{22} - y_2\alpha_2^{old}K_{12}$$
$$= y_2\alpha_2^{old}\kappa + (f(x_1) - y_1) - (f(x_2) - y_2)$$

最终得到：

$$a_2^{new} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\kappa}$$

最后的到的 a_2^{new} 要根据：

$$\alpha_2^{new} = \begin{cases} V & if\ \alpha_2^{new} > V \\ a_2^{new}\ if \mathrm{U} \le \alpha_2^{\mathrm{new}} \le \mathrm{V} \\ U & if\ \alpha_2^{\mathrm{new}} < \mathrm{U} \end{cases}$$

进行剪辑，确保 a_2^{new} 的值在要求范围内。

在特殊情况下，κ 可能不为正，如果核函数 K 不满足 Mercer 定理，那么目标函数可能变得非正定，κ 可能出现负值。即使 K 是有效的核函数，如果训练样本中出现相同的特征 x，那么 κ 仍有可能为 0。SMO 算法在 κ 不为正值的情况下仍有效。为保证有效性，我们可以推导出 κ 就是 $W(\alpha_2)$ 的二阶导数，$\kappa < 0$，$W(\alpha_2)$ 没有极小值，最小值在边缘处取到（类比 $y = -x^2$），$\kappa = 0$ 时更是单调函数了，最小值也在边缘处取得，而 α_2 的边缘就是 U 和 V。这样将 $\alpha_2 = U$ 和 $\alpha_2 = V$ 分别代入 $W(\alpha_2)$ 中即可求得 $W(\alpha_2)$ 的最小值，相应的 $\alpha_2 = U$ 还是 $\alpha_2 = V$ 也可以知道了。具体计算公式如下：

$$f_1 = y_1(E_1 + b) - \alpha_1 K(\vec{x}_1, \vec{x}_1) - s\alpha_2 K(\vec{x}_1, \vec{x}_2),$$
$$f_2 = y_2(E_2 + b) - s\alpha_1 K(\vec{x}_1, \vec{x}_2) - \alpha_2 K(\vec{x}_2, \vec{x}_2),$$
$$L_1 = \alpha_1 + s(\alpha_2 - L),$$
$$H_1 = \alpha_1 + s(\alpha_2 - H),$$
$$\Psi_L = L_1 f_1 + L f_2 + \tfrac{1}{2} L_1^2 K(\vec{x}_1, \vec{x}_1) + \tfrac{1}{2} L^2 K(\vec{x}_2, \vec{x}_2) + sLL_1 K(\vec{x}_1, \vec{x}_2),$$
$$\Psi_H = H_1 f_1 + H f_2 + \tfrac{1}{2} H_1^2 K(\vec{x}_1, \vec{x}_1) + \tfrac{1}{2} H^2 K(\vec{x}_2, \vec{x}_2) + sHH_1 K(\vec{x}_1, \vec{x}_2).$$

5.1.2 b 值的更新规则

由于 $f(x_i) = \sum \alpha_j y_j K(x_j, x_i) + b$ 所以有

$$f(x_i) = \sum \alpha_j y_j K(x_j, x_i) + b$$
$$\Leftrightarrow$$
$$b = \alpha_1 y_1 K_{11} + \alpha_2 y_2 K_{21} - y_1 + \text{constant} \quad \text{其中} f(x_i) = y_1$$
$$b^{new} = \alpha^{new}{}_1 y_1 K_{11} + \alpha^{new}{}_2 y_2 K_{21} - y_1 + \text{constant} \quad \text{其中} f(x_i) = y_1$$
$$b = \alpha_1 y_1 K_{12} + \alpha_2 y_2 K_{22} - y_2 + \text{constant} \quad \text{其中} f(x_2) = y_2$$
$$b^{new} = \alpha^{new}{}_1 y_1 K_{12} + \alpha^{new}{}_2 y_2 K_{22} - y_2 + \text{constant} \quad \text{其中} f(x_2) = y_2$$

两两想减得：

$$b^{new} = y_1(\alpha_1^{new} - \alpha_1)K_{11} + y_2(\alpha_2^{new} - \alpha_2)K_{21} + b$$
$$b^{new} = y_1(\alpha_1^{new} - \alpha_1)K_{12} + y_2(\alpha_2^{new} - \alpha_2)K_{22} + b$$

进而改造成：

$$b_1^{new} = E_1 + y_1(\alpha_1^{new} - \alpha_1)K_{11} + y_2(\alpha_2^{new} - \alpha_2)K_{21} + b$$
$$b_2^{new} = E_2 + y_1(\alpha_1^{new} - \alpha_1)K_{12} + y_2(\alpha_2^{new} - \alpha_2)K_{22} + b$$

可见：如果 $0 < \alpha_1 < C$ 并且 $0 < \alpha_2 < C$ 则 $E_1 = 0, E_2 = 0$，上述两种方法计算出的 b^{new} 值相同。如果只有其中一个值处于 $0 < \alpha_i < C, i = 1,2$，则 b^{new} 取值对应的那个，因为另一个计算的 b^{new} 要大。如果 $a_i = 0$ 或 $a_i = \text{C}$ i=1,2，则处于 b_1^{new} 与 b_2^{new} 之间的任何数值都不违反下面条件，一般取值 $\frac{1}{2}(b_1^{new} + b_2^{new})$

我们可以得出：

$$\alpha_i = 0 \Leftrightarrow y_i f(x_i) \geq 1$$
$$0 < \alpha_i < C \Leftrightarrow y_i f(x_i) = 1$$
$$\alpha_i = C \Leftrightarrow y_i f(x_i) \leq 1$$

违背上述条件也就是违背了 KKT 条件。

5.2 LibSVM 计算阈值 b 的方法

LibSVM 并不在每次更新 α 过程中计算 b 值，而是在所有 α 计算完毕后在计算 b 值，对于完美支持向量的点 x_m，有 $0 \leq \alpha_m \leq C$，并且 $f(x_m) = \sum_i \alpha_i y_i K(x_i, x_m) + b = y_m$ 其中 $y_m = \pm 1$，这样

$$\sum y_m y_i \alpha_i K(x_i, x_m) + b \times y_m = y_m \times y_m$$
$$\Leftrightarrow$$
$$\sum y_m y_i \alpha_i K(x_i, x_m) + b \times y_m = 1$$
$$\Leftrightarrow$$
$$b = -\sum y_m y_i \alpha_i K(x_i, x_m) + 1$$

这样对任意完美支持向量的点都可以计算出一个 b 值出来，最后取其平均值作为最终的 b 值。

5.3 LibSVM 计算 α 的方法

$$\alpha_2^{\text{new}} = \alpha_2^{\text{old}} + \frac{\text{y}_2(\text{E}_1 - \text{E}_2)}{\kappa}$$
$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + \frac{\text{y}_1(\text{E}_2 - \text{E}_1)}{\kappa}$$

5.4 SMO 算法停止条件

原始目标函数值与对偶目标函数值的间隙只有在驻点（优化点）才会消失，这个间隙称为可行间隙，其计算方式如下：

原始目标 - 对偶目标为

$$\begin{aligned}&\frac{1}{2}<w\cdot w>+C\sum\xi_i-L(w,b,\xi,\alpha,r)\\&=\sum_i\alpha_i[y_i(\sum y_j\alpha_jK(x_j,x_i)+b)-1+\xi_i]+\sum_i r_i\xi_i\end{aligned}$$

又因为：$\gamma_i=C-\alpha_i$，所以：

$$\begin{aligned}&\sum_i\alpha_i[y_i(\sum y_j\alpha_jK(x_j,x_i)+b)-1+\xi_i]+\sum_i r_i\xi_i\\&=\sum_i\alpha_i[y_i(\sum_j y_j\alpha_jK(x_j,x_i))-1]+C\sum_i\xi_i\\&=\sum_i\alpha_i-2W(\alpha)+C\sum_i\xi_i\end{aligned}$$

其中

$$w(\alpha)=-\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_jy_iy_jK(x_i\cdot x_j)+\sum_i\alpha_i$$

因而其停止条件可以用如下公式衡量：

$$\begin{aligned}&\frac{\text{原始目标-对偶目标}}{\text{原始目标+1}}\\&=\frac{\sum_i\alpha_i-2W(\alpha)+C\sum_i\xi_i}{W(\alpha)+\sum_i\alpha_i-2W(\alpha)+C\sum_i\xi_i+1}\\&=\frac{\sum_i\alpha_i-2W(\alpha)+C\sum_i\xi_i}{\sum_i\alpha_i-W(\alpha)+C\sum_i\xi_i+1}\end{aligned}$$

检查它是否小于某个值，比如 0.001。

5.5 LibSVM 选择 α 的方法与停止条件

对于那些完美支持向量的点 x_i：

$$
\begin{aligned}
&f(x_i) = y_i \text{ 其中} y_i = \pm 1 \\
&\Leftrightarrow \\
&\sum_m (\alpha_m \times y_m \times < x_m, x_i >) + b = y_i \\
&\Leftrightarrow \\
&\sum_m (\alpha_m \times y_m \times y_i \times < x_m, x_i >) + b \times y_i = 1 \\
&\Leftrightarrow \\
&\sum_m (\alpha_m \times y_m \times y_i \times < x_m, x_i >) - 1 + b \times y_i = 0 \\
&\Leftrightarrow \\
&G_i + b \times y_i = 0 \\
&\Leftrightarrow \\
&b = -y_i G_i
\end{aligned}
$$

其中 G_i 的定义在代码中。

进而我们可以看到。定义：

$$
\begin{aligned}
I_{up} &= \{i \mid \alpha_i < C; y_i = 1 \text{ or } \alpha_i > 0; y_i = -1\} \\
I_{low} &= \{i \mid \alpha_i < C; y_i = -1 \text{ or } \alpha_i > 0; y_i = 1\}
\end{aligned}
$$

$$
\begin{aligned}
-y_i G_i &<= b; i \in I_{up} \\
-y_i G_i &>= b; i \in I_{low}
\end{aligned}
$$

LibSVM 先在 I_{up} 中选择一个使 $-y_i G_i$ 值最大的 x_i，这是的最大值是最近接实际的 b 值的，然后选择 x_j，根据公式：

$$
\alpha_i^{new} = \alpha_i^{old} + \frac{y_i (E_j - E_i)}{\kappa}
$$

要使 α_i^{new} 相对于 α_i^{old} 变化的最大也就是最大化 $|\frac{y_i (E_j - E_i)}{\kappa}|$。

当 $-y_i G_i$ 和 $-y_j G_j$ 值很接近时，算法终止，因为这两个值都是一次一次在接近真实的 b 值，真实的 b 值得到了，算法自然停止了。

5.6 总结

SMO 算法每步选择两个元素 α_i 和 α_j，共同优化，在其它参数固定的前提下，找到这两个参数元素的最优值，并更新相应的 α 向量。两个点的优化可以有解析解，并且没有矩阵操作，不需要再内存中存储核矩阵，核矩阵的引入可以提高速度，但会增加空间复杂度。

SMO 算法虽然每次都选择两个参数进行优化，这两个参数起其中至少有一个参数是违背 KKT 条件的，经过一次这样的优化步骤，优化目标就会更优一步。依据 Osuna 的定理[3]，这个过程可以保证收敛于最优解。为了加快收敛速度，SMO 采用了一些启发式的规则来选择需要优化的两个参数。如下：

SMO 采用了两种启发式规则，分别应用在两个元素 α_1 和 α_2 上。应用在 α_1 上的启发规则体现在 SMO 算法的外围循环上。外围循环首先遍历整个训练集合判断当前点是不是违背如下的 KKT 条件。

---

[3] http://fourier.eng.hmc.edu/e161/lectures/gaussianprocess/node8.html

正定函数：除零点外恒为正数的标量函数

$$\alpha_i = 0 \Leftrightarrow y_i f(x_i) \geq 1$$
$$0 < \alpha_i < C \Leftrightarrow y_i f(x_i) = 1$$
$$\alpha_i = C \Leftrightarrow y_i f(x_i) \leq 1$$

如果找到一个违背此 KKT 条件的点，它变被选择用来做优化。此轮遍历整个训练数据结束后，外围循环将会缩小遍历范围，此时它只遍历那些α值非 0 且非 C 的点集，同样检查当前点是不是违背上述的 KKT 条件，外围循环会循环到这些点全部都遵循 KKT 条件为止，此后外围循环会回到开始状态继续遍历整个训练集合。这样外围循环在“一次遍历整个训练数据集合”与“若干次遍历α值非 0 且非 C 的点集”之间转换直到算法结束。这个启发规则保证算法优先处理那些最可能违背 KKT 条件的点，即α值非 0 且非 C 的点，因为这样的点，即使在其它点的α值发生改变后，其α值也很可能不会改动。所以 SMO 算法首先保证那些α值非 0 且非 C 的点集遵循 KKT 条件，再扫描整个训练数据集。

Notice that the KKT conditions are checked to be within of fulfillment. Typically, is set to be 10-3. Recognition systems typically do not need to have the KKT conditions fulfilled to high accuracy: it is acceptable for examples on the positive margin to have outputs between 0.999 and 1.001. The SMO algorithm (and other SVM algorithms) will not converge as quickly if it is required to produce very high accuracy output.

一旦第一个需要优化α的点找到，SMO会依据另外一条启发式规则选择第二个需要优化的点。此条启发规则要最大化第一个点α值的优化步伐，依据公式：

$$a^{new} = \alpha^{old} + \frac{y_2(E_1 - E_2)}{\kappa}$$

由于计算K比较耗时，SMO采用最大化$|E_1 - E_2|$的办法来最大化第一个点α值的优化步伐。计算中SMO会缓存那些α值非0且非C的点的E值。也就是说当$E1$正值时，SMO选择最小的$E2$，当为负值时，SMO选择最大的$E2$. 在一些非常规的情况下，SMO依据上述第二条启发规则选择出来的第二个点并不能优化目标函数，比如这两个点对应的特征向量是完全一样的，此时会造成目标函数为半正定的。在这种情况下，SMO使用第二条启发规则下的一条分级规则来寻找一对能优化目标函数的点，它是这样的，如果第二条启发规则找到的点不能给优化目标函数带来正面效果，SMO开始遍历所有α值非0且非C的点，搜索出一个可以给优化目标函数带来正面效果的点，如果不能找到任何一个这样的点，SMO开始遍历训练集中所有的点，直到找到一个合适的点。这两种遍历都是起始于一个随机位置，为了避免SMO偏向于训练数据开始位置的点。如果最终也没能搜索到一个符合条件的点，此时第一个被选中的参数将被跳过，SMO继续依据第一条启发规则选择出下一个被优化的第一个点。

5.7 伪代码

```
target = desired output vector
point = training point matrix
procedure takeStep(i1,i2)
        if (i1 == i2) return 0
        alph1 = Lagrange multiplier for i1
        y1 = target[i1]
        E1 = SVM output on point[i1] – y1 (check in error cache)
        s = y1*y2
        Compute L, H
        // case 1: C=0 此时，ai=0.
        // case 2: a1==0
        if (L == H) return 0
        k11 = kernel(point[i1],point[i1])
        k12 = kernel(point[i1],point[i2])
```

```
        k22 = kernel(point[i2],point[i2])
        eta = k11+k22-2*k12
        if (eta > 0){
                a2 = alph2 + y2*(E1-E2)/eta
                if (a2 < L) a2 = L
                else if (a2 > H) a2 = H
        }else{
                Lobj = objective function at a2=L
                Hobj = objective function at a2=H
                if (Lobj < Hobj-eps)  a2 = L
                else if (Lobj > Hobj+eps)  a2 = H
                else  a2 = alph2
        }
        if (|a2-alph2| < eps*(a2+alph2+eps))  return 0
        a1 = alph1+s*(alph2-a2)
        Update threshold to reflect change in Lagrange multipliers //b
        Update weight vector to reflect change in a1 & a2, if SVM is linear
        Update error cache using new Lagrange multipliers
        Store a1 in the alpha array
        Store a2 in the alpha array
        return 1
endprocedure
procedure examineExample(i2)
        y2 = target[i2]
        alph2 = Lagrange multiplier for i2
        E2 = SVM output on point[i2] – y2 (check in error cache)
        r2 = E2*y2 //f(x2)*y2-1
        if ((r2 < -tol && alph2 < C) || (r2 > tol && alph2 > 0)){
                if (number of non-zero & non-C alpha > 1){
                        i1 = result of second choice heuristic
                        if takeStep(i1,i2) return 1
                }

                loop over all non-zero and non-C alpha, starting at a random point{
                        i1 = identity of current alpha
                        if   takeStep(i1,i2)  return 1
                }
                loop over all possible i1, starting at a random point{
                        i1 = loop variable
                        if (takeStep(i1,i2)   return 1
                }
```

```
        }
        return 0
endprocedure
main routine:
        numChanged = 0;
        examineAll = 1;
        while (numChanged > 0 || examineAll){
                numChanged = 0;
                if (examineAll)
                        loop I over all training examples
                                numChanged += examineExample(I)
                else
                        loop I over examples where alpha is not 0 & not C
                                numChanged += examineExample(I)
                if (examineAll == 1)  examineAll = 0
                else if (numChanged == 0)  examineAll = 1
        }
```

Epsilon-SVR

1. 简介

给定训练数据 $\{(x_1,y_1),...(x_n,y_n)\}\in\chi\times R$， $\varepsilon-\text{SVR}$ 的目标是寻找一个函数 $f(x)$，使其对于 x_i，其值与真实值 y_i 的差值不超过 ε，也就是 $|y_i-f(x_i)|<\varepsilon$

$$f(x)=<w\cdot x>+b$$
$$w\in\chi,b\in R$$

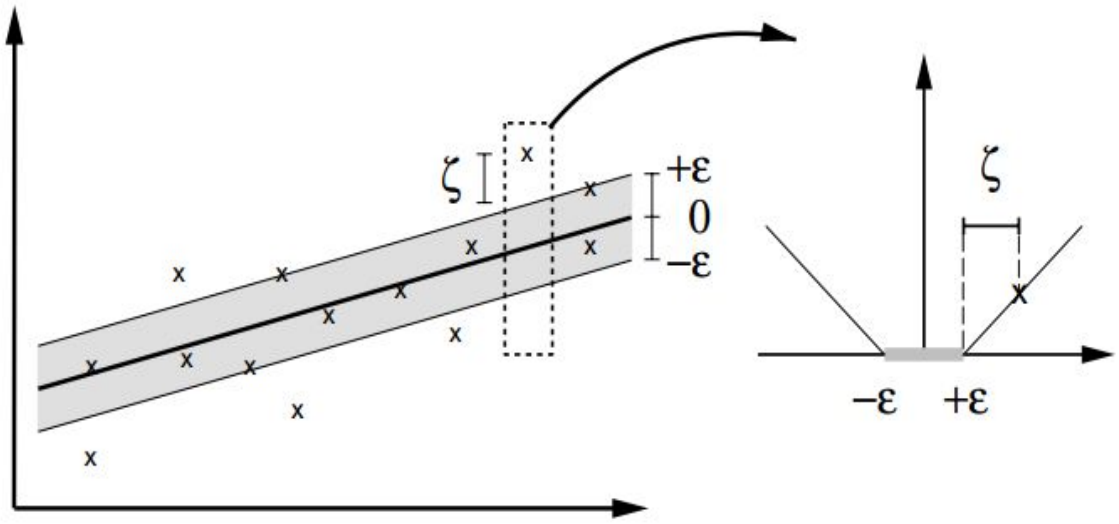
求解 w 一个常见的方法是最小化 w 的 2-范数 $\|w\|^2$，同时满足约束条件，也就是：

$$\min(\frac{\|w\|^2}{2})$$
$$\text{s.t.}\begin{cases}y_i-<w\cdot x_i>-b\le\varepsilon\\<w\cdot x_i>+b-y_i\le\varepsilon\end{cases}$$

为了适应现实中的噪音数据，类似于软间隔超平面，我们同样可以引入松弛变量 ξ_i,ξ_i^*，这样：

$$\min(\frac{\|w\|^2}{2}+C\sum_{i=0}^{n}(\xi_i+\xi_i^*))$$
$$\text{s.t.}\begin{cases}y_i-<w\cdot x_i>-b\le\varepsilon+\xi_i\\<w\cdot x_i>+b-y_i\le\varepsilon+\xi_i^*\\\xi_i,\xi_i^*\ge0\end{cases}$$

其中常数 $C>0$，用于在控制松弛变量所起的作用的大小。如下图



其中对于阴影部分的点$|\xi|=0$，其对确定 w 不起作用，对于其余部分的点$|\xi|>0$，这些点是那些完美支持向量的点和存在误差的点。

2. 公式推导

根据拉格朗日乘子法，公式

$$\min(\frac{\|\mathrm{w}\|^2}{2})$$

$$\text{s.t.}\begin{cases}\mathrm{y_i}-<\mathrm{w}\cdot\mathrm{x_i}>-\mathrm{b}\le\varepsilon\\<\mathrm{w}\cdot\mathrm{x_i}>+\mathrm{b}-\mathrm{y_i}\le\varepsilon\end{cases}$$

可以改写为

$$\min(\mathrm{L})=\frac{1}{2}<\mathrm{w}\cdot\mathrm{w}>$$

$$+\mathrm{C}\sum_{\mathrm{i}}^{\mathrm{n}}(\xi_{\mathrm{i}}+\xi_{\mathrm{i}}^{*})+\sum_{\mathrm{i=1}}^{\mathrm{n}}\alpha_{\mathrm{i}}(\varepsilon+\xi_{\mathrm{i}}-\mathrm{y_i}+<\mathrm{w}\cdot\mathrm{x_i}>+\mathrm{b})+\sum_{\mathrm{i=1}}^{\mathrm{n}}\alpha_{\mathrm{i}}^{*}(\varepsilon+\xi_{\mathrm{i}}^{*}+\mathrm{y_i}-<\mathrm{w}\cdot\mathrm{x_i}>-\mathrm{b})+\sum_{\mathrm{i=1}}^{\mathrm{n}}(\eta_{\mathrm{i}}\times\xi_{\mathrm{i}}+\eta_{\mathrm{i}}^{*}\times\xi_{\mathrm{i}}^{*})$$

$$\text{s.t.}\alpha_{\mathrm{i}},\alpha_{\mathrm{i}}^{*},\eta_{\mathrm{i}},\eta_{\mathrm{i}}^{*}\ge 0$$

因为：

$$\frac{\partial\mathrm{L}}{\partial\mathrm{b}}=\sum_{\mathrm{i=1}}^{\mathrm{n}}(\alpha_{\mathrm{i}}-\alpha_{\mathrm{i}}^{*})=0$$

$$\frac{\partial\mathrm{L}}{\partial\mathrm{w}}=\mathrm{w}+\sum_{\mathrm{i=1}}^{\mathrm{n}}(\alpha_{\mathrm{i}}-\alpha_{\mathrm{i}}^{*})\,\mathrm{x_i}=0$$

$$=>\mathrm{w}=\sum_{\mathrm{i=1}}^{\mathrm{n}}(\alpha_{\mathrm{i}}^{*}-\alpha_{\mathrm{i}})\,\mathrm{x_i}$$

$$\frac{\partial\mathrm{L}}{\partial\xi_{\mathrm{i}}}=\mathrm{C}+\alpha_{\mathrm{i}}+\eta_{\mathrm{i}}=0$$

$$\frac{\partial\mathrm{L}}{\partial\xi_{\mathrm{i}}^{*}}=\mathrm{C}+\alpha_{\mathrm{i}}^{*}+\eta_{\mathrm{i}}^{*}=0$$

带入上式得：

$$\min(L)=\frac{1}{2}\sum_{i,j}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)<x_i\cdot x_j>+\varepsilon\sum_i(\alpha_i+\alpha_i^*)-y_i\sum_i(\alpha_i-\alpha_i^*)$$

$$\Leftrightarrow$$

$$\max(L)=-\frac{1}{2}\sum_{i,j}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)<x_i\cdot x_j>-\varepsilon\sum_i(\alpha_i+\alpha_i^*)+y_i\sum_i(\alpha_i-\alpha_i^*)$$

s.t.

$$\begin{cases}\sum_i(\alpha_i-\alpha_i^*)=0\\ 0\le\alpha_i,\alpha_i^*\le C\end{cases}$$

这样：

$$f(x)=\sum_i(\alpha_i^*-\alpha_i)<x_i\cdot x>+b$$

根据 KKT 条件我们可以得知：

1) 对于 $0\le\alpha_i,\alpha_i^*\le 0$ 的点是那些完美支持向量的点。

2) $\alpha_i,\alpha_i^*=0$ 是那些不起作用的点，即阴影部分中的点。

3) $\alpha_i,\alpha_i^*=C$ 是那些存在误差的点。

对于那些完美支持向量的点，有 KKT 调节得出：

$$\alpha_i(\varepsilon+\xi_i-y_i+<w\cdot x_i>+b)=0$$
$$\alpha_i^*(\varepsilon+\xi_i^*+y_i-<w\cdot x_i>-b)=0$$
$$(C-\alpha_i)\xi_i=0$$
$$(C-\alpha_i^*)\xi_i^*=0$$

因为 $0\le\alpha_i,\alpha_i^*\le 0$，此时：

$$\varepsilon+\xi_i-y_i+<w\cdot x_i>+b=0$$
$$\varepsilon+\xi_i^*+y_i-<w\cdot x_i>-b=0$$
$$\xi_i=0$$
$$\xi_i^*=0$$

我们可以看到 b 值可以这样计算得到：

$$b=y_i-<w\cdot x_i>-\varepsilon\ \text{ for }\ 0<\alpha_i<C$$
$$b=y_i-<w\cdot x_i>+\varepsilon\ \text{ for }\ 0<\alpha_i^*<C$$

3. 求解过程

从前面我们可以得知，C-SVM 的求解公式为：

$$\max_{\alpha\geq 0,} = -\frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j K(x_i,x_j) + \sum_i \alpha_i$$

$$\Leftrightarrow$$

$$\min_{\alpha\geq 0,} = \frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \sum_i \alpha_i$$

$$s.t.$$

$$\sum \alpha_i y_i = 0 \; and \; 0 \leq \alpha_i \leq C$$

从下面的推导过程我们可以看出，$\varepsilon - \mathrm{SVR}$ 可以转换成 C-SVM 的样子，进而使用上面的方法求得最优值。只要能转换成如下 C-SVM 公式的格式，就可以转换成求解 C-SVM 问题.

$$\min_{\alpha\geq 0,} = \frac{1}{2}\alpha^T Q\alpha + p^T\alpha$$

$$s.t.$$

$$y^T\alpha = 0$$

其中 Q 是一个方阵，在 C-SVM 中其为 $\mathrm{n}\times\mathrm{n}$ 的方阵，$\mathrm{Q[i][j]} = \mathrm{y[i]}\times\mathrm{y[j]}\times<\mathrm{x[i]}\cdot\mathrm{x[j]}>$。

由：

$$\min(\mathrm{L}) = \frac{1}{2}\sum_{\mathrm{i,j}}(\alpha_\mathrm{i}-\alpha_\mathrm{i}^*)(\alpha_\mathrm{j}-\alpha_\mathrm{j}^*)<\mathrm{x_i}\cdot\mathrm{x_j}> + \varepsilon\sum_\mathrm{i}(\alpha_\mathrm{i}+\alpha_\mathrm{i}^*) - \mathrm{y_i}\sum_\mathrm{i}(\alpha_\mathrm{i}-\alpha_\mathrm{i}^*)$$

$$\Leftrightarrow$$

$$\max(\mathrm{L}) = -\frac{1}{2}\sum_{\mathrm{i,j}}(\alpha_\mathrm{i}-\alpha_\mathrm{i}^*)(\alpha_\mathrm{j}-\alpha_\mathrm{j}^*)<\mathrm{x_i}\cdot\mathrm{x_j}> - \varepsilon\sum_\mathrm{i}(\alpha_\mathrm{i}+\alpha_\mathrm{i}^*) + \mathrm{y_i}\sum_\mathrm{i}(\alpha_\mathrm{i}-\alpha_\mathrm{i}^*)$$

s.t.

$$\begin{cases}\sum_\mathrm{i}(\alpha_\mathrm{i}-\alpha_\mathrm{i}^*) = 0 \\ 0\leq\alpha_\mathrm{i},\alpha_\mathrm{i}^*\leq\mathrm{C}\end{cases}$$

推出：

$$\min(\mathrm{L}) = \frac{1}{2}[(\boldsymbol{\alpha}^*)^\mathrm{T},\boldsymbol{\alpha}]\begin{bmatrix}\mathrm{Q} & ,-\mathrm{Q}\\ -\mathrm{Q}, & \mathrm{Q}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}^*\\ \boldsymbol{\alpha}\end{bmatrix} + [\varepsilon\boldsymbol{e}^\mathrm{T}-\boldsymbol{y}^\mathrm{T},\varepsilon\boldsymbol{e}^\mathrm{T}+\boldsymbol{y}^\mathrm{T}]\begin{bmatrix}\boldsymbol{\alpha}^*\\ \boldsymbol{\alpha}\end{bmatrix}$$

s.t.

$$\boldsymbol{y}^\mathrm{T}\begin{bmatrix}\boldsymbol{\alpha}^*\\ \boldsymbol{\alpha}\end{bmatrix} = 0, 0\leq\alpha_\mathrm{i},\alpha_\mathrm{i}^*\leq\mathrm{C}$$

其中：

$$\boldsymbol{y} = \left[\underbrace{1,...,1}_{n} \quad \underbrace{-1,...,-1}_{n} \right]^{T}$$

注意：对于每条训练数据，其对应两个参数 α_i, α_i^*，其中 $Q[i][j] =< x[i] \cdot x[j] >$。

这样我们便将 $\varepsilon - \text{SVR}$ 转换成了 C-SVM 的形式，进而变可以使用上面的方法求解出最优的 α_i, α_i^*。

http://svms.org/tutorials/SmolaScholkopf1998.pdf