

# 支持向量机 (SVM) 中的 SMO 算法

## 1. 前言

最近又重新复习了一遍支持向量机 (SVM)。其实个人感觉SVM整体可以分成三个部分：

1. SVM理论本身：包括最大间隔超平面 (Maximum Margin Classifier)，拉格朗日对偶 (Lagrange Duality)，支持向量 (Support Vector)，核函数 (Kernel) 的引入，松弛变量的软间隔优化 (Outliers)，最小序列优化 (Sequential Minimal Optimization) 等。
2. 核方法 (Kernel)：其实核方法的发展是可以独立于SVM来看待的，核方法在很多其它算法中也会应用到。
3. 优化理论：这里主要介绍的是最小序列优化 (Sequential Minimal Optimization)，优化理论的发展也是独立于SVM的。

## 2. SVM理论基础

SVM的理论基础在上一篇博客的总结中可以参考：

<http://www.cnblogs.com/bentuwuying/p/6444249.html>。

对于支持向量机 (SVM) 的简单总结：

### 1. Maximum Margin Classifier

$$\begin{aligned} \min_{\gamma, w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \dots, m \end{aligned}$$

### 2. Lagrange Duality

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle. \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \dots, m \\ & \sum_{i=1}^m \alpha_i y^{(i)} = 0, \end{aligned}$$

### 3. Support Vector

### 4. Kernel

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{s.t.} \quad & \alpha_i \geq 0, i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

## 5. Outliers

$$\begin{aligned} \min_{\gamma, w, b} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \dots, m \\ & \xi_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m \\ & \sum_{i=1}^m \alpha_i y^{(i)} = 0, \end{aligned}$$

## 6. Sequential Minimal Optimization

个人觉得SMO又可以分为两部分：

- (1) 如何选择每次迭代时候的目标工作集，即选择哪两个拉格朗日乘子来迭代。
- (2) 如何对选择好的工作集（拉格朗日乘子）进行更新迭代。

## 3. SMO最初的版本 (Platt, 1998)

$$\begin{aligned} \max W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j (K(x_i, x_j)) \\ \text{s.t.} \quad \sum_{i=1}^n y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \quad (i = 1, 2, \dots, n) \end{aligned}$$

SMO就是要解这个凸二次规划问题，这里的C是个很重要的参数，它从本质上说是用来折中经验风险和置信风险的，C越大，置信风险越大，经验风险越小；并且所有的拉格朗日乘子都被限制在了以C为边长的大盒子里。SMO的出现使得我们不必去求助于昂贵的第三方工具去解决这个凸二次规划问题，目前对它的改进版本很多，这一节先介绍它的最初形式和思想。

SMO是Microsoft Research的John C. Platt在《Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines》一文中提出的，其基本思想是将Vapnik在1982年提出的Chunking方法推到极致，即：通过将原问题分解为一系列小规模凸二次规划问题而获得原问题解的方法，每次迭代只

优化由2个点组成的工作集，SMO算法每次启发式地选择两个拉格朗日乘子同时固定其它拉格朗日乘子来找到这两个拉格朗日乘子的最优值，直到达到**停止条件**。

## (1)、KKT条件

SMO是以C-SVC的KKT条件为基础进行后续操作的，这个KKT条件是：

$$\alpha_i = 0 \Leftrightarrow y_i u_i \geq 1$$

$$0 \leq \alpha_i \leq C \Leftrightarrow y_i u_i = 1$$

$$\alpha_i = C \Leftrightarrow y_i u_i \leq 1$$

其中  $u_i = \langle w, x_i \rangle + b$

上述条件其实就是KT互补条件，[SVM学习——软间隔优化](#)一文，有如下结论：

$$\alpha_i (y_i (\langle w, x_i \rangle + b) - 1 + \xi_i) = 0$$

$$\mu_i \xi_i = (\alpha_i - C) \xi_i = 0$$

从上面式子可以得到的信息是：当  $\alpha_i = C$  时，松弛变量  $\xi_i \geq 0$ ，此时有：

$y_i (\langle w, x_i \rangle + b) = 1 - \xi_i \Rightarrow y_i (\langle w, x_i \rangle + b) \leq 1 \Rightarrow y_i u_i \leq 1$ ，对应样本点就是误分点；当  $\alpha_i = 0$  时，松弛变量  $\xi_i$  为零，此时有  $y_i (\langle w, x_i \rangle + b) \geq 1 \Rightarrow y_i u_i \geq 1$ ，对应样本点就是内部点，即分类正确而又远离最大间隔分类超平面的那些样本点；而  $0 < \alpha_i < C$  时，松弛变量  $\xi_i$  为零，有  $y_i (\langle w, x_i \rangle + b) = 1 \Rightarrow y_i u_i = 1$ ，对应样本点就是支持向量。

## (2)、凸优化问题停止条件

对于凸优化问题，在实现时总需要适当的停止条件来结束优化过程，停止条件可以是：

1、监视目标函数  $W(\alpha)$  的增长率，在它低于某个容忍值时停止训练，这个条件是最直白和简单的，但是效果不好；

2、监视原问题的KKT条件，对于凸优化来说它们是收敛的充要条件，但是由于KKT条件本身是比较苛刻的，所以也需要设定一个容忍值，即所有样本在容忍值范围内满足KKT条件则认为训练可以结束；

3、监视可行间隙，它是原始目标函数值和对偶目标函数值的间隙，对于凸二次优化来说这个间隙是零，以一阶范数软间隔为例：

原始目标函数  $O(w, b)$  与对偶目标函数  $W(\alpha)$  的差为：

$$Gap = \frac{1}{2} \langle w, w \rangle + C \sum_{i=1}^n \xi_i - \left( \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j (K(x_i, x_j)) \right)$$

$$= \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j K(x_i, x_j) + C \sum_{i=1}^n \xi_i - \left( \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j (K(x_i, x_j)) \right)$$

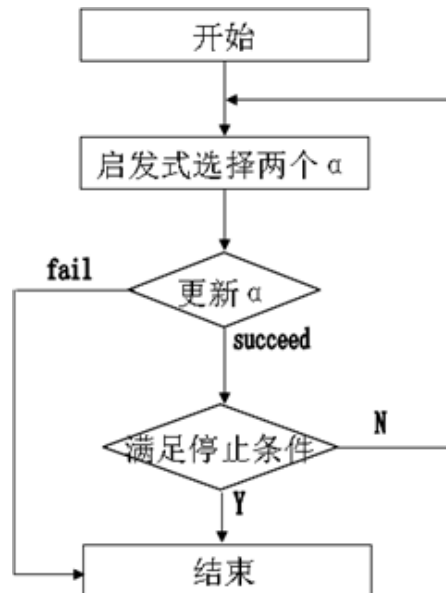
$$= \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j K(x_i, x_j) + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i$$

$$\begin{aligned}
 &= 2 \sum_{i=1}^n \alpha_i - 2W(\alpha) + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \\
 &= \sum_{i=1}^n \alpha_i - 2W(\alpha) + C \sum_{i=1}^n \xi_i
 \end{aligned}$$

定义比率:  $\frac{O(w,b)-W(\alpha)}{O(w,b)+1}$  , 可以利用这个比率达到某个容忍值作为停止条件。

### (3)、SMO思想

沿袭分解思想, 固定“Chunking工作集”的大小为2, 每次迭代只优化两个点的最小子集且可直接获得解析解, 算法流程:



### (4)、仅含两个Langrange乘子解析解

为了描述方便定义如下符号:

$$K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$$

$$f(\mathbf{x}_i) = \sum_{j=1}^n y_j \alpha_j K_{ij} + b$$

$$v_i = \sum_{j=3}^n y_j \alpha_j K_{ij} = f(\mathbf{x}_i) - \sum_{j=1}^2 y_j \alpha_j K_{ij} - b$$

于是目标函数就变成了:

$$W(\alpha_2) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j$$

$$= \alpha_1 + \alpha_2 + \sum_{i=3}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n (\sum_{j=1}^2 y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{j=3}^n y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j))$$

$$\begin{aligned}
&= \alpha_1 + \alpha_2 + \sum_{i=3}^n \alpha_i - \frac{1}{2} \sum_{i=1}^2 \left( \sum_{j=1}^2 y_i y_j \alpha_i \alpha_j K(x_i x_j) + \sum_{j=3}^n y_i y_j \alpha_i \alpha_j K(x_i x_j) \right) \\
&\quad - \frac{1}{2} \sum_{i=3}^n \left( \sum_{j=1}^2 y_i y_j \alpha_i \alpha_j K(x_i x_j) + \sum_{j=3}^n y_i y_j \alpha_i \alpha_j K(x_i x_j) \right) \\
&= \alpha_1 + \alpha_2 + \sum_{i=3}^n \alpha_i - \frac{1}{2} \sum_{i=1}^2 \sum_{j=1}^2 y_i y_j \alpha_i \alpha_j K(x_i x_j) - \sum_{i=1}^2 \sum_{j=3}^n y_i y_j \alpha_i \alpha_j K(x_i x_j) \\
&\quad - \frac{1}{2} \sum_{i=3}^n \sum_{j=3}^n y_i y_j \alpha_i \alpha_j K(x_i x_j) \\
&= \alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - y_1 y_2 K_{12} \alpha_1 \alpha_2 \\
&\quad - y_1 \alpha_1 \sum_{j=3}^n y_j \alpha_j K(x_1 x_j) - y_2 \alpha_2 \sum_{j=3}^n y_j \alpha_j K(x_2 x_j) \\
&\quad + \sum_{i=3}^n \alpha_i - \frac{1}{2} \sum_{i=3}^n \sum_{j=3}^n y_i y_j \alpha_i \alpha_j K(x_i x_j) \\
&= \alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - y_1 y_2 K_{12} \alpha_1 \alpha_2 \\
&\quad - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \text{constant}
\end{aligned}$$

注意第一个约束条件:  $\sum_{i=1}^n y_i \alpha_i = 0$ , 可以将  $\alpha_3, \dots, \alpha_n, y_3, \dots, y_n$  看作常数, 有  $\alpha_1 y_1 + \alpha_2 y_2 = C'$  ( $C'$  为常数, 我们不关心它的值), 等式两边同时乘以  $y_1$ , 得到  $\alpha_1 = \gamma - s \alpha_2$  ( $\gamma$  为常数, 其值为  $C' y_1$ , 我们不关心它,  $s = y_1 y_2$ )。将  $\alpha_1$  用上式替换则得到一个只含有变量  $\alpha_2$  的求极值问题:

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

这下问题就简单了, 对  $\alpha_2$  求偏导数得到:

$$\frac{\partial W(\alpha_2)}{\partial \alpha_2} = -s + 1 + s K_{11} \gamma - K_{11} \alpha_2 - K_{22} \alpha_2 - s \gamma K_{12} + 2 K_{12} \alpha_2 + y_2 v_1 - y_2 v_2 = 0$$

将  $y_2^2 = 1$ 、 $s = y_1 y_2$  带入上式有:

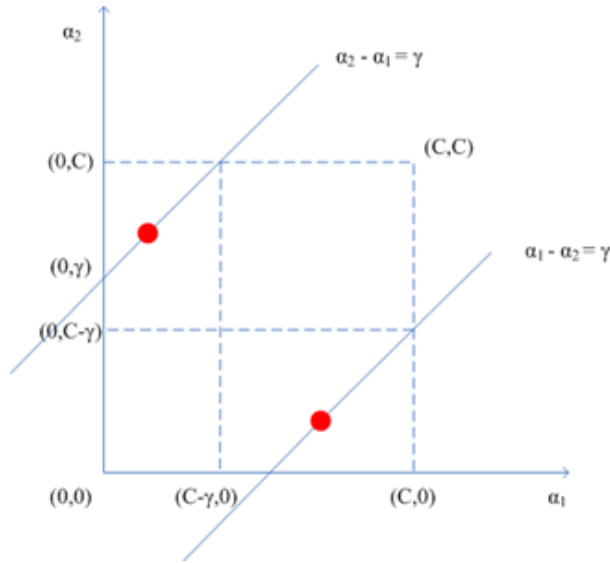
$$\alpha_2^{\text{new}} = \frac{y_2(y_2 - y_1 + y_1 \gamma (K_{11} - K_{12}) + v_1 - v_2)}{K_{11} + K_{22} - 2K_{12}}$$

带入  $v_i$ 、 $\gamma = \alpha_1^{\text{old}} + s \alpha_2^{\text{old}}$ , 用  $E_i = f(x_i) - y_i$  表示误差项(可以想象, 即使分类正确,  $f(x_i)$  的值也可能很大)、 $\eta = K_{11} + K_{22} - 2K_{12} = \|\Phi(x_1) - \Phi(x_2)\|^2$  ( $\Phi$  是原始空间向特征空间的映射), 这里  $\sqrt{\eta}$  可以看成是一个度量两个样本相似性的距离, 换句话说, **一旦选择核函数则意味着你已经定义了输入空间中元素的相似性。**

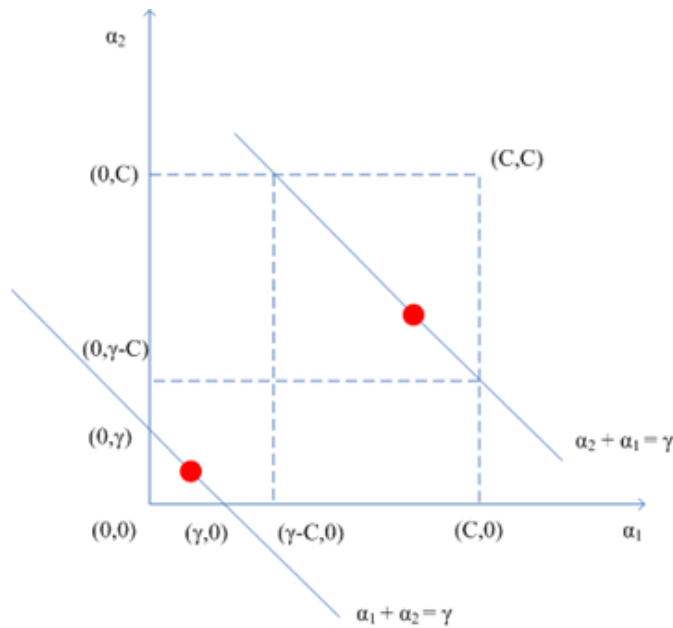
最后得到迭代式:

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

注意第二个约束条件——那个强大的盒子： $0 \leq \alpha_i \leq C$ ，这意味着 $\alpha_2^{new}$ 也必须落入这个盒子中，综合考虑两个约束条件，下图更直观：



$y_1$ 和 $y_2$ 异号的情形



$y_1$ 和 $y_2$ 同号的情形

可以看到 $\alpha_1, \alpha_2$ 两个乘子既要位于边长为C的盒子里又要在相应直线上，于是对于 $\alpha_2$ 的界来说，有如下情况：

$$\begin{cases} L = \max\{0, \alpha_2^{old} - \alpha_1^{old}\} & y_1 y_2 = -1, \\ L = \max\{0, \alpha_1^{old} + \alpha_2^{old} - C\} & y_1 y_2 = 1, \\ H = \min\{C, C + \alpha_2^{old} - \alpha_1^{old}\} & y_1 y_2 = -1 \\ H = \min\{C, \alpha_1^{old} + \alpha_2^{old}\} & y_1 y_2 = 1 \end{cases}$$

整理得下式：

$$\alpha_2^{new, clipped} = \begin{cases} L & \alpha_2^{new} \leq L \\ \alpha_2^{new} & L < \alpha_2^{new} < H \\ H & \alpha_2^{new} \geq H \end{cases}$$

又因为  $\alpha_1^{old} = \gamma - s\alpha_2^{old}$ ,  $\alpha_1^{new} = \gamma - s\alpha_2^{new,clipped}$ , 消去  $\gamma$  后得到:

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new,clipped})$$

## (5)、启发式的选择方法

根据选择的停止条件可以确定怎么样选择点能对算法收敛贡献最大, 例如使用**监视可行间隙**的方法, 一个最直白的选择就是首先优化那些**最违反KKT条件的点**, 所谓违反KKT条件是指:

$$\alpha_i = 0 \ \&\& \ y_i u_i < 1$$

$$0 \leq \alpha_i \leq C \ \&\& \ y_i u_i \neq 1$$

$$\alpha_i = C \ \&\& \ y_i u_i > 1$$

由前面的停止条件3可知, 对可行间隙贡献最大的点是那些

$$Gap_i = \alpha_i (y_i (\sum_{j=1}^n \alpha_j y_j K(x_i, x_j)) - 1) + C \xi_i = \alpha_i (y_i u_i - 1 - y_i b) + C \xi_i$$

$$\text{其中, } \xi_i = \max(0, 1 - y_i u_i)$$

取值大的点, 这些点导致可行间隙变大, 因此应该首先优化它们, 原因如下:

1、当满足KKT条件: 即  $\alpha_i = 0 \Leftrightarrow y_i u_i \geq 1$  时,  $Gap_i = \alpha_i (y_i u_i - 1 - y_i b) + C \xi_i = 0$

当违背KKT条件: 即  $\alpha_i = 0 \ \&\& \ y_i u_i < 1$  时,  $\xi_i > 0$ , 于是  $Gap_i = \alpha_i (y_i u_i - 1 - y_i b) + C \xi_i > 0$

可见, 由于违背KKT条件导致可行间隙变大;

2、当满足KKT条件: 即  $0 \leq \alpha_i \leq C \Leftrightarrow y_i u_i = 1$  时,  $Gap_i = \alpha_i (y_i u_i - 1 - y_i b) + C \xi_i = -\alpha_i y_i b$

当违背KKT条件: 即  $0 \leq \alpha_i \leq C \ \&\& \ y_i u_i \neq 1$  时

若  $y_i u_i > 1$  则  $\xi_i = 0$  且  $Gap_i = \alpha_i (y_i u_i - 1 - y_i b) + C \xi_i = \alpha_i (y_i u_i - 1) - \alpha_i y_i b$ , 其中  $\alpha_i (y_i u_i - 1) > 0$

若  $y_i u_i < 1$  则  $\xi_i = 1 - y_i u_i$  且  $Gap_i = \alpha_i (y_i u_i - 1 - y_i b) + C(1 - y_i u_i) = (C - \alpha_i)(1 - y_i u_i) - \alpha_i y_i b$ , 其中  $(C - \alpha_i)(1 - y_i u_i) > 0$

可见, 由于违背KKT条件依然导致可行间隙变大;

3、当满足KKT条件: 即  $\alpha_i = C \Leftrightarrow y_i u_i \leq 1$  时,  $Gap_i = \alpha_i (y_i u_i - 1 - y_i b) + C \xi_i = -C y_i b$

当违背KKT条件: 即  $\alpha_i = C \ \&\& \ y_i u_i > 1$  时,  $\xi_i = 0$  且  $Gap_i = \alpha_i (y_i u_i - 1 - y_i b) + C \xi_i = C(y_i u_i - 1) - C y_i b$ , 其中  $C(y_i u_i - 1) > 0$

可见, 由于违背KKT条件还是会导致可行间隙变大。

SMO的启发式选择有两个策略:

启发式选择1:

最外层循环，首先，在所有样本中选择违反KKT条件的一个乘子作为最外层循环，用“启发式选择2”选择另外一个乘子并进行这两个乘子的优化，接着，从所有非边界样本中选择违反KKT条件的一个乘子作为最外层循环，用“启发式选择2”选择另外一个乘子并进行这两个乘子的优化(之所以选择非边界样本是为了提高找到违反KKT条件的点的机会)，最后，如果上述非边界样本中没有违反KKT条件的样本，则再从整个样本中去找，直到所有样本中没有需要改变的乘子或者满足其它停止条件为止。

启发式选择2：

内层循环的选择标准可以从下式看出：

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

要加快第二个乘子的迭代速度，就要使  $\alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$  最大，而在  $\eta$  上没什么文章可做，于是只能使  $|E_1 - E_2|$  最大。

确定第二个乘子方法：

- 1、首先在非界乘子中寻找使得  $|E_1 - E_2|$  最大的样本；
- 2、如果1中没找到则从随机位置查找非界乘子样本；
- 3、如果2中也没找到，则从随机位置查找整个样本(包含界上和界下乘子)。

## (6)、关于两乘子优化的说明

由式子

$$\frac{\partial W(\alpha_2)}{\partial \alpha_2} = -s + 1 + sK_{11}\gamma - K_{11}\alpha_2 - K_{22}\alpha_2 - s\gamma K_{12} + 2K_{12}\alpha_2 + y_2v_1 - y_2v_2$$

可知：

$$\frac{\partial W^2(\alpha_2)}{\partial \alpha_2^2} = -K_{11} - K_{22} + 2K_{12} = -\eta$$

于是对于这个单变量二次函数而言,如果其二阶导数  $-\eta < 0$ ，则二次函数开口向下，可以用上述迭代的方法更新乘子，如果  $-\eta \geq 0$ ，则目标函数只能在边界上取得极值(此时二次函数开口向上)，换句话说，SMO要能处理  $\eta$  取任何值的情况，于是在  $-\eta \geq 0$  时有以下式子：

1、 $\alpha_2^{new,clipped} = L$  时：

$$\alpha_1^{new} = \alpha_1^{old} + s(\alpha_2^{old} - L)$$

2、 $\alpha_2^{new,clipped} = H$  时：

$$\alpha_1^{new} = \alpha_1^{old} + s(\alpha_2^{old} - H)$$

3、 $W(\alpha_1, \alpha_2) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j K(x_i, x_j) \alpha_i \alpha_j$



$$\begin{aligned}
&= \alpha_1 + \alpha_2 - \frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 - y_1y_2K_{12}\alpha_1\alpha_2 - y_1\alpha_1v_1 - y_2\alpha_2v_2 + \text{constant} \\
&= \alpha_1(1 - y_1v_1) + \alpha_2(1 - y_2v_2) - \frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 - y_1y_2K_{12}\alpha_1\alpha_2 + \text{constant} \\
&= \alpha_1(1 - y_1v_1) + \alpha_2(1 - y_2v_2) - \frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 - y_1y_2K_{12}\alpha_1\alpha_2 + \text{constant} \\
&= \alpha_1y_1(y_1 - (f(x_1) - \alpha_1y_1K_{11} - \alpha_2y_2K_{12} - b)) + \alpha_2y_2(y_2 - (f(x_2) - \alpha_1y_1K_{12} - \alpha_2y_2K_{22} - b)) \\
&\quad - \frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 - y_1y_2K_{12}\alpha_1\alpha_2 + \text{constant} \\
&= \alpha_1^{\text{new}}(y_1(b - E_1) + \alpha_1^{\text{old}}K_{11} + s\alpha_2^{\text{old}}K_{12}) + \alpha_2^{\text{new,clipped}}(y_2(b - E_2) + \alpha_2^{\text{old}}K_{22} + s\alpha_1^{\text{old}}K_{12}) \\
&\quad - \frac{1}{2}K_{11}\alpha_1^{\text{new}2} - \frac{1}{2}K_{22}\alpha_2^{\text{new,clipped}2} - y_1y_2K_{12}\alpha_1^{\text{new}}\alpha_2^{\text{new,clipped}} + \text{constant}
\end{aligned}$$

分别将乘子带入得到两种情况下的目标函数值： $W_L$ 和 $W_H$ 。显然，哪种情况下目标函数值最大，则乘子就往哪儿移动，如果目标函数的差在某个指定精度范围内，说明优化没有进展。

另外发现，每一步迭代都需要计算输出 $u$ 进而得到 $E$ ，于是还要更新阈值 $b$ ，使得新的乘子 $\alpha_1$ 、 $\alpha_2$ 满足KKT条件，考虑 $\alpha_1$ 、 $\alpha_2$ 至少有一个在界内，则需要满足 $0 \leq \alpha_i \leq C \Leftrightarrow y_i u_i = 1$ ，于是 $b$ 的迭代可以这样得到：

1、设 $\alpha_1^{\text{new}}$ 在界内，则：

$$y_1 u_1^{\text{new}} = 1 \Rightarrow y_1(\alpha_1^{\text{new}}y_1K_{11} + \alpha_2^{\text{new,clipped}}y_2K_{21} + \sum_{i=3}^n (\alpha_i y_i K_{i1}) + b^{\text{new}}) = 1$$

又因为：

$$\begin{aligned}
E_1 &= \alpha_1^{\text{old}}y_1K_{11} + \alpha_2^{\text{old}}y_2K_{21} + \sum_{i=3}^n (\alpha_i y_i K_{i1}) + b^{\text{old}} - y_1 \\
\Rightarrow \sum_{i=3}^n (\alpha_i y_i K_{i1}) &= E_1 - \alpha_1^{\text{old}}y_1K_{11} - \alpha_2^{\text{old}}y_2K_{21} - b^{\text{old}} + y_1
\end{aligned}$$

于是有：

$$\begin{aligned}
&y_1(\alpha_1^{\text{new}}y_1K_{11} + \alpha_2^{\text{new,clipped}}y_2K_{21} + \sum_{i=3}^n (\alpha_i y_i K_{i1}) + b^{\text{new}}) \\
&= y_1(\alpha_1^{\text{new}}y_1K_{11} + \alpha_2^{\text{new,clipped}}y_2K_{21} + E_1 - \alpha_1^{\text{old}}y_1K_{11} - \alpha_2^{\text{old}}y_2K_{21} - b^{\text{old}} + y_1 + b^{\text{new}}) = 1
\end{aligned}$$

等式两边同乘 $y_1$ 后移项得：

$$\begin{aligned}
b^{\text{new}} &= -\alpha_1^{\text{new}}y_1K_{11} - \alpha_2^{\text{new,clipped}}y_2K_{21} - E_1 + \alpha_1^{\text{old}}y_1K_{11} + \alpha_2^{\text{old}}y_2K_{21} + b^{\text{old}} \\
&= (\alpha_1^{\text{old}} - \alpha_1^{\text{new}})y_1K_{11} + (\alpha_2^{\text{old}} - \alpha_2^{\text{new,clipped}})y_2K_{21} - E_1 + b^{\text{old}};
\end{aligned}$$

2、设  $\alpha_2^{new,clipped}$  在界内，则：

$$b^{new} = (\alpha_1^{old} - \alpha_1^{new})y_1K_{12} + (\alpha_2^{old} - \alpha_2^{new,clipped})y_2K_{22} - E_2 + b^{old};$$

3、设  $\alpha_1^{new}$ 、 $\alpha_2^{new,clipped}$  都在界内，则：情况1和情况2的  $b$  值相等，任取一个；

4、设  $\alpha_1^{new}$ 、 $\alpha_2^{new,clipped}$  都不在界内，则： $b^{new}$  取值为情况1和情况2之间的任意值。

## (7)、提高SMO的速度

从实现上来说，对于标准的SMO能提高速度的地方有：

1、能用缓存的地方尽量用，例如，缓存核矩阵，减少重复计算，但是增加了空间复杂度；

2、如果SVM的核为线性核时候，可直接更新  $w$ ，毕竟每次计算  $w = \sum_{i=1}^n y_i \alpha_i x_i$  的代价较高，于是可以利用旧的乘子信息来更新  $w$ ，具体如下：

$w^{new} = w^{old} + (\alpha_1^{new} - \alpha_1^{old})y_1x_1 + (\alpha_2^{new} - \alpha_2^{old})y_2x_2$ ，应用到这个性质的例子可以参见[SVM学习——Coordinate Descent Method](#)。

3、关注可以并行的点，用并行方法来改进，例如可以使用MPI，将样本分为若干份，在查找  $|E_1 - E_2|$  最大的乘子时可以现在各个节点先找到局部最大点，然后再从中找到全局最大点；又如停止条件是监视对偶间隙，那么可以考虑在每个节点上计算出局部可行间隙，最后在master节点上将局部可行间隙累加得到全局可行间隙。

对标准SMO的改进有很多文献，例如使用“Maximal Violating Pair”去启发式的选择乘子是一种很有效的方法，还有使用“Second Order Information”的方法，我觉得理想的算法应该是：算法本身的收敛速度能有较大提高，同时算法可并行程度也较高。

## 4. SMO更新的版本 (Fan, 2005)

前面提到过，SMO可以分为两部分：

**(1) 如何选择每次迭代时候的目标工作集，即选择哪两个拉格朗日乘子来迭代。**

**(2) 如何对选择好的工作集（拉格朗日乘子）进行更新迭代。**

而如何选择工作集，是SMO算法很重要的一个部分，因为不同的选择方式可以导致不同的训练速度。

Rong-En Fan等人在2005的paper《Working Set Selection Using Second Order Information for Training Support Vector Machines》介绍了每次迭代时几种不同的工作集选择方法。

首先还是放出SMO需要优化的目标函数：

$$\begin{aligned} \min_{\alpha} \quad & f(\alpha) = \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, i = 1, \dots, l, \\ & y^T \alpha = 0, \end{aligned} \quad (1)$$

## (1) Algorithm 1 (SMO-type decomposition method)

1. Find  $\alpha^1$  as the initial feasible solution. Set  $k = 1$ .
2. If  $\alpha^k$  is an optimal solution of (1), stop. Otherwise, find a *two-element* working set  $B = \{i, j\} \subset \{1, \dots, l\}$ . Define  $N \equiv \{1, \dots, l\} \setminus B$  and  $\alpha_B^k$  and  $\alpha_N^k$  to be sub-vectors of  $\alpha^k$  corresponding to  $B$  and  $N$ , respectively.
3. Solve the following sub-problem with the variable  $\alpha_B$ :

$$\begin{aligned}
 \min_{\alpha_B} \quad & \frac{1}{2} [\alpha_B^T \quad (\alpha_N^k)^T] \begin{bmatrix} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{bmatrix} \begin{bmatrix} \alpha_B \\ \alpha_N^k \end{bmatrix} - [\mathbf{e}_B^T \quad \mathbf{e}_N^T] \begin{bmatrix} \alpha_B \\ \alpha_N^k \end{bmatrix} \\
 = \quad & \frac{1}{2} \alpha_B^T Q_{BB} \alpha_B + (-\mathbf{e}_B + Q_{BN} \alpha_N^k)^T \alpha_B + \text{constant} \\
 = \quad & \frac{1}{2} [\alpha_i \quad \alpha_j] \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (-\mathbf{e}_B + Q_{BN} \alpha_N^k)^T \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + \text{constant} \\
 \text{subject to} \quad & 0 \leq \alpha_i, \alpha_j \leq C, \\
 & y_i \alpha_i + y_j \alpha_j = -\mathbf{y}_N^T \alpha_N^k,
 \end{aligned} \tag{2}$$

where  $\begin{bmatrix} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{bmatrix}$  is a permutation of the matrix  $Q$ .

4. Set  $\alpha_B^{k+1}$  to be the optimal solution of (2) and  $\alpha_N^{k+1} \equiv \alpha_N^k$ . Set  $k \leftarrow k + 1$  and goto Step 2.

## (2) WSS 1 (Working set selection via the “maximal violating pair” )

1. Select

$$\begin{aligned}
 i &\in \arg \max_t \{-y_t \nabla f(\alpha^k)_t \mid t \in I_{\text{up}}(\alpha^k)\}, \\
 j &\in \arg \min_t \{-y_t \nabla f(\alpha^k)_t \mid t \in I_{\text{low}}(\alpha^k)\},
 \end{aligned}$$

where

$$\begin{aligned}
 I_{\text{up}}(\alpha) &\equiv \{t \mid \alpha_t < C, y_t = 1 \text{ or } \alpha_t > 0, y_t = -1\}, \text{ and} \\
 I_{\text{low}}(\alpha) &\equiv \{t \mid \alpha_t < C, y_t = -1 \text{ or } \alpha_t > 0, y_t = 1\}.
 \end{aligned} \tag{3}$$

2. Return  $B = \{i, j\}$ .

这个working set selection是Keerthi等人在2001年提出的，在2001年发布的libSVM中有所应用。

该working set selection可以由 (1) 式的KKT条件得出：假设存在向量  $\alpha$  是 (1) 式的解，则必然存在实数  $b$  和两个非负向量  $\lambda$  and  $\mu$  使得下式成立：

$$\begin{aligned}
 \nabla f(\alpha) + b\mathbf{y} &= \lambda - \mu, \\
 \lambda_i \alpha_i &= 0, \mu_i (C - \alpha_i) = 0, \lambda_i \geq 0, \mu_i \geq 0, i = 1, \dots, l,
 \end{aligned}$$

其中， $\nabla f(\alpha) \equiv Q\alpha - \mathbf{e}$  是目标函数的梯度。

上面的条件可以被重写为：

$$\begin{aligned} \nabla f(\alpha)_i + by_i &\geq 0 & \text{if } \alpha_i < C, \\ \nabla f(\alpha)_i + by_i &\leq 0 & \text{if } \alpha_i > 0. \end{aligned}$$

$$-y_i \nabla f(\alpha)_i \leq b, \forall i \in I_{\text{up}}(\alpha), \text{ and}$$

$$-y_i \nabla f(\alpha)_i \geq b, \forall i \in I_{\text{low}}(\alpha),$$

进一步，有

$$m(\alpha) \equiv \max_{i \in I_{\text{up}}(\alpha)} -y_i \nabla f(\alpha)_i, \text{ and } M(\alpha) \equiv \min_{i \in I_{\text{low}}(\alpha)} -y_i \nabla f(\alpha)_i$$

令，

则目标函数存在最优解  $\alpha$  的条件是：

$$m(\alpha) \leq M(\alpha), \quad (6)$$

**Definition 1 (Violating pair)** If  $i \in I_{\text{up}}(\alpha), j \in I_{\text{low}}(\alpha)$ , and  $-y_i \nabla f(\alpha)_i > -y_j \nabla f(\alpha)_j$ , then  $\{i, j\}$  is a “violating pair.”

由上面这个关于 “Violating pair” 的定义可以看出，最大程度上违反 (6) 式条件的  $\{i, j\}$  pair 即是 working set 的最佳选择，因为我们需要对这些最违反 (6) 式的  $\{i, j\}$  pair 做更新迭代，让它们符合 (6) 式的要求，便会逐步让目标函数得到最优值。具体的理论定理如下：

**Theorem 2 (Hush and Scovel, 2003)** Assume  $Q$  is positive semi-definite. SMO-type methods have the strict decrease of the function value (i.e.,  $f(\alpha^{k+1}) < f(\alpha^k), \forall k$ ) if and only if  $B$  is a violating pair.

有趣的是，选择最大程度上违反 KKT 条件的  $\{i, j\}$  pair 与 “求目标函数的一阶近似的最小值” 时候取得的  $\{i, j\}$  pair 是一致的。即通过 **WSS 1** 获得的  $\{i, j\}$  pair 满足：

$$\{i, j\} = \arg \min_{B: |B|=2} \text{Sub}(B), \quad (7)$$

where

$$\text{Sub}(B) \equiv \min_{\mathbf{d}_B} \nabla f(\alpha^k)_B^T \mathbf{d}_B \quad (8a)$$

$$\text{subject to } \mathbf{y}_B^T \mathbf{d}_B = 0, \quad (8b)$$

$$d_t \geq 0, \text{ if } \alpha_t^k = 0, t \in B, \quad (8b)$$

$$d_t \leq 0, \text{ if } \alpha_t^k = C, t \in B, \quad (8c)$$

$$-1 \leq d_t \leq 1, t \in B. \quad (8d)$$

通过定义  $\mathbf{d}^T \equiv [\mathbf{d}_B^T, \mathbf{0}_N^T]$ ，(8a) 的目标函数即是对  $f(\alpha^k + \mathbf{d})$  的一阶近似进行求最优解：

$$\begin{aligned} f(\alpha^k + \mathbf{d}) &\approx f(\alpha^k) + \nabla f(\alpha^k)^T \mathbf{d} \\ &= f(\alpha^k) + \nabla f(\alpha^k)_B^T \mathbf{d}_B \end{aligned}$$

其中,  $\mathbf{y}_B^T \mathbf{d}_B = 0$  是由于  $\mathbf{y}^T(\boldsymbol{\alpha}^k + \mathbf{d}) = 0$  and  $\mathbf{y}^T \boldsymbol{\alpha}^k = 0$ , 而 (8b) 和 (8c) 是由于  $0 \leq \alpha_t \leq C$ 。由于 (8a) 是线性函数,  $-1 \leq d_t \leq 1, t \in B$  则避免了目标函数取值无穷小。

第一眼看上去, (7) 式似乎需要对所有的拉格朗日乘子遍历一遍才能够找出最优的  $\{i, j\}$  pair, 然而, **WSS 1** 可以在线性时间复杂度内找到最优值。证明如下:

## Proof

For any given  $\{i, j\}$ , we can substitute  $\hat{d}_i \equiv y_i d_i$  and  $\hat{d}_j \equiv y_j d_j$  to (8), so the objective function becomes

$$(-y_i \nabla f(\boldsymbol{\alpha}^k)_i + y_j \nabla f(\boldsymbol{\alpha}^k)_j) \hat{d}_j. \quad (44)$$

As  $d_i = d_j = 0$  is feasible for (8), the minimum of (44) is zero or a negative number. If  $-y_i \nabla f(\boldsymbol{\alpha}^k)_i > -y_j \nabla f(\boldsymbol{\alpha}^k)_j$ , using the condition  $\hat{d}_i + \hat{d}_j = 0$ , the only possibility for (44) to be negative is  $\hat{d}_j < 0$  and  $\hat{d}_i > 0$ . From (3), (8b), and (8c), this corresponds to  $i \in I_{\text{up}}(\boldsymbol{\alpha}^k)$  and  $j \in I_{\text{low}}(\boldsymbol{\alpha}^k)$ . Moreover, the minimum occurs at  $\hat{d}_j = -1$  and  $\hat{d}_i = 1$ . The situation of  $-y_i \nabla f(\boldsymbol{\alpha}^k)_i < -y_j \nabla f(\boldsymbol{\alpha}^k)_j$  is similar.

Therefore, solving (7) is essentially the same as

$$\begin{aligned} & \min \left\{ \min(y_i \nabla f(\boldsymbol{\alpha}^k)_i - y_j \nabla f(\boldsymbol{\alpha}^k)_j, 0) \mid i \in I_{\text{up}}(\boldsymbol{\alpha}^k), j \in I_{\text{low}}(\boldsymbol{\alpha}^k) \right\} \\ & = \min(-m(\boldsymbol{\alpha}^k) + M(\boldsymbol{\alpha}^k), 0). \end{aligned}$$

Hence, if there are violating pairs, the maximal one solves (7).

## (3) A New Working Set Selection

上面是使用了目标函数的一阶近似作为代替进行优化, 于是乎, 我们可以再进一步, 使用目标函数的二阶近似作为代替进行优化:

$$\begin{aligned} f(\boldsymbol{\alpha}^k + \mathbf{d}) - f(\boldsymbol{\alpha}^k) &= \nabla f(\boldsymbol{\alpha}^k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\boldsymbol{\alpha}^k) \mathbf{d} \\ &= \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B + \frac{1}{2} \mathbf{d}_B^T \nabla^2 f(\boldsymbol{\alpha}^k)_{BB} \mathbf{d}_B \end{aligned}$$

$$\min_{B: |B|=2} \text{Sub}(B), \quad (10)$$

where

$$\text{Sub}(B) \equiv \min_{\mathbf{d}_B} \frac{1}{2} \mathbf{d}_B^T \nabla^2 f(\boldsymbol{\alpha}^k)_{BB} \mathbf{d}_B + \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B \quad (11a)$$

$$\text{subject to } \mathbf{y}_B^T \mathbf{d}_B = 0, \quad (11b)$$

$$d_t \geq 0, \text{ if } \alpha_t^k = 0, t \in B, \quad (11c)$$

$$d_t \leq 0, \text{ if } \alpha_t^k = C, t \in B. \quad (11d)$$

## (4) WSS 2 (Working set selection using second order information)

1. Select

$$i \in \arg \max_t \{-y_t \nabla f(\alpha^k)_t \mid t \in I_{\text{up}}(\alpha^k)\}.$$

2. Consider  $\text{Sub}(B)$  defined in (11) and select

$$j \in \arg \min_t \{\text{Sub}(\{i, t\}) \mid t \in I_{\text{low}}(\alpha^k), -y_t \nabla f(\alpha^k)_t < -y_i \nabla f(\alpha^k)_i\}. \quad (12)$$

3. Return  $B = \{i, j\}$ .

下面的理论证明按照**WSS 2**可以有效地解决 (11) 式中的最优值问题:

**Theorem 3** *If  $B = \{i, j\}$  is a violating pair and  $K_{ii} + K_{jj} - 2K_{ij} > 0$ , then (11) has the optimal objective value*

$$-\frac{(-y_i \nabla f(\alpha^k)_i + y_j \nabla f(\alpha^k)_j)^2}{2(K_{ii} + K_{jj} - 2K_{ij})}.$$

**Proof** Define  $\hat{d}_i \equiv y_i d_i$  and  $\hat{d}_j \equiv y_j d_j$ . From  $\mathbf{y}_B^T \mathbf{d}_B = 0$ , we have  $\hat{d}_i = -\hat{d}_j$  and

$$\begin{aligned} & \frac{1}{2} \begin{bmatrix} d_i & d_j \end{bmatrix} \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} d_i \\ d_j \end{bmatrix} + \begin{bmatrix} \nabla f(\alpha^k)_i & \nabla f(\alpha^k)_j \end{bmatrix} \begin{bmatrix} d_i \\ d_j \end{bmatrix} \\ &= \frac{1}{2} (K_{ii} + K_{jj} - 2K_{ij}) \hat{d}_j^2 + (-y_i \nabla f(\alpha^k)_i + y_j \nabla f(\alpha^k)_j) \hat{d}_j. \end{aligned} \quad (13)$$

Since  $K_{ii} + K_{jj} - 2K_{ij} > 0$  and  $B$  is a violating pair, we can define

$$a_{ij} \equiv K_{ii} + K_{jj} - 2K_{ij} > 0 \quad \text{and} \quad b_{ij} \equiv -y_i \nabla f(\alpha^k)_i + y_j \nabla f(\alpha^k)_j > 0. \quad (14)$$

Then (13) has the minimum at

$$\hat{d}_j = -\hat{d}_i = -\frac{b_{ij}}{a_{ij}} < 0, \quad (15)$$

and

$$\text{the objective function (11a)} = -\frac{b_{ij}^2}{2a_{ij}}.$$

Moreover, we can show that  $\hat{d}_i$  and  $\hat{d}_j$  ( $d_i$  and  $d_j$ ) indeed satisfy (11c)-(11d). If  $j \in I_{\text{low}}(\alpha^k)$ ,  $\alpha_j^k = 0$  implies  $y_j = -1$  and hence  $d_j = y_j \hat{d}_j > 0$ , a condition required by (11c). Other cases are similar. Thus  $\hat{d}_i$  and  $\hat{d}_j$  defined in (15) are optimal for (11). ■

Note that if  $K$  is positive definite, then for any  $i \neq j$ ,  $K_{ii} + K_{jj} - 2K_{ij} > 0$ . Using Theorem 3, (12) in WSS 2 is reduced to a very simple form:

$$j \in \arg \min_t \left\{ -\frac{b_{it}^2}{a_{it}} \mid t \in I_{\text{low}}(\alpha^k), -y_t \nabla f(\alpha^k)_t < -y_i \nabla f(\alpha^k)_i \right\},$$

## (5) Non-Positive Definite Kernel Matrices

前面的方法并没有涵盖  $K_{ii} + K_{jj} - 2K_{ij} \leq 0$  的情况, 对于这种情况, Chen等人在2006年给出了解决方法:



$$\begin{aligned}
\min_{\alpha_i, \alpha_j} \quad & \frac{1}{2} [\alpha_i \quad \alpha_j] \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (-\mathbf{e}_B + Q_{BN} \alpha_N^k)^T \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + \\
& \frac{\tau - a_{ij}}{4} ((\alpha_i - \alpha_i^k)^2 + (\alpha_j - \alpha_j^k)^2) \\
\text{subject to} \quad & 0 \leq \alpha_i, \alpha_j \leq C, \\
& y_i \alpha_i + y_j \alpha_j = -\mathbf{y}_N^T \alpha_N^k,
\end{aligned} \tag{16}$$

$$\begin{aligned}
\text{Sub}(B) \equiv \min_{d_B} \quad & \frac{1}{2} \mathbf{d}_B^T \nabla^2 f(\alpha^k)_{BB} \mathbf{d}_B + \nabla f(\alpha^k)_B^T \mathbf{d}_B + \frac{\tau - a_{ij}}{4} (d_i^2 + d_j^2) \\
\text{subject to} \quad & \text{constraints of (11)}.
\end{aligned} \tag{18}$$

## (6) WSS 3 (Working set selection using second order information: any symmetric K)

1. Define  $a_{ts}$  and  $b_{ts}$  as in (14), and

$$\bar{a}_{ts} \equiv \begin{cases} a_{ts} & \text{if } a_{ts} > 0, \\ \tau & \text{otherwise.} \end{cases} \tag{19}$$

Select

$$\begin{aligned}
i & \in \arg \max_t \{-y_t \nabla f(\alpha^k)_t \mid t \in I_{\text{up}}(\alpha^k)\}, \\
j & \in \arg \min_t \left\{ -\frac{b_{it}^2}{\bar{a}_{it}} \mid t \in I_{\text{low}}(\alpha^k), -y_t \nabla f(\alpha^k)_t < -y_i \nabla f(\alpha^k)_i \right\}.
\end{aligned} \tag{20}$$

2. Return  $B = \{i, j\}$ .

于是，使用**WSS 3**来对SMO-type 分解方法选择working set的步骤为：

## (7) Algorithm 2 (An SMO-type decomposition method using WSS 3)

1. Find  $\alpha^1$  as the initial feasible solution. Set  $k = 1$ .
2. If  $\alpha^k$  is a stationary point of (1), stop. Otherwise, find a working set  $B = \{i, j\}$  by WSS 3.
3. Let  $a_{ij}$  be defined as in (14). If  $a_{ij} > 0$ , solve the sub-problem (2). Otherwise, solve (16). Set  $\alpha_B^{k+1}$  to be the optimal point of the sub-problem.
4. Set  $\alpha_N^{k+1} \equiv \alpha_N^k$ . Set  $k \leftarrow k + 1$  and goto Step 2.

分类: [Support Vector Machine](#)

标签: [SVM](#), [SMO](#), [Optimization](#)

[好文要顶](#) [关注我](#) [收藏该文](#)

[笨兔勿应](#)

[关注 - 4](#)

[粉丝 - 43](#)[+加关注](#)

0

0

[« 上一篇: 《这就是搜索引擎》框架图](#)[» 下一篇: CTR预估中的贝叶斯平滑方法 \(二\) 参数估计和代码实现](#)posted @ 2017-02-26 17:47 [笨兔勿应](#) 阅读(4032) 评论(2) [编辑](#) [收藏](#)