SMO算法在SVM中的应用

机器学习的一些算法 svm 动态规划

序列最小优化算法(Sequential Minimal Optimization, SMO)

SMO在SVM算法中的应用

整体算法流程

先来看一下我们的优化目标：

$$T(\lambda_1, \lambda_2, \ldots, \lambda_m) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j)$$

$$max_{\lambda} T(\lambda_1, \lambda_2, \ldots, \lambda_m)$$

$$s.t. \sum_{i=1}^{m} \lambda_i y_i = 0$$

$$0 \leq \lambda_i \leq C, i = 1, 2, \ldots, m$$

一共有 m 个参数需优化。

SMO 是一种动态规划算法，它的基本思想非常简单：每次只优化一个参数，其他参数先固定住，仅求当前这一个优化参数的极值。

可惜，我们的优化目标有约束条件：$\sum(\lambda_i y_i) = 0$，其中$i = 1，2，\ldots, m$。如果我们一次只优化一个参数，就没法体现约束条件了。

于是，我们这样做：

1. 选择两个需要更新的变量 λ_i和 λ_j，固定它们以外的其他变量。
   这样，约束条件就变成了：

   $$\lambda_i y_i + \lambda_j y_j = c, \lambda_i \geqslant 0, \lambda_j \geqslant 0$$

   其中：
   $C = -\sum_{k \neq i,j} \lambda_k y_k$
   这样由此，可得出 $\lambda_j = \frac{(C - \lambda_i y_i)}{y_j}$，也就是我们可以用 λ_i 的表达式代替λ_j。
   将这个替代式带入优化目标函数。就相当于把目标问题转化成了一个单变量的二次规划问题，仅有的约束是 $\lambda_i \geqslant 0$。
2. 对于仅有一个约束条件的最优化问题，我们完全可以在λ_i上，对问题函数$T(\lambda_i)$求（偏）导，令导数为零，从而求出变量值 $\lambda_{i_{new}}$，然后再根据 $\lambda_{i_{new}}$ 求出 $\lambda_{j_{new}}$。
   如此一来，λ_i和 λ_j就都被更新了。
3. 多次迭代上面1-2步， 直至收敛。

具体算法步骤

1. 获得没有修剪的原始解

假设我们选取的两个需要优化的参数为 λ_1，λ_2，剩下的 $\lambda_3, \lambda_4, \dots, \lambda_N$ 则固定，作为常数处理。将SVM优化问题进行展开就可以得到(把与 λ_1, λ_2 无关的项合并成常数项 C):

$$T(\lambda_1, \lambda_2) = \lambda_1 + \lambda_2 - \frac{1}{2} K_{1,1} y_1^2 \lambda_1^2 - \frac{1}{2} K_{2,2} y_2^2 \lambda_2^2 - K_{1,2} y_1 y_2 \lambda_1 \lambda_2 - y_1 \lambda_1 \sum_{i=3}^{N} \lambda_i y_i K_{i,1} - y_2 \lambda_2 \sum_{i=3}^{N} \lambda_i y_i K_{i,2} + C$$

可能有的同学对这一步稍微有点疑惑，首先对于$\sum_{i=1}^{m} \lambda_i$部分，就只有i=1和i=2的部分是变量，其余都在C里面。

然后是$\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y_i y_j \left(x_i \cdot x_j\right)$部分，其中$\left(x_i \cdot x_j\right)$我们将它表示为$K_{i,j}$，而整体的这个式子中，我们想想那些是和我们的$\lambda_1, \lambda_2$相关的是不是有下面这些情况:

- 当i=1时j=1和j=2，i=2时j=1和j=2，这四种情况对应的就是我们的 $\frac{1}{2} K_{1,1} y_1^2 \lambda_1^2 + \frac{1}{2} K_{2,2} y_2^2 \lambda_2^2 + K_{1,2} y_1 y_2 \lambda_1 \lambda_2$，大家想一想应该就明白了，
- 当i=1时，j>=3的话，我们得到是这个式子：$y_1 \lambda_1 \sum_{i=3}^{N} \lambda_i y_i K_{i,1}$，其中要注意，$\sum_{i=3}^{N} \lambda_i y_i K_{i,1}$实际上是一个常数。
- 当然i=2，j>=3也是一样的。
- 剩余的其他部分完全和λ_1, λ_2无关，所以我们直接全扔到C里面。

于是就是一个二元函数的优化:

$$\max_{\lambda_1, \lambda_2} T(\lambda_1, \lambda_2)$$

那么根据我们的约束条件：

$$\sum_{i=1}^{m} \lambda_i y_i = 0$$

我们可以得到λ_1, λ_2的关系：

$$\lambda_1 y_1 + \lambda_2 y_2 = -\sum_{i=3}^{N} \lambda_i y_i = \zeta$$

两边同时乘上 y_1，由于 $y_i^2 = 1$ 得到:$\lambda_1 = \zeta y_1 - \lambda_2 y_1 y_2$

令 $v_1 = \sum_{i=3}^{N} \lambda_i y_i K_{i,1}, v_2 = \sum_{i=3}^{N} \lambda_i y_i K_{i,2}$，我们将$\lambda_1$的表达式代入上面得到的式子然后可以得到:

$$T(\lambda_2) = -\frac{1}{2} K_{1,1} (\zeta - \lambda_2 y_2)^2 - \frac{1}{2} K_{2,2} \lambda_2^2 - y_2 (\zeta - \lambda_2 y_2) \lambda_2 K_{1,2} - v_1 (\zeta - \lambda_2 y_2) - v_2 y_2 \lambda_2 + \lambda_1 + \lambda_2 + C$$

这样我们的参数里面是不是就剩下一个变量了，一元函数求极值就是很熟悉的内容了。

我们来进行求偏导，得到：

$$\frac{\partial T(\lambda_2)}{\partial \lambda_2} = -(K_{1,1} + K_{2,2} - 2K_{1,2}) \lambda_2 + K_{1,1} \zeta y_2 - K_{1,2} \zeta y_2 + v_1 y_2 - v_2 y_2 - y_1 y_2 + y_2^2 = 0$$

下面我们稍微对上式进行下变形，使得 λ_2^{new} 能够用更新前的 λ_2^{old} 表示，而不是使用不方便计算的 ζ。

因为SVM对数据点的预测值为: $f(x) = wx + b = \sum_{i=1}^{N} \lambda_i y_i K(x_i, x) + b$

则 v_1 以及 v_2 的值可以表示成:

$$v_1 = \sum_{i=3}^{N} \lambda_i y_i K_{1,i} = f(x_1) - \lambda_1 y_1 K_{1,1} - \lambda_2 y_2 K_{1,2} - b$$

$$v_2 = \sum_{i=3}^{N} \lambda_i y_i K_{2,i} = f(x_2) - \lambda_1 y_1 K_{1,2} - \lambda_2 y_2 K_{2,2} - b$$

已知 $\lambda_1 = (\zeta - \lambda_2 y_2) y_1$ ，可得到:

$$v_1 - v_2 = f(x_1) - f(x_2) - K_{1,1}\zeta + K_{1,2}\zeta + (K_{1,1} + K_{2,2} - 2K_{1,2})\lambda_2 y_2$$

将 $v_1 - v_2$ 的表达式代入到 $\frac{\partial T(\lambda_2)}{\partial \lambda_2}$ 中可以得到:

$$\frac{\partial T(\lambda_2)}{\partial \lambda_2} = -(K_{1,1} + K_{2,2} - 2K_{1,2})\lambda_2^{new} + (K_{1,1} + K_{2,2} - 2K_{1,2})\lambda_2^{old} + y_2\left[y_2 - y_1 + f(x_1) - f(x_2)\right]$$

我们记E_i 为SVM预测值与真实值的误差: $E_i = f(x_i) - y_i$

令 $\eta = K_{1,1} + K_{2,2} - 2K_{1,2}$得到最终的一阶导数表达式:

$$\frac{\partial T(\lambda_2)}{\partial \lambda_2} = -\eta\lambda_2^{new} + \eta\lambda_2^{old} + y_2(E_1 - E_2) = 0$$

得到:

$$\lambda_2^{new} = \lambda_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$$

这样我们就得到了通过旧的 λ_2获取新的λ_2 的表达式,λ_1^{new} 便可以通过λ_2^{new} 得到。

2. 对原始解进行修剪

上面我们通过对一元函数求极值的方式得到的最优 λ_i, λ_j 是未考虑约束条件下的最优解，我们便更正我们上部分得到的 λ_2^{new} 为 $\lambda_2^{new,unclipped}$, 即:

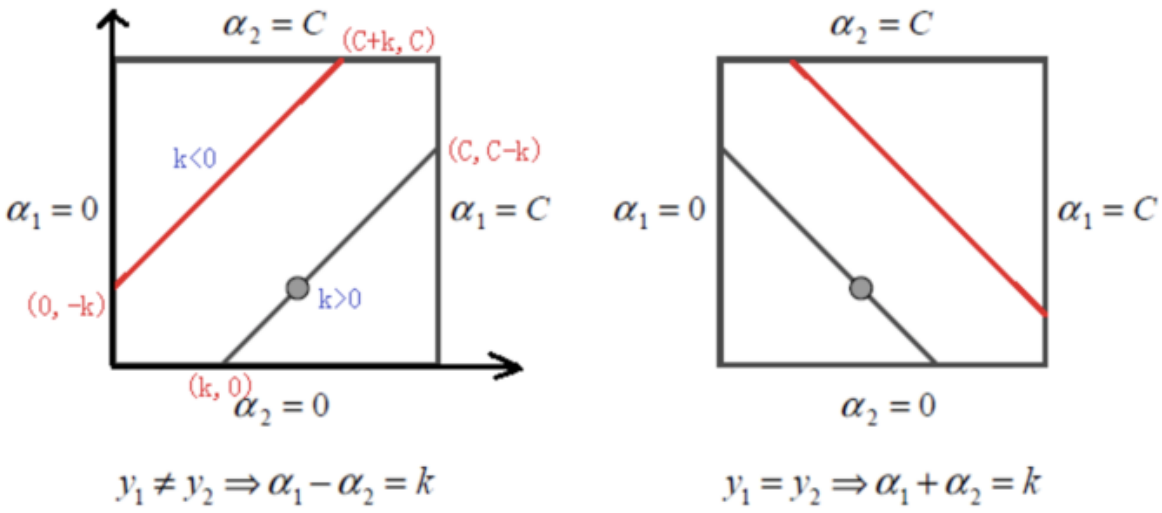
$$\lambda_2^{new,unclipped} = \lambda_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$$

但是在SVM中我们的 λ_i是有约束的，即:

$$\lambda_1 y_1 + \lambda_2 y_2 = -\sum_{i=3}^{N} \lambda_i y_i = \zeta$$

$$0 \le \lambda_i \le C$$

此约束为方形约束(Bosk constraint), 在二维平面中我们可以看到这是个限制在方形区域中的直线（见下图）。



(如左图) 当 $y_1 \neq y_2$ 时，线性限制条件可以写成: $\lambda_1 - \lambda_2 = k$ ，根据 k 的正负可以得到不同的上下界，因此统一表示成:

下界: $L = \max(0, \lambda_2^{old} - \lambda_1^{old})$
上界: $H = \min(C, C + \lambda_2^{old} - \lambda_1^{old})$

对于此处如果看图理解不了，还可以这么理解：
首先我们有$0 \leq \lambda_1 \leq C$,$0 \leq \lambda_2 \leq C$，且$\lambda_1 = \lambda_2 + k$，同样的我们就有：

$$0 \leq \lambda_2 + k \leq C$$

，也就是$-k \leq \lambda_2 \leq C - k$，我们现在取的是两个范围的交集，对于下界我们当然取最大值，对于上界我们当然取最小值。

(如右图) 当 $y_1 = y_2$ 时，限制条件可写成: $\lambda_1 + \lambda_2 = k$, 上下界表示成:

下界: $L = \max(0, \lambda_1^{old} + \lambda_2^{old} - C)$
上界: $H = \min(C, \lambda_2^{old} + \lambda_1^{old})$
根据得到的上下界，我们可以得到修剪后的 λ_2^{new}:

$$\lambda_2^{new} = \begin{cases} H & \lambda_2^{new,unclipped} > H \\ \lambda_2^{new,unclipped} & L \leq \lambda_2^{new,unclipped} \leq H \\ L & \lambda_2^{new,unclipped} < L \end{cases}$$

得到了 λ_2^{new}我们便可以根据 $\lambda_1^{old} y_1 + \lambda_2^{old} y_2 = \zeta = \lambda_1^{new} y_1 + \lambda_2^{new} y_2$ 得到λ_1^{new}:

$$\lambda_1^{new} = \lambda_1^{old} + y_1 y_2 (\lambda_2^{old} - \lambda_2^{new})$$

OK， 这样我们就知道如何将选取的一对 λ_i,λ_j进行优化更新了。

3. 更新阈值b

当我们更新了一对 λ_i, λ_j之后都需要重新计算阈值 b ， 因为 b 关系到我们 f(x) 的计算，关系到下次优化的时候误差E_i的计算。

为了使得被优化的样本都满足KKT条件，

当λ_1^{new} 不在边界，即 $0 < \lambda_1^{new} < C$, 根据KKT条件可知相应的数据点为支持向量，满足$y_1(w^T + b) = 1$ ，两边同时乘上y_1得到 $\sum_{i=1}^{N} \lambda_i y_i K_{i,1} + b = y_1$, 进而得到 b_1^{new} 的值:

$$b_1^{new} = y_1 - \sum_{i=3}^{N} \lambda_i y_i K_{i,1} - \lambda_1^{new} y_1 K_{1,1} - \lambda_2^{new} y_2 K_{2,1}$$

其中上式的前两项可以写成:

$$y_1 - \sum_{i=3}^{N} \lambda_i y_i K_{i,1} = -E_1 + \lambda_1^{old} y_1 K_{1,1} + \lambda_2^{old} y_2 K_{2,1} + b^{old}$$

当 $0 < \lambda_2^{new} < C$, 可以得到bnew2的表达式(推导同上):

$$b_2^{new} = -E_2 - y_1 K_{1,2} (\lambda_1^{new} - \lambda_1^{old}) - y_2 K_{2,2} (\lambda_2^{new} - \lambda_2^{old}) + b^{old}$$

当b_1 和 b_2都有效的时候他们是相等的, 即$b^{new} = b_1^{new} = b_2^{new}$ 。

当两个乘子 λ_1, λ_2都在边界上，且 $L \neq H$时，$b1, b2$之间的值就是和KKT条件一直的阈值。SMO选择他们的中点作为新的阈值:

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

参考

https://zhuanlan.zhihu.com/p/29212107
http://cs229.stanford.edu/materials/smo.pdf