机器学习算法系列（12）：SVM（4）—SMO

SMO算法是一种启发式算法，其基本思想是：如果所有变量的解都满足最优化问题的KKT条件，那么这个优化问题的解就得到了，因为KKT条件是该优化问题的充分必要条件。否则，选择两个变量，固定其他变量，针对这两个变量构建一个二次规划问题。这个二次规划问题关于这两个变量的解应该是更接近原始二次规划问题的解，因为这会使得原始二次规划问题的目标函数值变得更小。重要的是，这时子问题可以通过解析方法求解，这样就可以大大提升整个算法的计算速度。子问题有两个变量，一个是违反KKT条件最严重的那个，另一个由约束条件自动确定。如果SMO算法将原问题不断分解为子问题并对子问题求解，进而达到求解原问题的目的。

四、序列最小最优化算法（SMO）

通常对于优化问题，我们没有办法的时候就会想到最笨的办法，也就是梯度下降。注意我们这里的问题是要求最大值，只要在前面加上一个负号就可以转化为求最小值，所以$GradientDescent$和$GradientAscend$并没有什么本质的区别，其基本思想直观上来说就是：梯度是函数值增幅最大的方向，因此只要沿着梯度的反方向走，就能使得函数值减小得越大，从而期望迅速达到最小值。当然普通的$GradientDescent$并不能保证达到最小值，因为很有可能陷入一个局部极小值。不过对于二次规划问题，极值只有一个，所以是没有局部极值的问题。

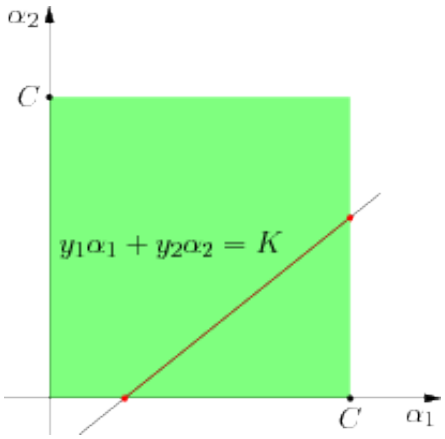
另外还有一种叫做$CoordinateDescend$的变种，它每次只选择一个维度，例如$a=(a_1,\cdots,a_n)$，它每次选取a_i为变量，而将其他都看成是常数，从而原始的问题在这一步编程一个一元函数，然后针对这个一元函数求最小值，如此反复轮换不同的维度进行迭代。$CoordinateDescend$的主要用处在于那些原本很复杂，但是如果只限制在一维的情况下则变得很简单甚至可以直接求极值的情况，例如我们这里的问题，暂且不管约束条件，如果只看目标函数的话，当a只有一个分量是变量的时候，这就是一个普通的一元二次函数的极值问题，初中生也会做，带入公式即可。

然后这里还有一个问题就是约束条件的存在，其实如果没有约束条件的话，本身就是一个多元的二次规划问题，也是很好求解的。但是有了约束条件，结果让$CoordinateDescend$变得很尴尬了，直接根据第二个约束条件$\sum_{i=1}^{N}a_iy_i=0$,a_1的值立即就可以定下来，事实上，迭代每个坐标维度，最后发现优化根本进行不下去，因为迭代了一轮之后会发现根本没有任何进展，一切停留在初始值。

所以SMO一次选取了两个坐标来进行优化。例如，我们假设现在选取a_1和a_2为变量，其余为常量，则根据约束条件我们有：

$$\sum_{i=1}^{N} a_i y_i = 0 \Longrightarrow a_2 = \frac{1}{y_2}\left(-\sum_{i=3}^{N} a_i y_i - a_1 y_1\right) \Longleftrightarrow y_2\left(K - a_1 y_1\right)$$

其中那个从3到n的作和都是常量，我们统一记作K。将这个式子代入原来的目标函数中，可以消去a_2，从而变成一个一元二次函数。总之现在变成了一个带区间约束的一元二次函数极值问题。唯一要注意的就是这里的约束条件，一个就是a_1本身需要满足$0 \leq a_i \leq C$,然后由于a_2也要满足同样的约束，即：$0 \leq y_2(K - a_1 y_1) \leq C$，可以得带$a_1$的一个可行区间，同$[0, C]$交集即可得到最终的可行区间。投影到$a_1$轴上所对应的区间即是$a_1$的取值范围，在这个区间内求二次函数的最大值即可完成SMO的一步迭代。



同$CoordinateDescent$一样，SMO也会选取不同的两个$coordinate$维度进行优化，可以看出由于每一个迭代步骤实际上是一个可以直接求解的一元二次函数极值问题，所以求解非常高效。此外，SMO也并不是一次或随机地选取两个坐标函数极值问题，而是有一些启发式的策略来选取最优的两个坐标维度。