Svm二次规划问题推导

<http://www.cnblogs.com/vivounicorn/archive/2010/12/06/1897702.html>

Step 1：

Import the Data from txt file

Step 2:

Get into the mathematic of SVM

Let the distance that a point to the separate plane be ‘d’

And let d==1, , if the point is the closest point to the hyper plane

if the point is the closest point in class ‘A’ to the plane

if the point is the closest point in class’B’ to the plane

So our target is to maximize the distance the two points to the plane, which is , in order to get the best generalization. Which is also a big advantage of SVM

(1)

An inequality constraints problem,

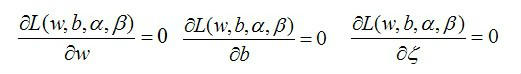
According to the explanation and prove in

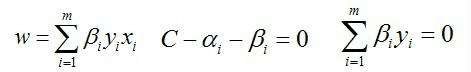
<http://www.cnblogs.com/vivounicorn/archive/2010/12/06/1897702.html>

It is a quadratic problem, the minimization problem is convex,

So the problem could be solved using the dual formulation, whose theory part has been interpreted very well in Wikipedia, especially that self-explanation graph.

The partial derivative of (1) with respect to w



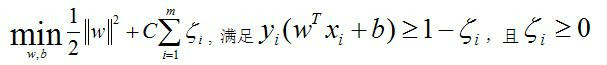


The partial derivative of (1) with respect to b

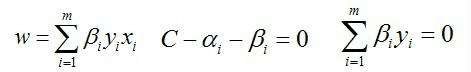
The dual problem:

Of course, it could be sensitive to noise, if there is some noise data that is very close to the separating line. We can solve this problem by introducing Slack variable.

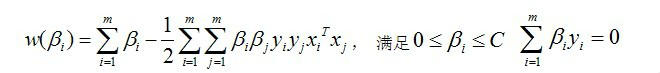
So the final problem is transformed into:



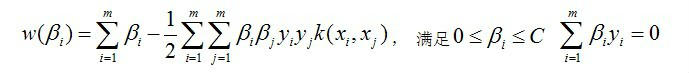
plug



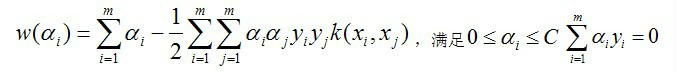
into the function above, we got this:



let’s bring in kernel:



Instead of beta, lets use alpha to be accord with the original functions

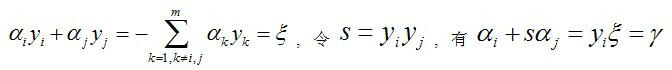


once we got alpha, w can be easily got by the partial derivative of (1) with respect to w

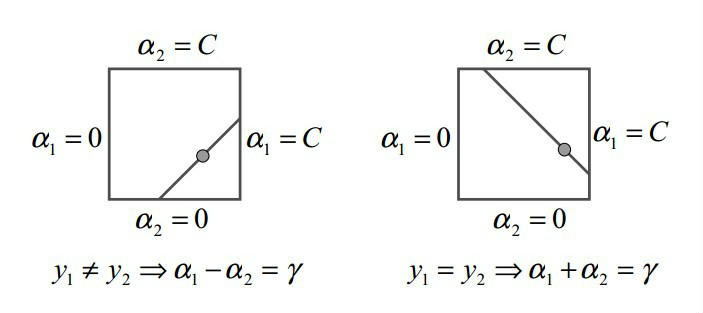
and b can be got by (we got w, any support vector points are known)

SMO algorithm work like this

Select two alpha from the alpha set, and optimize it

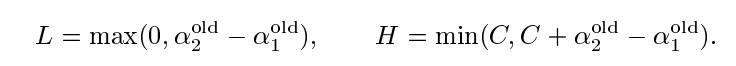


the constrains for alphas



once we know alpha2, we got alpha1, and the upper bound and lower bound are:

when y1\*y2 == -1, alpha2



when y1\*y2 = 1, alpha2



令，

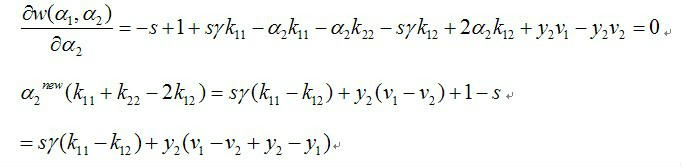
有

代入alpha1 = gama - s \* alpha2有：

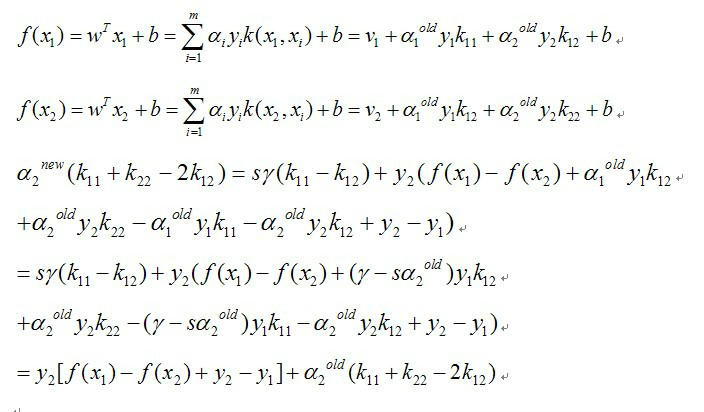


展开，令

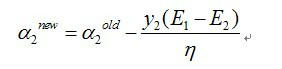
有：



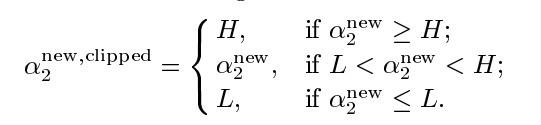
又由于：







得到了alpha2之后我们有：

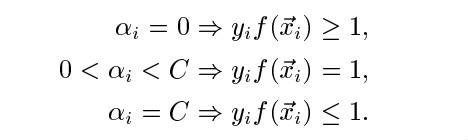


然后计算alpha1，有：



SMO always select alpha that broke the kkt condition.

The kkt conditions are like



so, we always select alphas that are big enough. Which means E1, and E2 are big eoungh.

In order to increase the tranning,