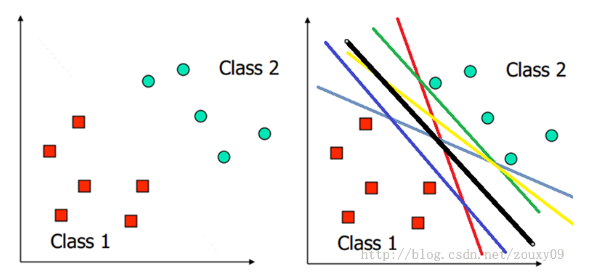
# SVM 算法

## 概述

SupportVector Machine，

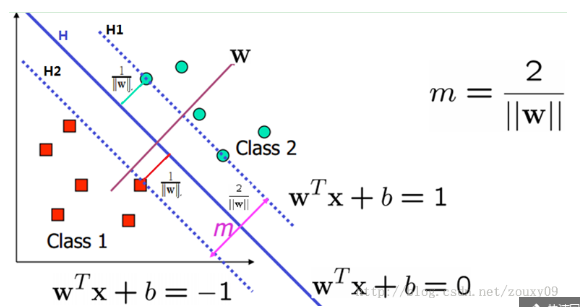
【用于分类的支持向量机，是一个二分类模型，给定一个包含正例和反例的样本集合，SVM寻找一个超平面来对样本进行分割——是的正例和反例之间的间隔最大】



二维空间：线， 三维空间：面， 多维：超平面

## 线性可分割svm和硬间隔最大化

SVM：尽最大努力令正例和反例的间隔（margin）最大——结果才可信（泛化能力）



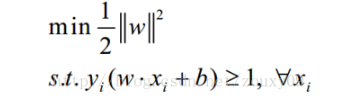
有N个训练样本{(**x**1, y1),(**x**2, y2), …, (**x**N, yN)}，**x**是d维向量，而yi∊{+1, -1}是样本的标签，分别代表两个不同的类，训练一个线性分类器：

f(**x**)=sgn(**w**T**x**+ b)———

寻找2个面，与该超平面机距离相等【条件：没有任何样本在两个平面之间；两个平面距离要最大】

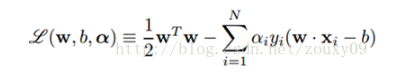
对于任何一个正样本yi=+1，它都要处于H1的右边，也就是要保证：y= **w**T**x**+ b>=+1。对于任何一个负样本yi=-1，它都要处于H2的左边，也就是要保证：y = **w**T**x**+ b<=-1。这两个约束，其实可以合并成同一个式子：yi (**w**T**x**i + b)>=1。

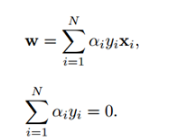
【问题转换】凸二次规划问题

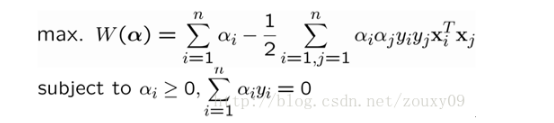


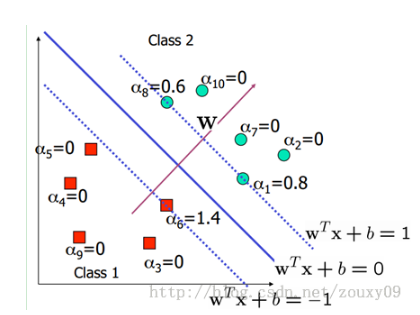
通过线程的QP（Quadratic Programming）优化工具得到最优解——拉格朗日对偶问题（约束最优化）

引入拉格朗日乘子



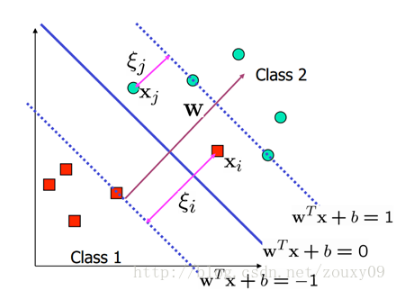




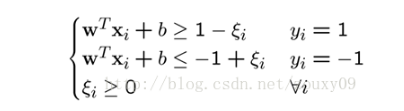


偏离正常位置很远的数据点：outlie【采集训练样本的噪声可能，可能是出错】

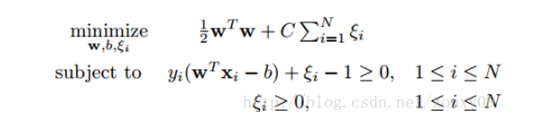
处理:允许一些数据点在一定程度偏离超平面，

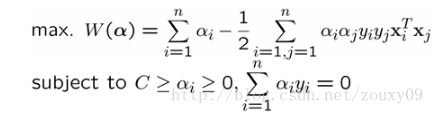


原来的约束条件变成



加一个惩罚项：



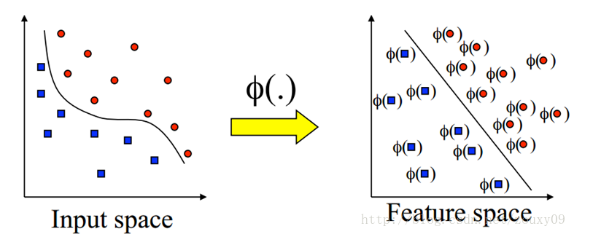


对偶优化问题。没有了参数ξi，与之前模型唯一不同在于**αi**又多了**αi**<=C的限制条件。需要提醒的是，b的求值公式也发生了改变，改变结果在SMO算法里面介绍。

## 核方法（kernel trick）

Svm对线性分割数据有效，对不可分的对策。

1. 首先使用一个非线性映射Φ(**x**)将全部原始数据**x**变换到另一个特征空间，在这个空间中，样本变得线性可分了；
2. 然后在特征空间中使用SVM进行学习分类

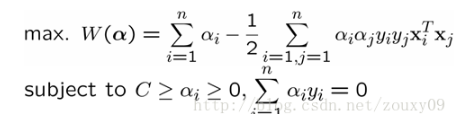


问题：

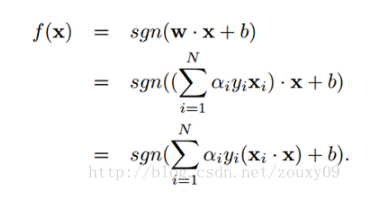
1. 哪些变换才能把数据映射为线性可分的呢?

Cover定理：找到一个所有样本映射到更高维空间的映射——核函数

Svm优化问题：



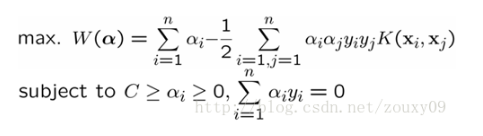
    可以看到，对样本**x**的利用，只是计算第i和第j两个样本的内积就可以了。



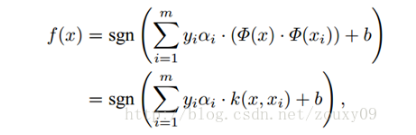
核函数：K(**xi, xj**)=Φ(**xi**)T Φ(**xj**)，   也就是两个样本**xi**和**xj**对应的高维空间的内积Φ(**xi**)T Φ(**xj**)通过一个核函数K(**xi, xj**)计算得到。

RBF

优化对偶问题变为：



决策函数变成：



**支持向量机的基本思想可以概括为，首先通过非线性变换将输入空间变换到一个高维的空间，然后在这个新的空间求最优分类面即最大间隔分类面，而这种非线性变换是通过定义适当的内积核函数来实现的。SVM实际上是根据统计学习理论依照结构风险最小化的原则提出的，要求实现两个目的：1）两类问题能够分开（经验风险最小）2）margin最大化（风险上界最小）既是在保证风险最小的子集中选择经验风险最小的函数。**

### 三、多分类SVM

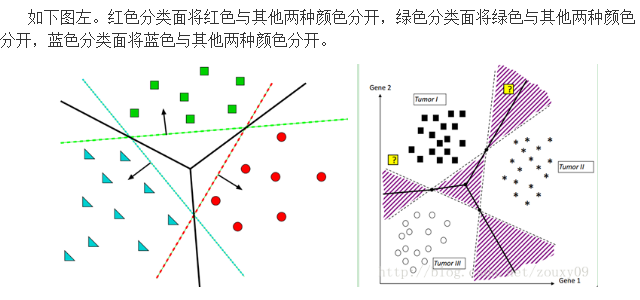
6.1”一对多”的方法

要分为5类：第一次：是1类的，不是1类的

第二次：是2类的，不是2类的

第三次：是3类的，不是3类的

......



3.2”一对一”的方法

增加分类器，

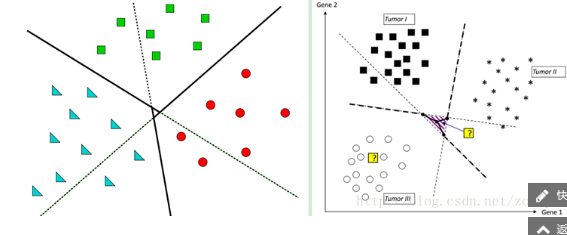
第一个分类器：“是第一类还是第二类”

第二个：‘1/3’

第三个：“1还是4”

K类别：分类器数目: k(K-1)/2

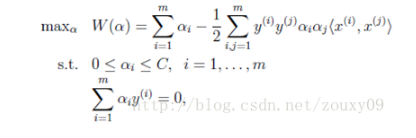
。。。。。



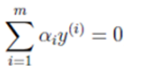
### Smo算法

http://blog.csdn.net/zouxy09/article/details/17292011

SMO是一次迭代优化两个α而不是一个。为什么要优化两个呢？



我们可以看到这个优化问题存在着一个约束，也就是



我们需要一次选取两个参数做优化，比如αi和αj，此时αi可以由αj和其他参数表示出来。这样回代入W中，W就只是关于αj的函数了，这时候就可以只对αj进行优化了。在这里就是对αj进行求导，令导数为0就可以解出这个时候最优的αj了。然后也可以得到αi。

这就是一次的迭代过程，一次迭代只调整两个拉格朗日乘子αi和αj。SMO之所以高效就是因为在固定其他参数后，对一个参数优化过程很高效（对一个参数的优化可以通过解析求解，而不是迭代。虽然对一个参数的一次最小优化不可能保证其结果就是所优化的拉格朗日乘子的最终结果，但会使目标函数向极小值迈进一步，这样对所有的乘子做最小优化，直到所有满足KKT条件时，目标函数达到最小）。

重复下面过程直到收敛{

（1）选择两个拉格朗日乘子αi和αj；

（2）固定其他拉格朗日乘子αk(k不等于i和j)，只对αi和αj优化w(**α**);

（3）根据优化后的αi和αj，更新截距b的值；

}