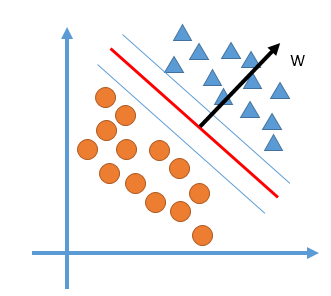
**从0到1支持向量机**

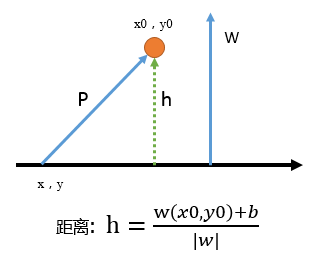
# 预备知识

1、【法向量】Wx+b=0，w是什么？

Wx+b=0是直线方程，其中w表示法向量，法向量的指向由具体值确定。例如x+y-2=0，法向量为(1,1)，指向右上方。



2、【距离公式】



3、【函数间隔】

当w确定的时候，距离的远近可以比较分子，也就是说wx+b的绝对值表示样本点到分类平面的相对距离，显然点在w指向的方向距离为正值，规定正类为这一侧，同样的分析应用到负类，那么y(wx+b)即可以表征分类准确性，又可以表征分类的置信度，具体而言绝对值的大小表示置信度，正负分别表示分类正确、错误，在svm中这被称作函数间隔。

4、【几何间隔】

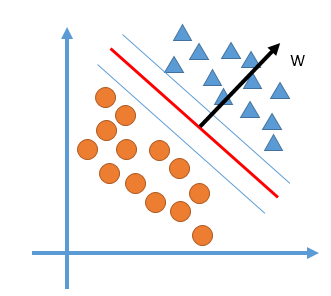
可以理解为真正的距离，与函数间隔是正比关系。

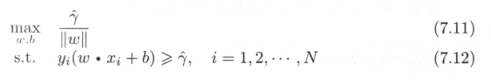
5、【核技巧】

6、【拉格朗日对偶性】

7、【hinge loss function】

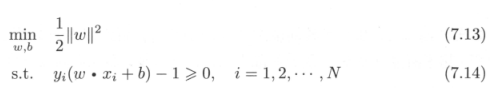
# 7.1 线型可分支持向量机





**解释**：γ是函数距离，γhat所有样本点中距离分类面最小的函数距离。由前置知识可以知道，目标函数是几何间隔最大化。

其中γ hat为到超平面最小的函数间隔，这时的超平面还是一个任意的平面，显然这是一个极大极小问题。因为优化的目标是找到最佳的自变量w,b的比例关系以确定分离超平面，而与变量的绝对大小无关，基于此，我们把函数间隔设置为1.优化目标变为：



满足所有点的函数间隔大于1，最小化w的模长，这时的w和b就是我们的目标值。其实是1还是2都没关系，这个数值是为支持向量准备的函数间隔的值，相当于设定了参考点，别的样本点的函数间隔都应该大于等于该值。

假如未能满足的话就会形成梯度对w和b优化。直到所有样本点满足约束。

能满足约束条件的参数组不止一组，例如w1，b1满足，现在将他们全部放大2倍，仍然满足，为了避免无意义的多解，目标函数对参数的2范数进行约束。使得解是唯一的。

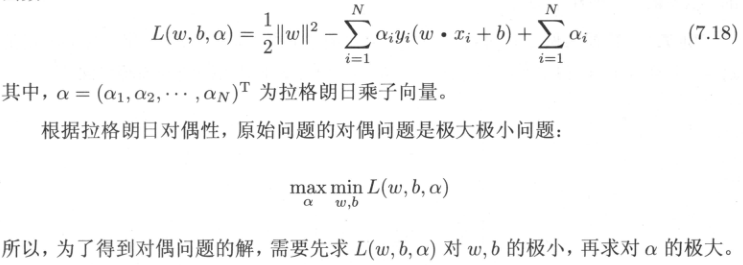
考虑一种比较有意思的情况：假如初始化已经满足了所有的约束条件，但是并没有处在分类中间，接下来会怎样呢？假如两边的支持距离分别为1和2,2说麻烦把b调一下，让我的距离小一点，之后两边的距离变成呢1.2,1.5，指挥部发现这样的话可以把参数缩小1.2倍，正和我意，于是距离变成了1，1.25，之后repeat直到两边的距离都是1。

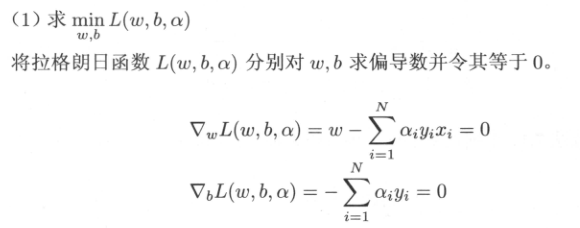
上面是一种情况，还有其他很多种，但是无论如何，最终都是只要w可以优化模型就会不断朝间隔最大化移动。因为：

**几何距离1 \*|w1| = 函数距离（都是1） = 几何距离2 \* |w2|**

几何距离越大，才能使得w的模长越小。

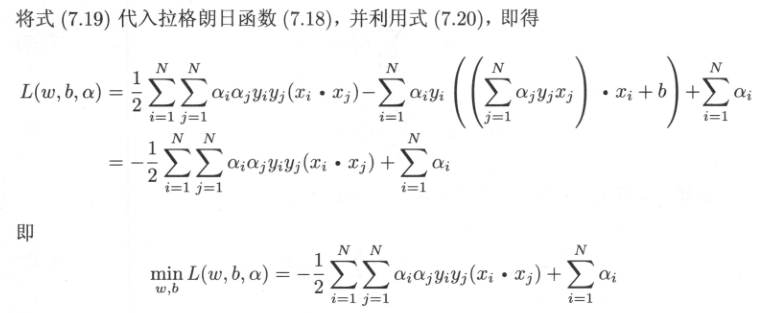
不太严谨的讲：在模型没有满足约束的时候，约束条件使得模型不得不调整到可以正确分类的位置，之后基于最小化范数，模型逐渐调整到中间的位置。以上只是帮助理解的，下面是解决方案：基于拉格朗日函数与对偶算法：构建target：



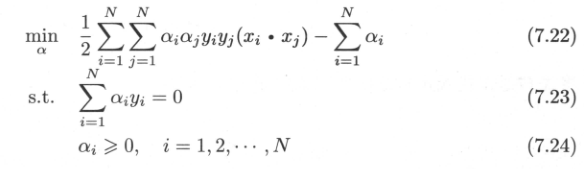




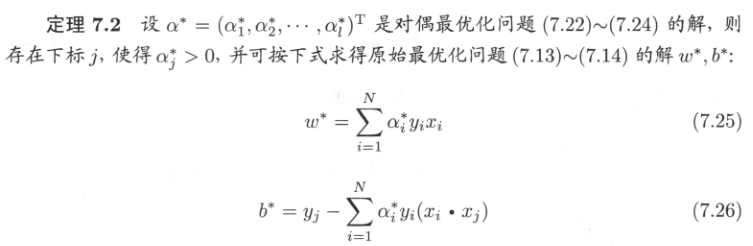




问题转化为：



基于当前的优化目标和约束条件求出α。就相当于在一个高维空间求最小值，然后附带了很多条件。



也就是说，现在问题的焦点在于α的求取。这已经是单变量了，可以直接求解（解析法，梯度法*（需要max（0，步进后的值）*）

# 7.2 线性支持向量机与软间隔最大化

由于数据集已经线型不可分，所以如果仍然要求所有点的函数间隔大于等于某一个正数已经不可能，也就是说每一个样本点需要设置一个专属的函数间隔，ξ为引入的松弛变量（可能大于1），约束条件为：



对松弛变量的理解：引入松弛变量在数学上等价于改变样本点的坐标。

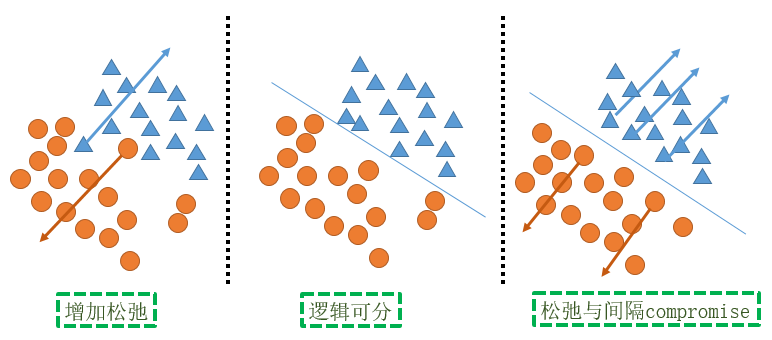
yi（w \* xi + b）+ yi（w \* Δx+ b）>=1，

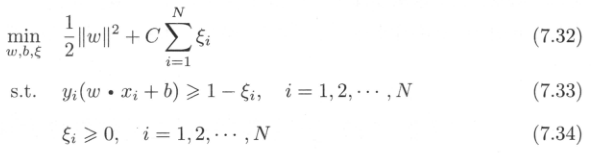
ξ = yi（w \* Δx+ b）

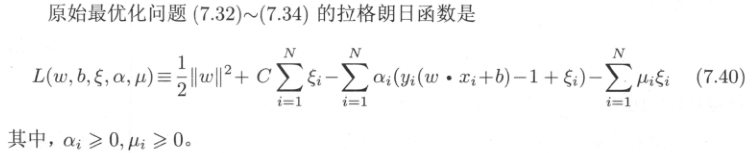
本来样本是不可分的，引入松弛后，变成可分的数据集。至于那些引入，这个不用我们管，模型优化的过程会自动的确定。所以目标函数仍然包含极小化w2，那么对ξ有没有要求呢？当然有，比如抗战时期，国共敌三者人员都是相互渗透的，现在想对敌我分分类，就需要对国军和敌后工作者发放松弛变量，不然就把它们错分为敌人了，但是如果对所有人发放松弛，那就会导致敌我不分。从这个角度，我们希望松弛绩极小。

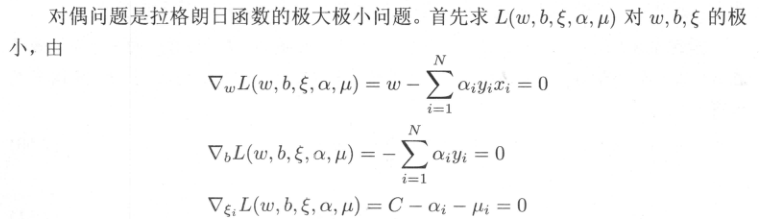
这里要区分逻辑分类结果和实际分类结果，实际上模型是不可分的，无论分类面在哪里，都没有分开，但是引入松弛后在逻辑上，实际上已经分开了。

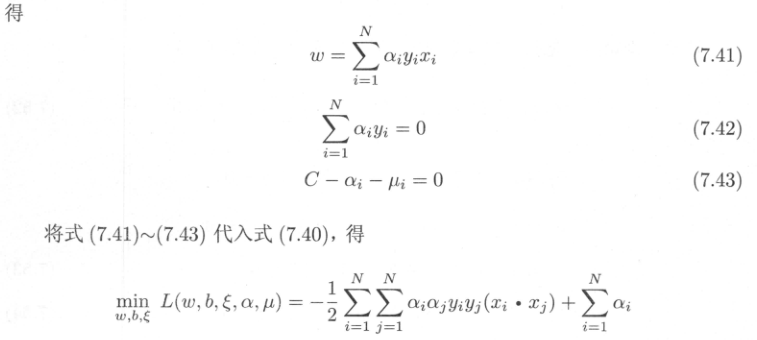
在逻辑可分的情况下，松弛发放的越少，分类间隔越小。继续发放松弛，逻辑上的几何间隔可以变大。即为了追求间隔最大，我们希望扩大松弛，但是松弛又是我们想极小化的，所以，现在不再单纯的追求间隔最大化或者松弛最小化，而是两者的和最小。引入松弛的过程可以理解为改变真正sv的过程，而sv决定间隔。



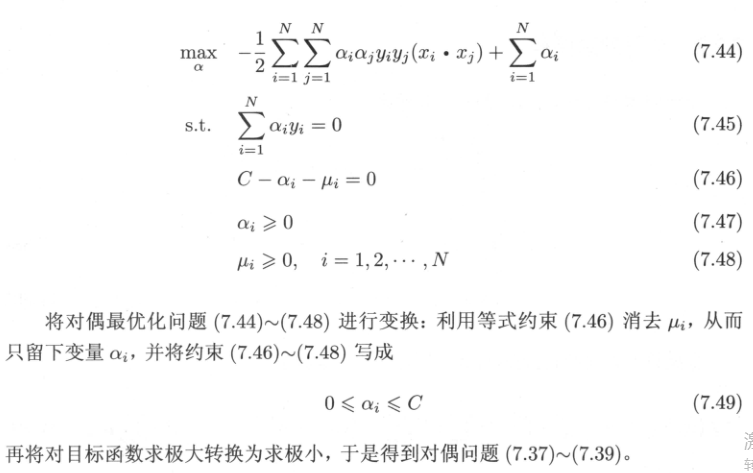


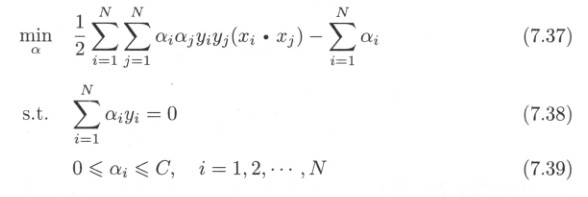


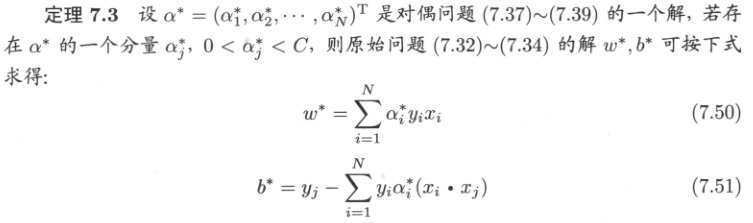




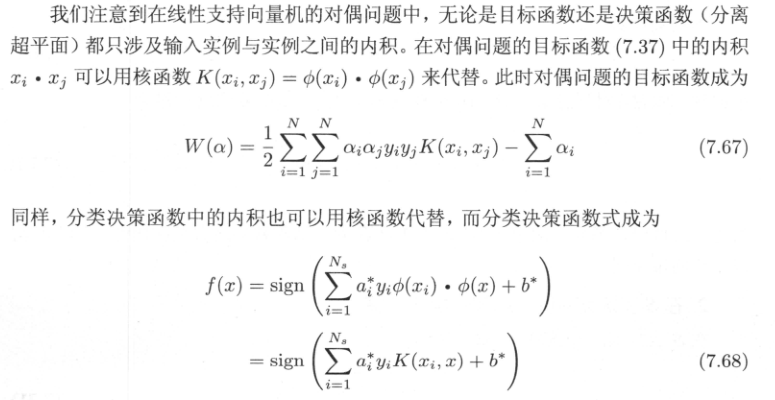








# 7.3 非线性支持向量机与核函数

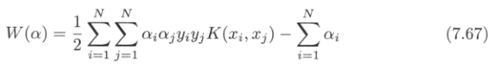


卧槽，发生了什么？怎么就非线性了？

在线型支持向量机的背景下，无论是目标函数，还是决策函数。关键部分都是两个向量的内积。但是在非线性的背景下，这些是不成立的，本质上是因为线型不可分，没有之前讨论的函数间隔，自然没有办法给出优化目标。

但是，设想一下，把现在的空间做一次映射，变成了线型可分的情况，而直接借用之前的目标函数的形式，那么目标函数仍然是向量内积的形式（映射后的向量），即特征空间的向量内积，但是没关系核函数的存在为两个空间建立了通道。

以下式为例进行理解：



可以理解为这是特征空间的目标函数。刚开始看这一部分的时候可能会很疑惑为什么直接从输入空间的内积变成特征空间的内积了？这会相等吗？其实，只是特征空间套用了输入空间的目标函数的模板而已。所以不存在相不相等的问题。而这个模板最关键的部分就是内积，核函数就是特征内积。

利用这个目标函数求α。

另外，从决策函数可以看出，分两类结果也要映射到特征空间进行判定。在整个过程中，我们可以这样看待：

特征空间是进行判定的空间，输入空间是数据源，而核函数决定了输入向量映射到特征空间所处的位置。

特征空间是前端，输入空间是后端，核函数是控制层

为什么映射函数可以不显式的给出？

因为我们关注的不是特征空间向量的具体形式，而是其内积，而内积有时等于输入空间的某种运算，所以直关注输入向量基于核函数在输入空间的运算结果即可。

怎样根据问题背景选择核函数，不都是用来非线性分类吗，难道是考虑输入空间的分布？

目前的想法：假如我们能知道映射函数的表达式，就可以根据映射函数写出核函数。如果不能确定映射，可能就采用实验的方法测试。

