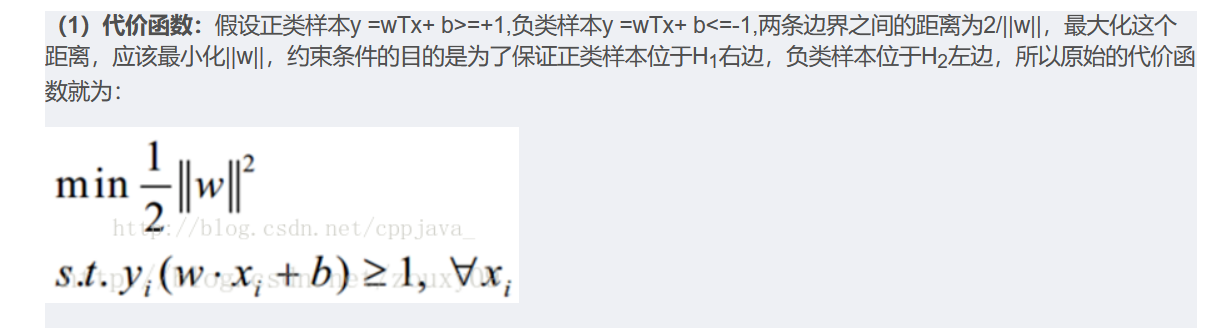
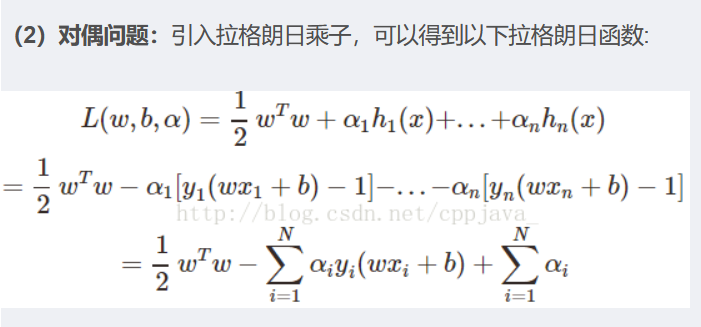
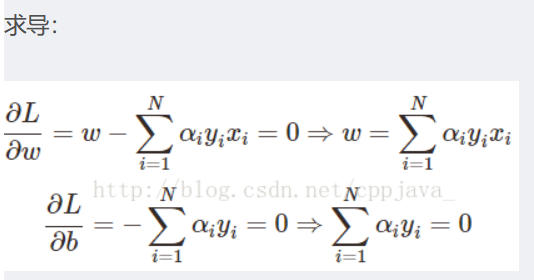
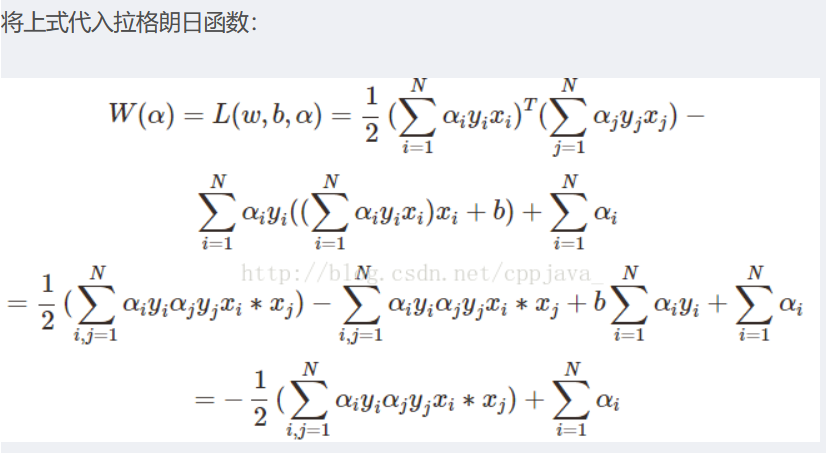
**Svm的基本原理**

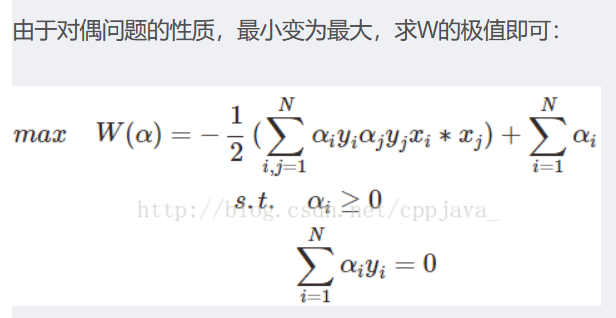
支持向量机为一个二分类模型, svm的目标是寻找一个最优的分离超平面, 将两类数据在空间中分离开来，并且使得这个超平面到最近的点的间隔最大，这些点称为支持向量。



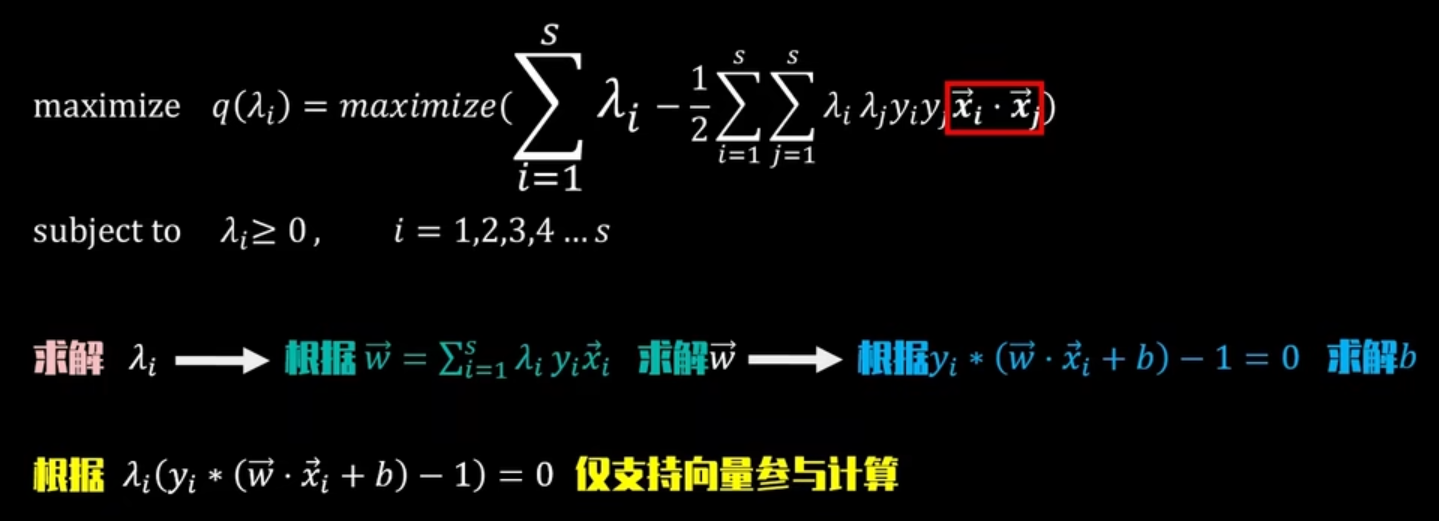




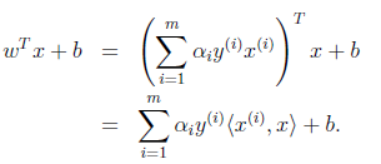




SMO求得使函数最大的α, 通过KKT条件中的



如果使用了核函数(RBF核函数), 由于RBF核函数相当于无限维度的W, 在计算λ时直接计算核函数的值, 无需计算升维特征的内积. 无需解出具体的W向量, 直接通过

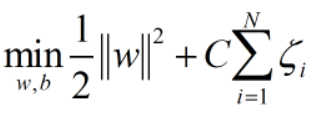


解出b, 新来样本x的话，只需将[clip_image057[4]](http://www.javashuo.com/link?url=http://images.cnblogs.com/cnblogs_com/jerrylead/201103/201103182034447701.png)替换成[clip_image059[6]](http://www.javashuo.com/link?url=http://images.cnblogs.com/cnblogs_com/jerrylead/201103/201103182034443765.png)，而后值的判断同上。

1. **为什么要转为对偶问题?**
2. 对偶问题往往更容易求解。当我们寻找约束条件下的最优点时，约束的存在虽然减少了需要搜寻的范围，但是却使问题变得更加复杂。我们的方法是把目标函数和约束全部融入一个新的函数， 即拉格朗日函数，在通过这个函数来寻找最优点；
3. 方便引入核函数。（因为对偶问题涉及的是数据的内积计算）进而推广到非线性分类问题。
4. **松弛向量与软间隔：**

原因： 一些离群点或者噪声点影响分界面；

解决方法： 允许某些样本不满足约束来换取更大的分类间隔。引入了松弛因子, 直观上看是一个分段线性函数, 在样本点上y=1的右侧值为0; 引入惩罚因子C,最后的目标函数:

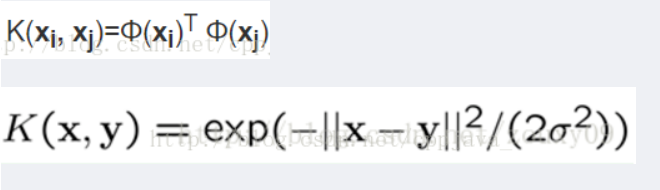


此时优化函数可以划分为相互制约的两部分, 当C越大时, 代价函数对松弛因子的容忍度更小, 放宽了对W的优化, 分类间隔就会变小, 容易出现过拟合。相反的， 如果C比较小, 那么间隔就会变大, 模型也就有了更好的泛化能力。

1. **核函数**

原因： 原始空间线性不可分， 可以使用一个非线性映射将原始数据X变换到另一个高维特征空间, 样本变得线性可分。

解决方法： 常用的一般是径向基RBF函数(线性核, 多项式核, 高斯核等)。



优点： 避免了高维空间的计算，计算核函数的复杂度和计算原始样本内积的复杂度没有实质性的增加。

1. **为什么要引入核函数？**

当样本在**原始空间中线性不可分**时， 可将样本从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内**线性可分**， 而 引入这样的映射后， 通过对偶问题来求解， 无需求解真正的映射函数， 而只需知道其核函数。核函数就是特征映射后的内积， 在学习和预测中只需要定义核函数而不显式地定义映射函数， 从而降低计算的难度。

1. **多分类**

主要通过组合多个二分类器来实现多分类器的构造，常见的方法有两种。

1. 一对多

训练时依次把某个类别的样本归为一类, 其他剩余的样本归为另一类, 这样K个类别的样本就构造出了k个SVM。分类是将未知样本分类为具有最大分类函数值的那类。

缺点:

1. 每个分类器的训练都是将全部的样本作为训练样本，这样在求解二次规划问题时， 训练速度会随着训练样本的数量的增加而急剧减慢；
2. 由于负类样本的数据要远远大于正类样本的数据，从而出现了样本不对称的情况。
3. 当有新的类别加入时，需要对所有的模型进行重新训练。
4. 一对一

其做法是在任意两类样本之间设计一个SVM, 因此K个类别的样本就需要k(k-1)/2个SVM, 当对一个未知样本进行分类时, 最后得票最多的类别即为该未知样本的类别。

优点： 不需要重新训练所有的SVM, 只需重新训练与增加样本相关的分类器。在训练单个模型时，相对速度较快。不会加剧正负样本的不平衡性。

缺点：所需构造和测试的二值分类器的数量关于K成二次函数增长, 总训练时间和测试时间相对较慢。

1. **不平衡样本的处理方法**
2. 改变分类阈值， 使分类结果更偏向于样本少的一类
3. 改变样本类别的权重
4. 对样本点少的正类采用较大的惩罚因子C。

（4） 采用数据合成方法进行过采样（[SMOTE](https://blog.csdn.net/The_dream1/article/details/115582791)合成少数类过采样技术）

1. **K折交叉验证**

原因: 数据有限, 单一的把数据都用来做训练模型, 容易导致过拟合, 需要在验证集做评判测试, (反过来, 如果数据足够多,完全可以不使用交叉验证。) 较小的K值会导致可用于建模的数据量太小, 所以数据集的交叉验证结果需要格外注意, 建议选择较大的K值, K一般取2-10。

1. **网格搜索C与gama**

在对数尺度上等间距设置C与gamma, 分别进行训练, 根据评价指标(查准率, 查全率, F1分数)选择最合适的参数,

1. 总结:

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。

扩展:

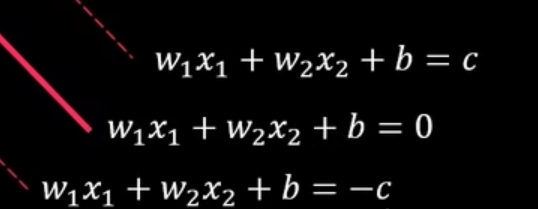
1.ganma:单个样本作为支持向量机的影响半径,且半径的大小与ganma的值成反比。如果γ值很大，那支持向量的影响半径就很小，它并不能把其他与它距离较远的样本很好的分类，那么超出它半径范围的样本也要成为支持向量，模型就会更加的复杂，更容易过拟合。

SVM详细解

支持向量机为一个二分类模型, svm的目标是寻找一个最优的分离超平面, 将两类数据在空间中分离开来，并且使得这个超平面到最近的点的间隔最大，

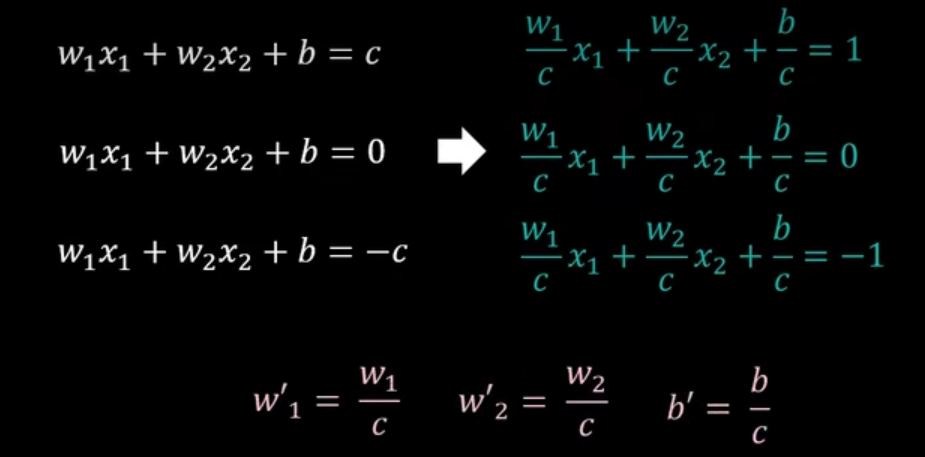
1. **决策超平面定义**

假设决策超平面为W1X1+W2X2+B=0;间隔上下边界就是

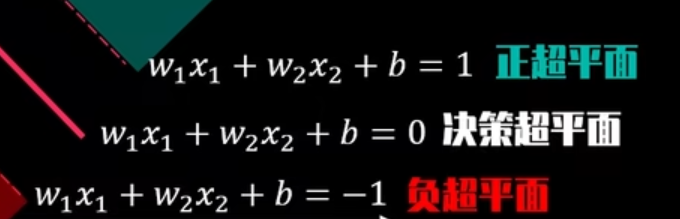


由于上下边界一定会经过一些样本数据点， 而这些点距离决策边缘最近，它们决定了间隔距离，我们称它为支持向量 support vector,这也是为什么我们将该方法命名为支持向量机的原因 。

我们将等式两边除以C使得等式右边为1，



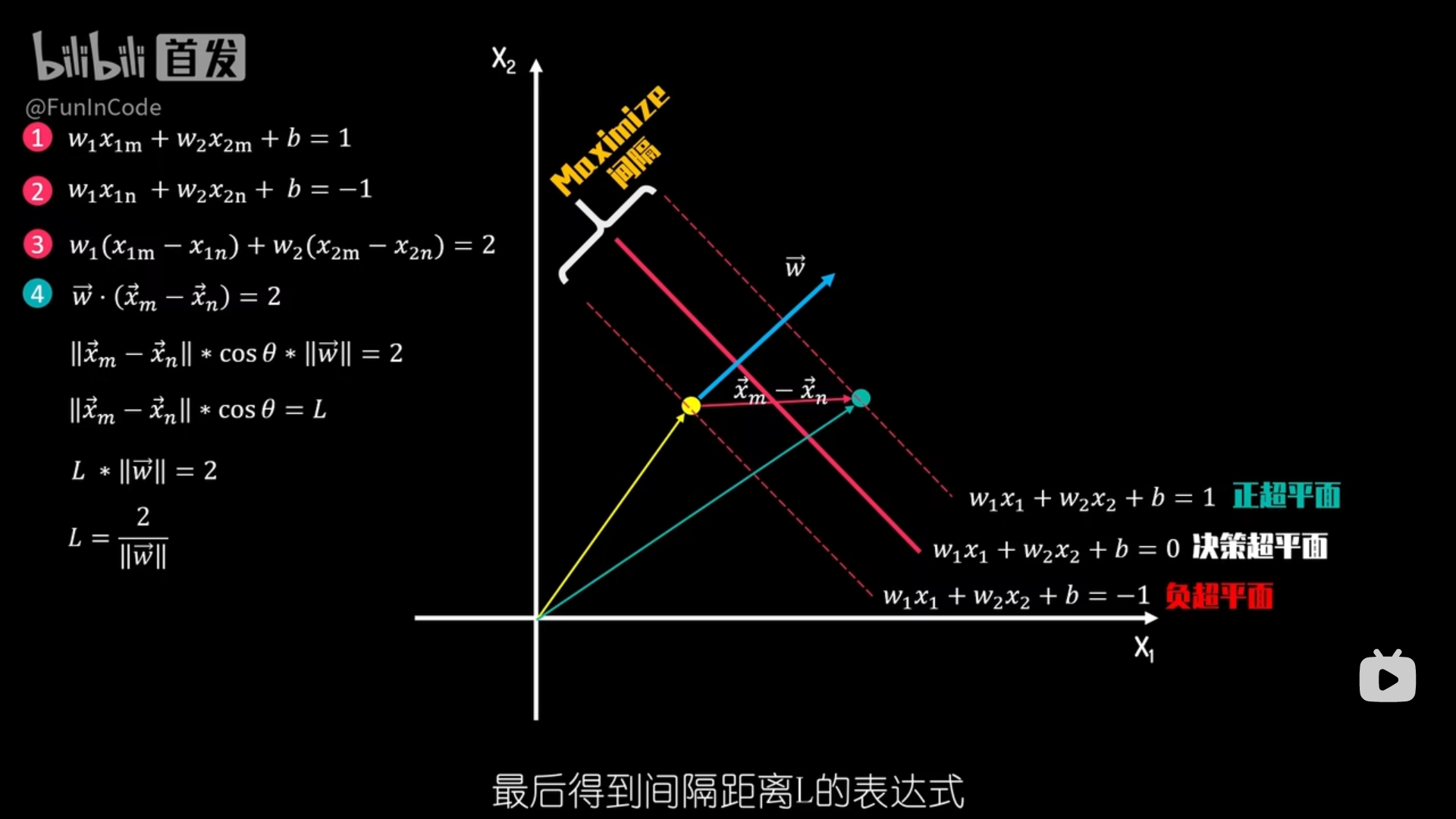
得到超平面方程式：正超平面，决策超平面，负超平面。

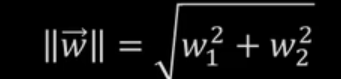
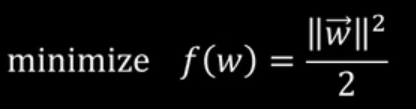


1. 求解SVM决策超平面

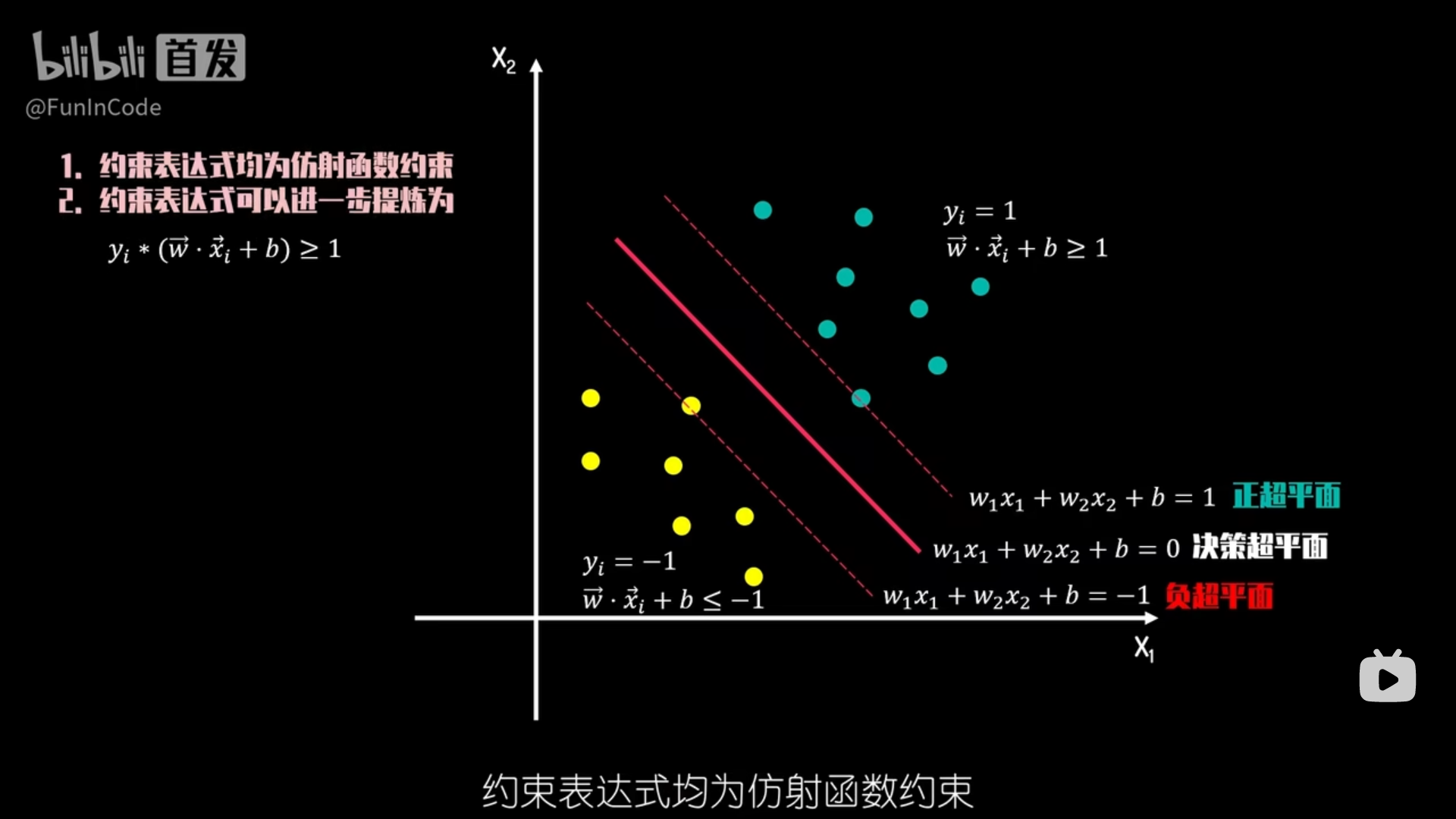
**间隔距离L的表达式：**

决策超平面的系数向量W是垂直于超平面的，利用向量的点积可以求得**L=2/向量W的长度**;使得间隔距离最大就是使向量w长度的最小化问题，w 的长度等于各分量的平方和开根号，为了把**根号去除**，并优化未来计算，我们可以对目标表达式先平方再除2 ，也就是w长度的平方/2。

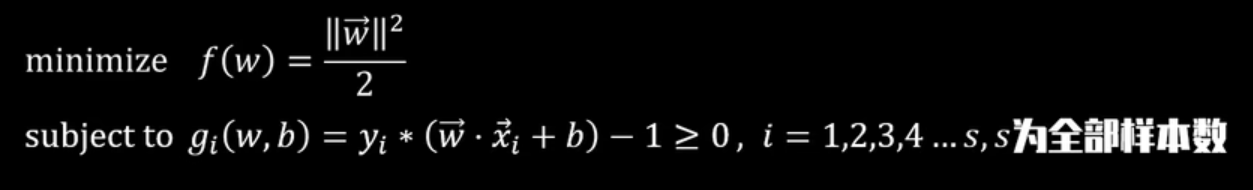


令正类为1，父类为-1，这样约束表达式可以进一步提炼为**y\*(w \*x+b)>=1**;

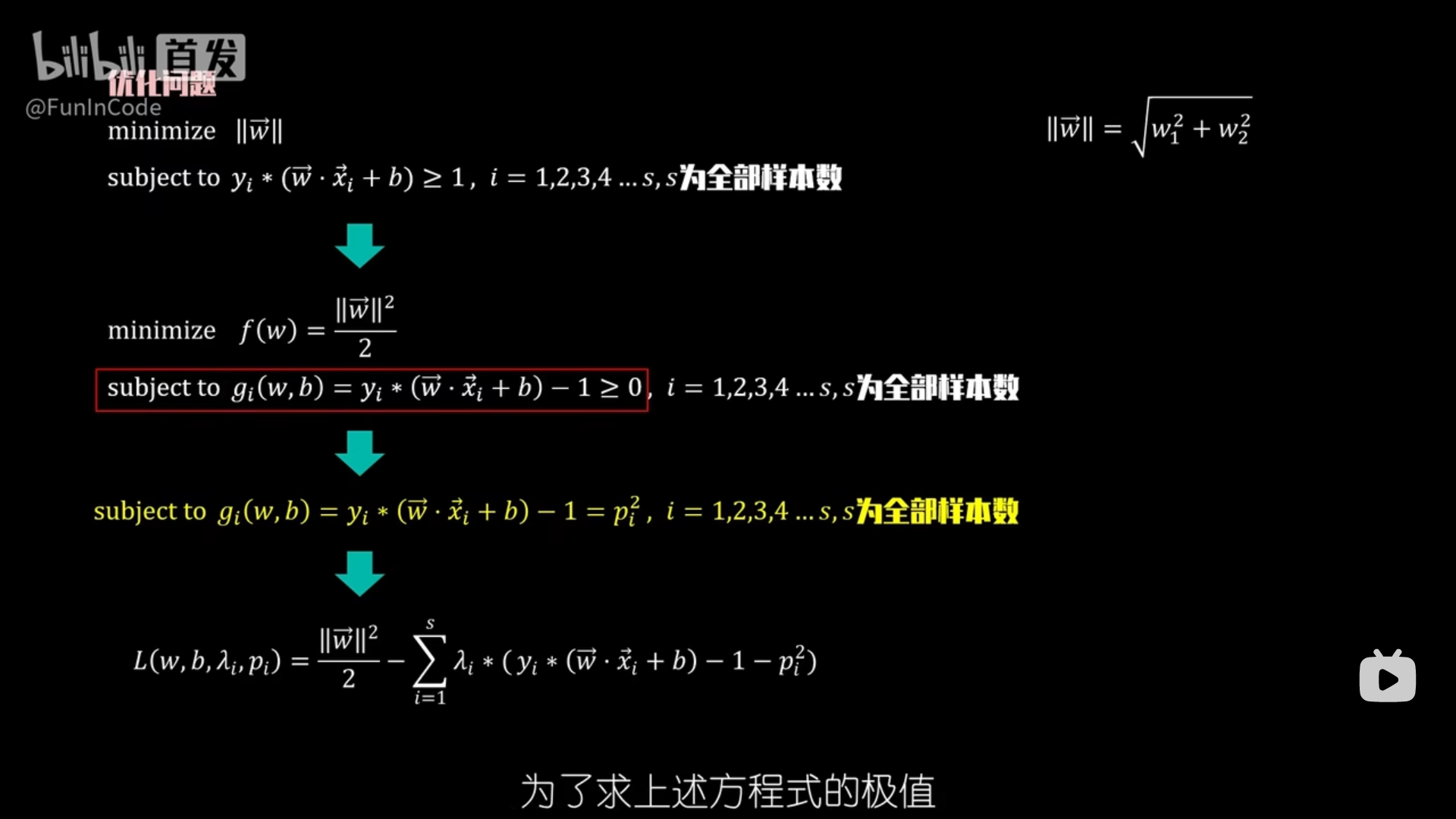


到此，可以得到最终的目标优化问题：

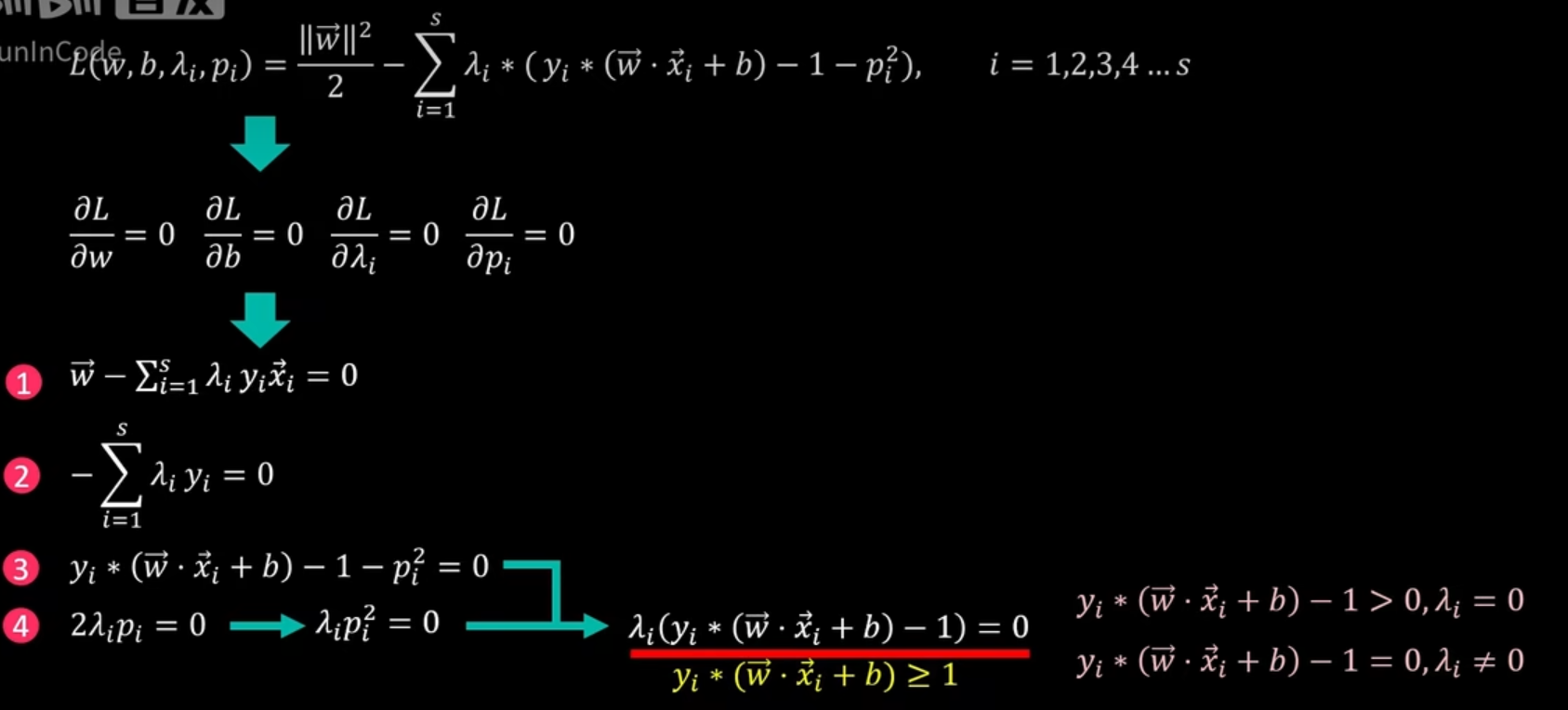


**求解KKT条件**

如果约束条件是等式的话， 我们可以直接使用拉格朗日乘子法求解，但是约束条件是不等式的话，通过增加一个非负变量pi的平方 ，将不等式转化为等式约束，再使用拉格朗日乘子法，



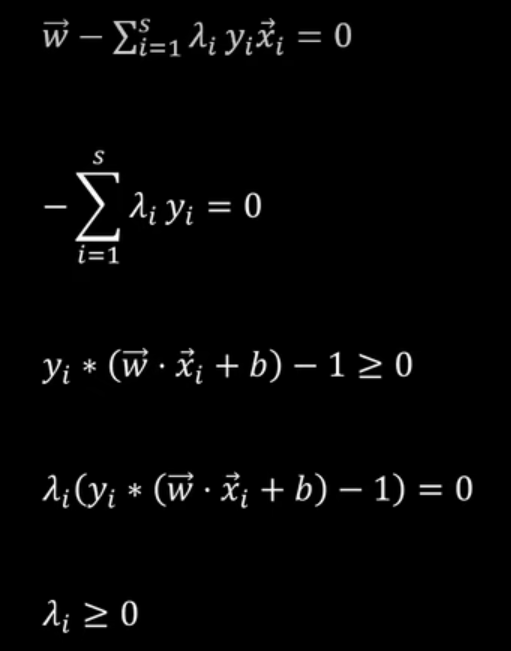
为了求上述方程式的极值，我们分别对w,b,λi,pi分别求偏导，消去pi,



并推出λi>=0,



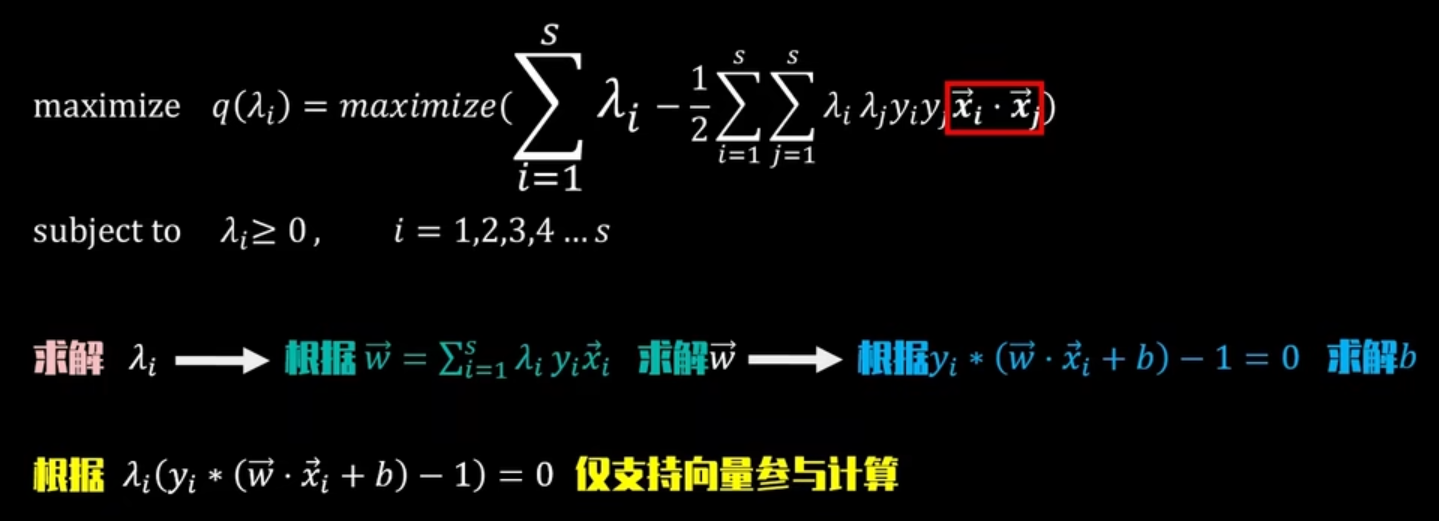
最后我们就得到五个条件，也就是KKT条件



基于KKT条件就可以求解最终的决策超平面了；

不过在svm中为了提高求解的效率和更方便的应用核技巧kernel trick。 **我们往往会将原问题转换为自身的对偶问题，再求解，在强对偶成立时，原问题与对偶问题的最优解同时得到解决。**几乎所有的凸优化问题都满足强对偶性, 比如SVM中，**它的损失函数是凸函数，**我们几乎就能断定它一定是一个强对偶的问题。

先对最小化部分求解，代入KKT条件，得到更精简的优化方程，求解λi后，根据KKT条件中W等式求解W, 只有正负超平面上的λi不为零，其余皆为零，所计算时只用到支持向量，计算量显著减少。W求得后，代入正负超平面方程即可求得b了。



之所以我们绕了这么大一圈去求对偶问题，除了对偶问题从表达式与约束条件更简洁以外，更主要的原因是对偶问题有一个非常好的特征，也就是最优解仅根据**支持向量**的点积结果所决定，方便引入核函数。

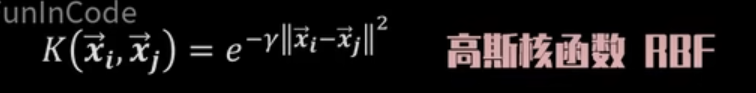
**核函数**

**对偶问题的λi求解**取决于1.每个数据点的正负标签Y值，2.数据点向量之间的两两点积结果，当遇到非线性分类问题时，原维度下决策超平面下无解，此时将维度转换函数T(x),将原维度数据进行升维转换，在高维空间中求解决策超平面。此时需要求解新维度下T(x),T(y)的点积结果，这里有两种方法，方法一，定义相应的维度转换函数T， 对数据完成维度转换向量后再求新维度向量的点积，方法二，直接套用核函数，通过原维度下的向量点积结果，直接计算出新维度下的向量点积结果。

线性核函数，

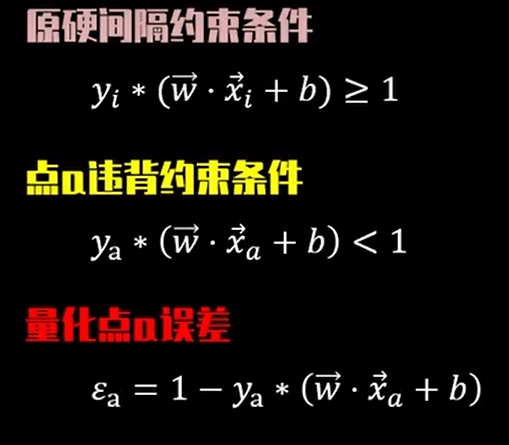
多项式核函数

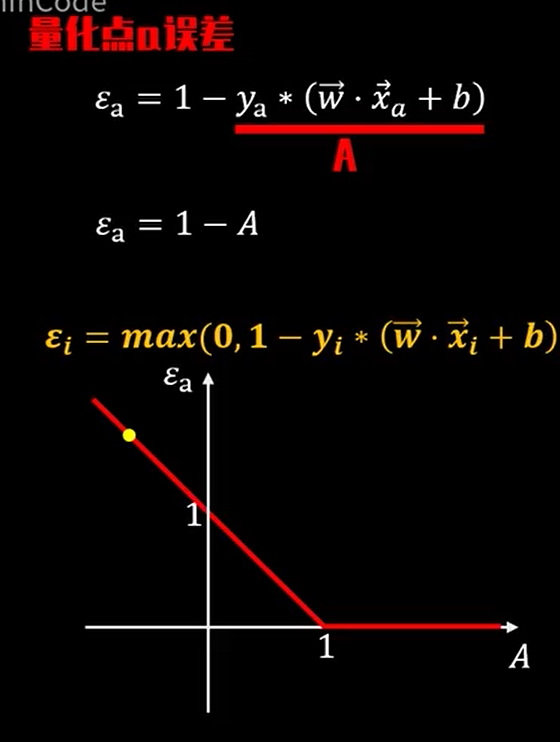
高斯核函数RBF

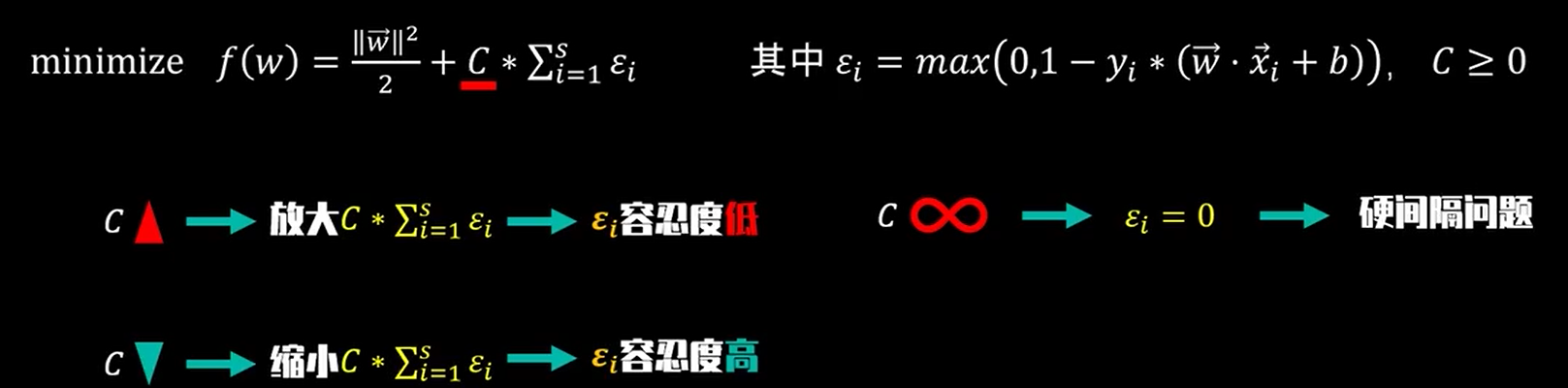


**软间隔：**

假使我们的初始数据发生了如下的变化, 如果我们忽略中间的异常点,间隔距离也会明显的增加.那么我们需不需要为这样一个异常值来牺牲我们的间隔距离呢?当有更多的异常值呢?所以我们可以引入损失因子这个概念.那些违背规则的损失点都会有对应的损失值,你可以把间隔距离看成经营收入, 而损失值想象成经营成本,我们的目标则转化为同时考虑”收入”与”成本”的因素去最大化我们的利润.这个最优解下形成的间隔称为软间隔(soft margin)。它有一定的容错率，目的是在**最大化间隔距离与最小化错误大小间**找到一个平衡。







**数学知识：**