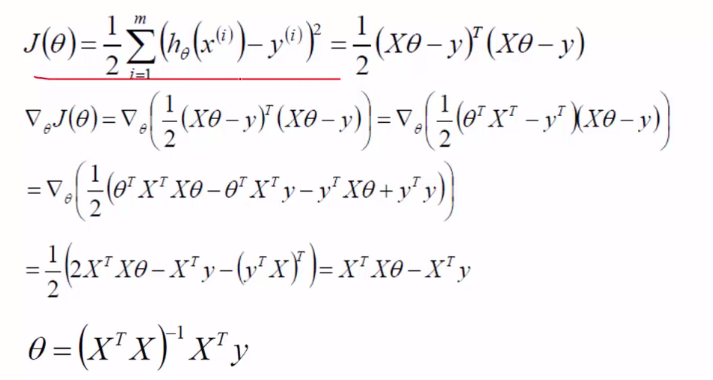
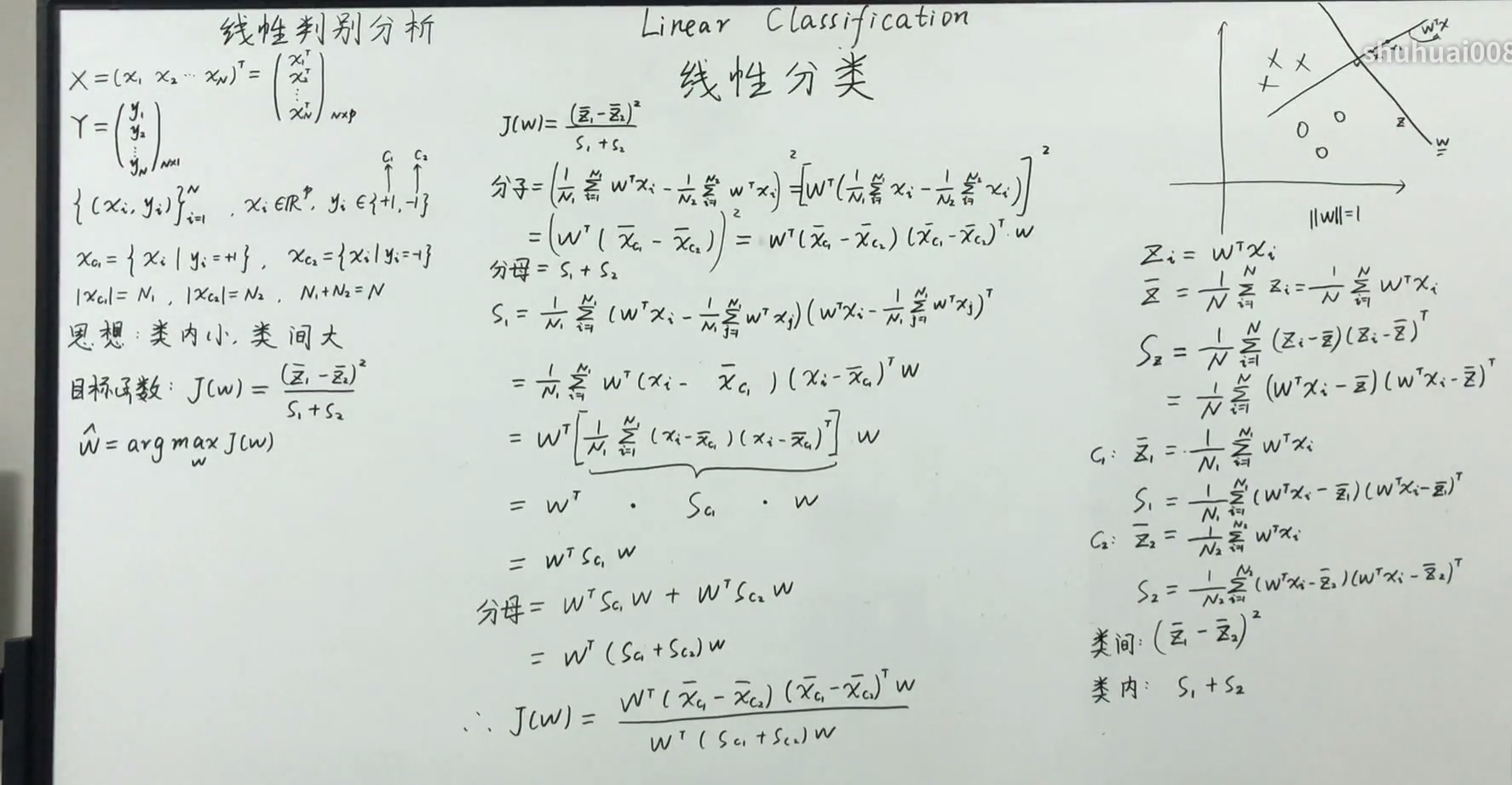
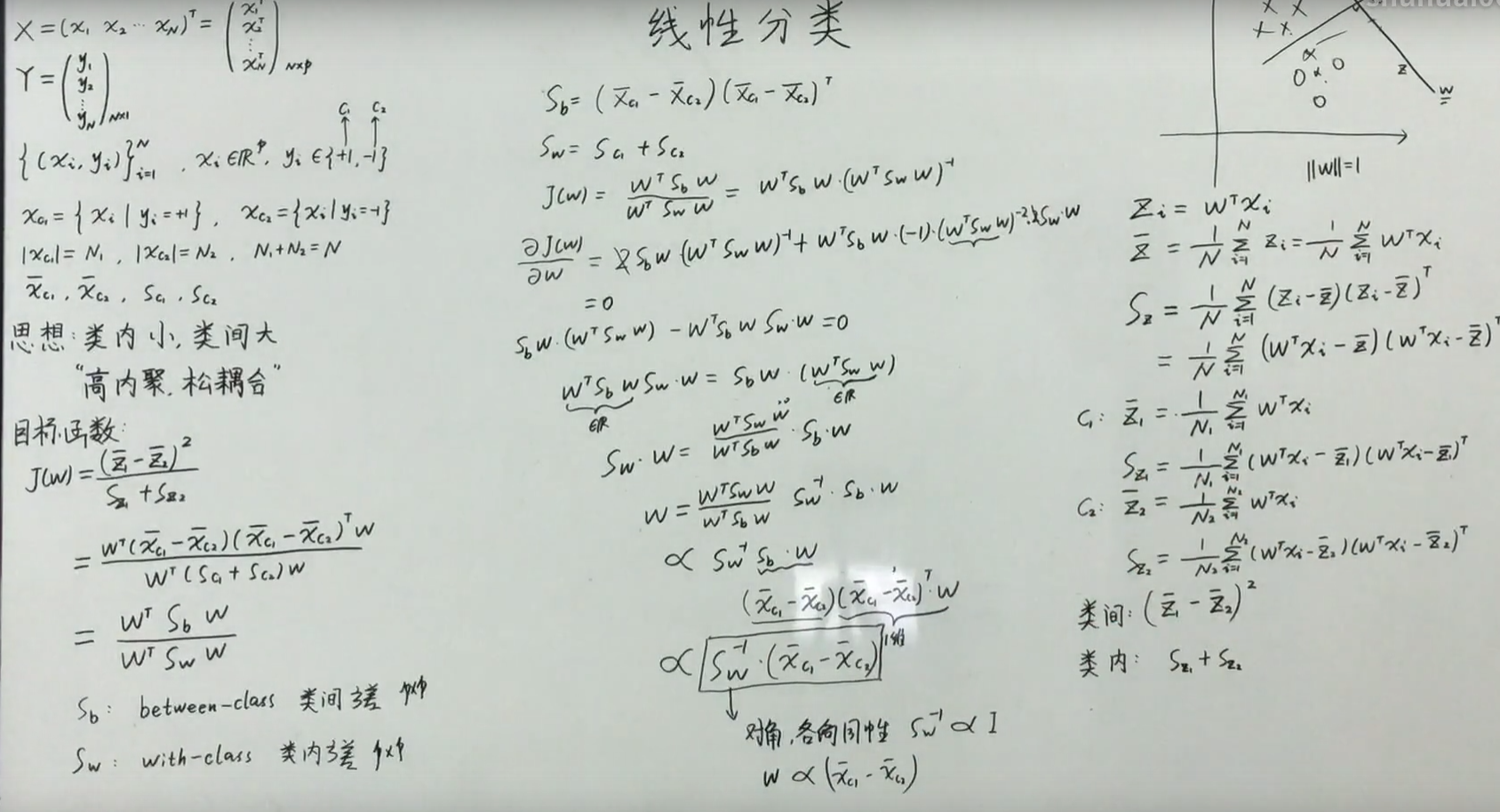
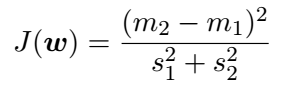
线性回归

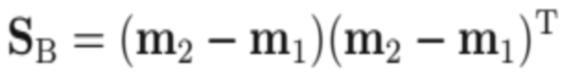


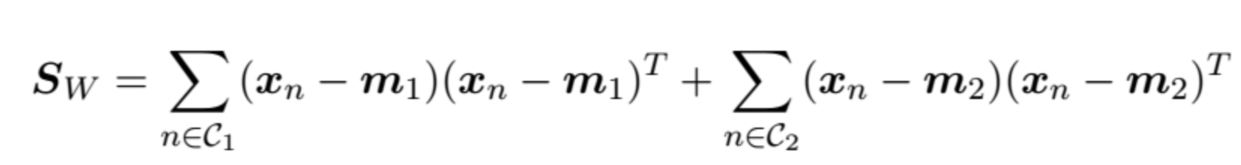
**Fisher线性判别**





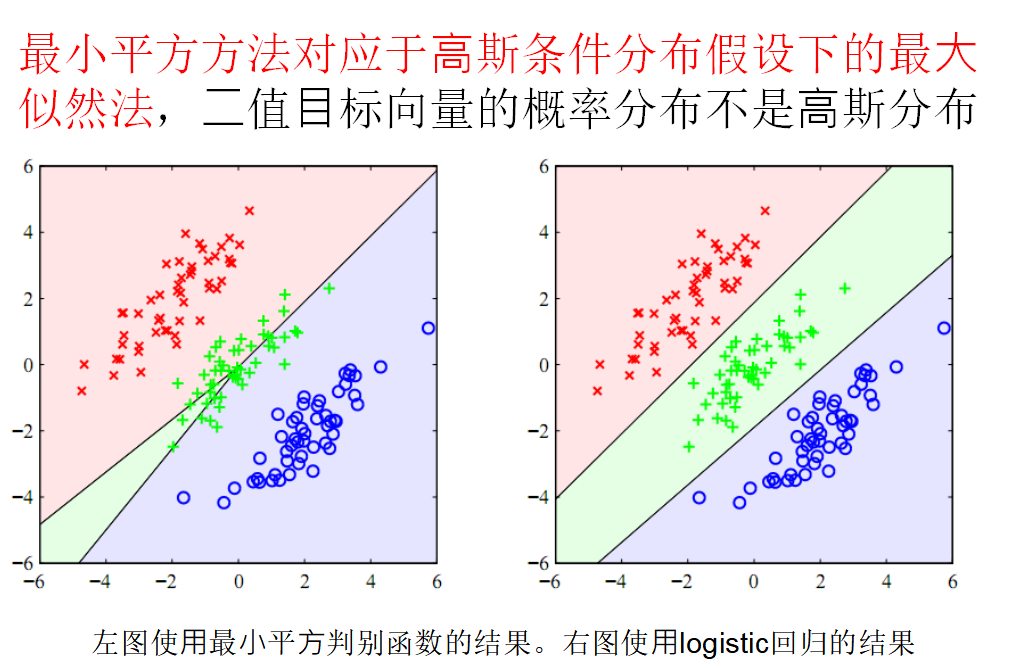
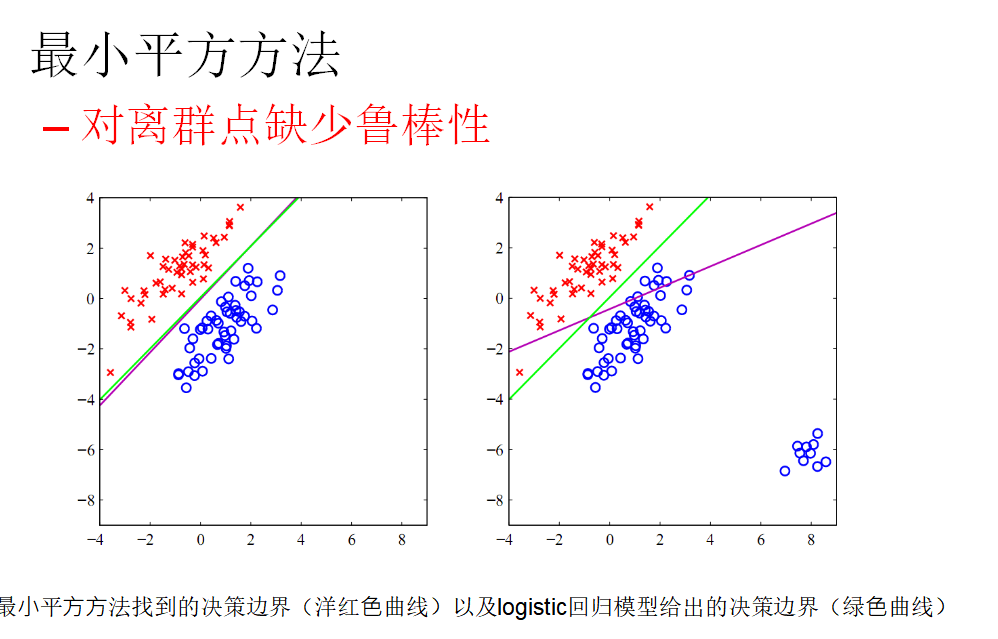
损失函数



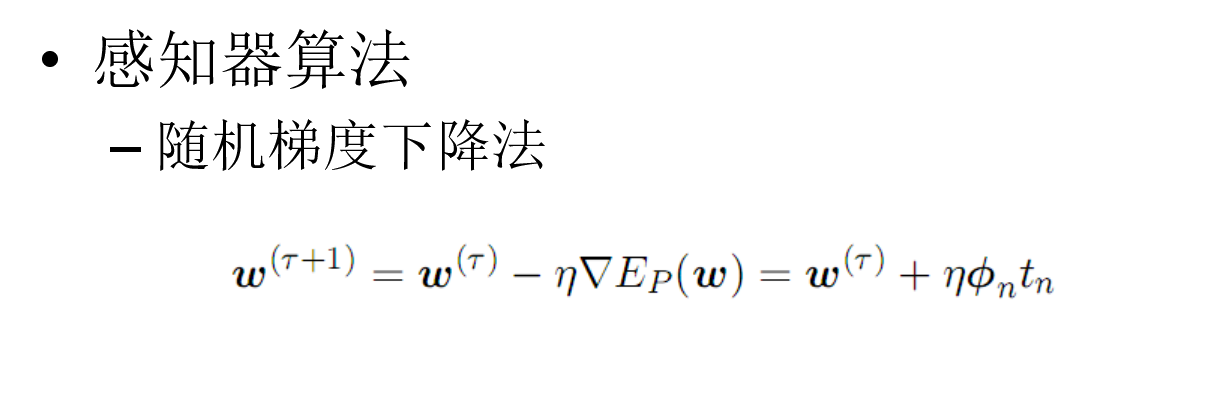
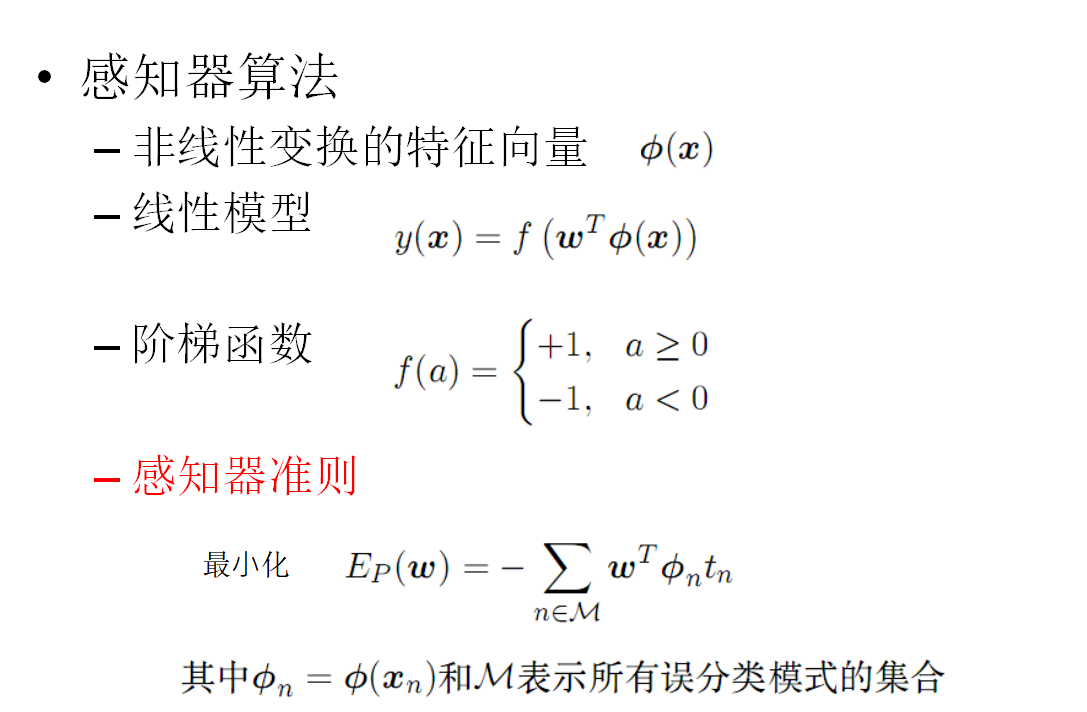


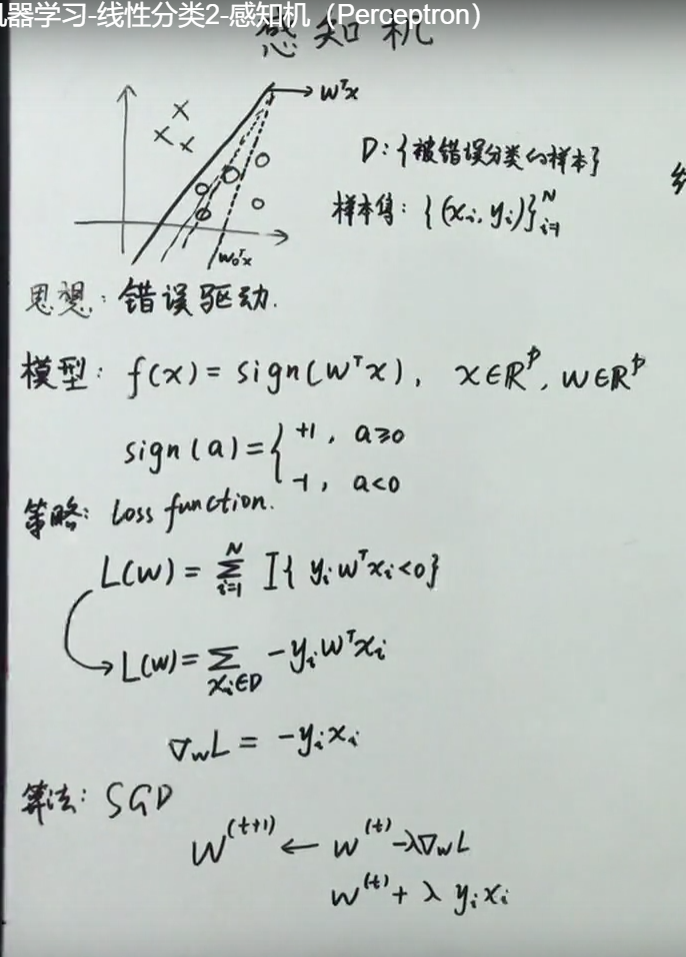


**最小平方方法**



**感知器**





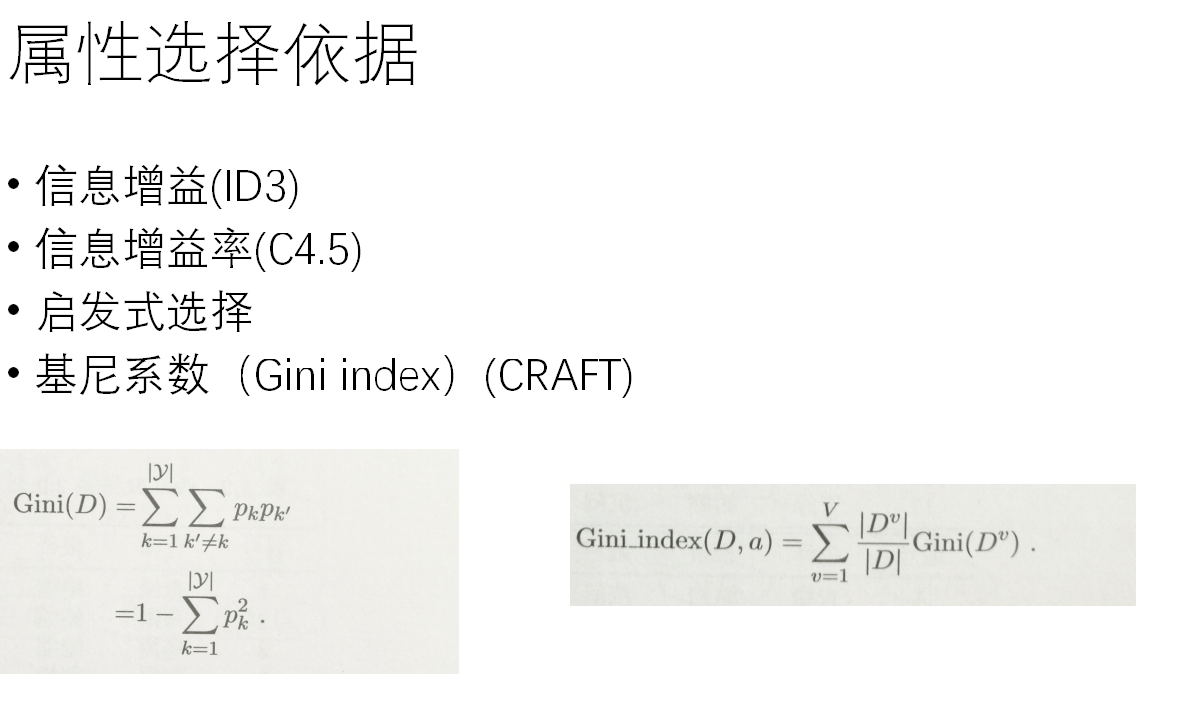
**决策树**

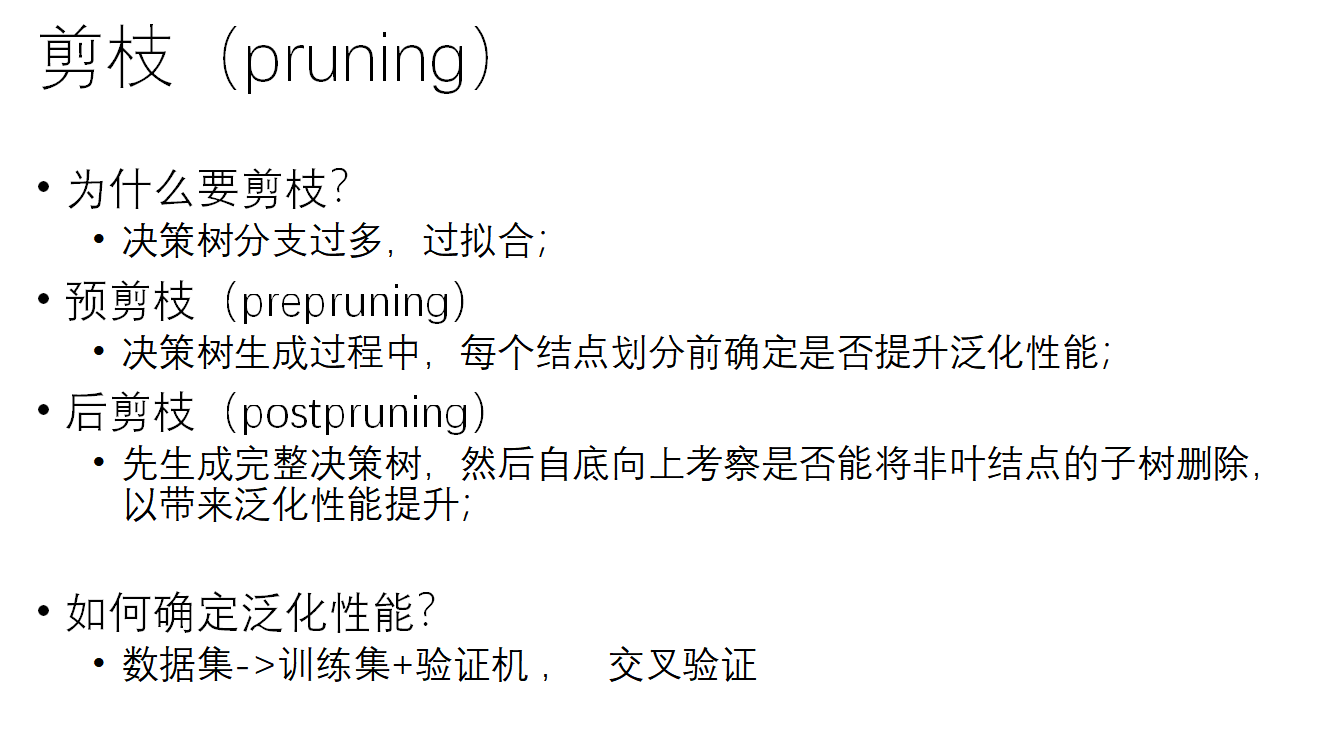
先求D的信息熵 Ent(D)

再求每个属性的信息增益Ent(ai) - Ent(D)

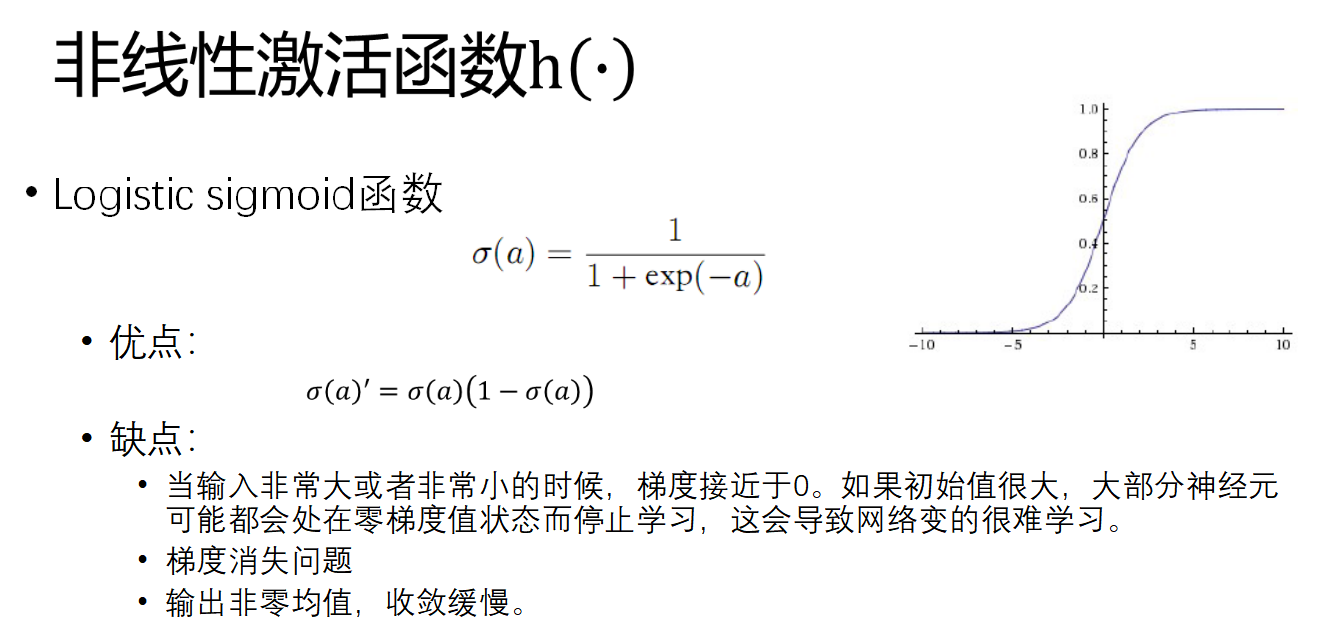
选择增益最大的属性作为根节点 然后分支

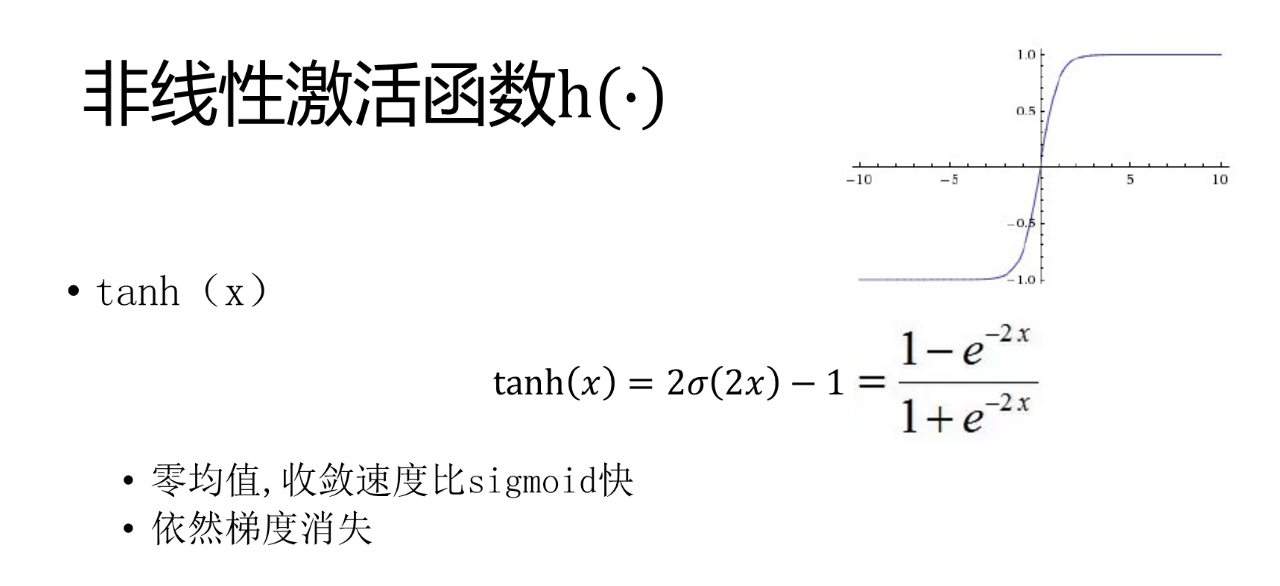
循环计算 得到决策树

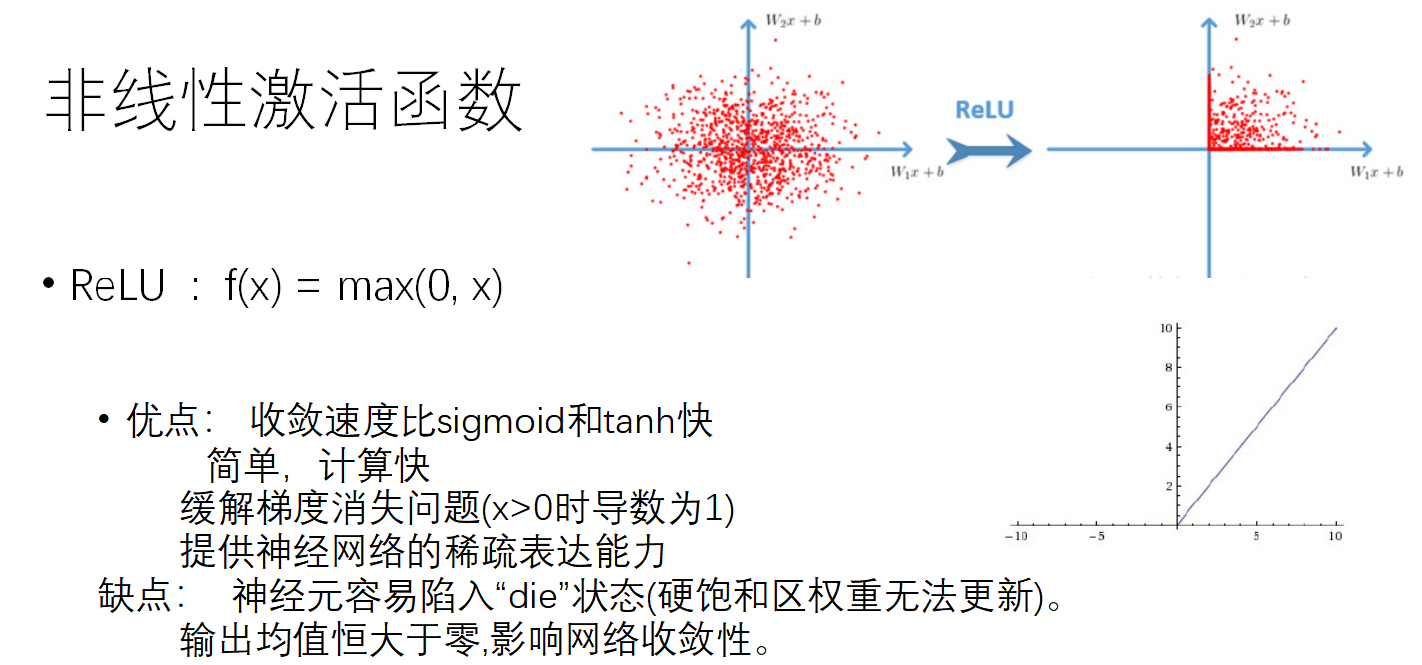


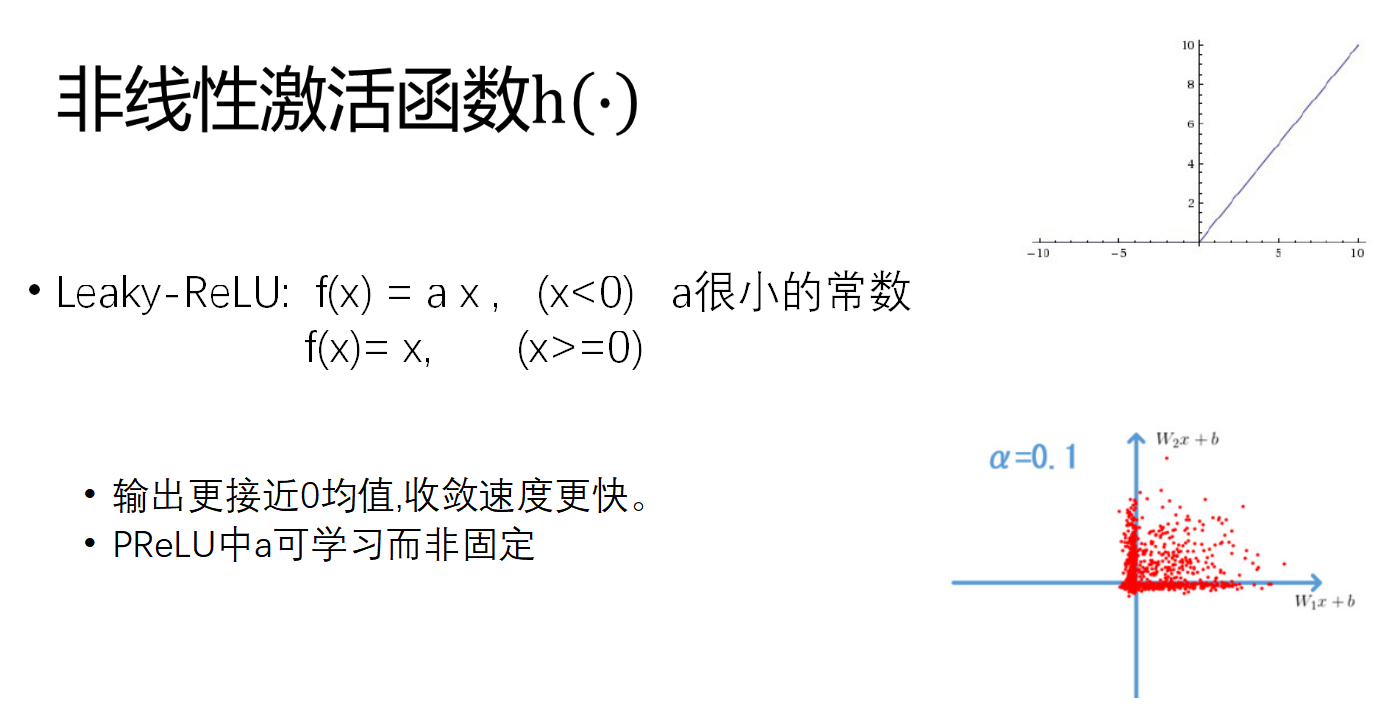


**非线性激活函数**

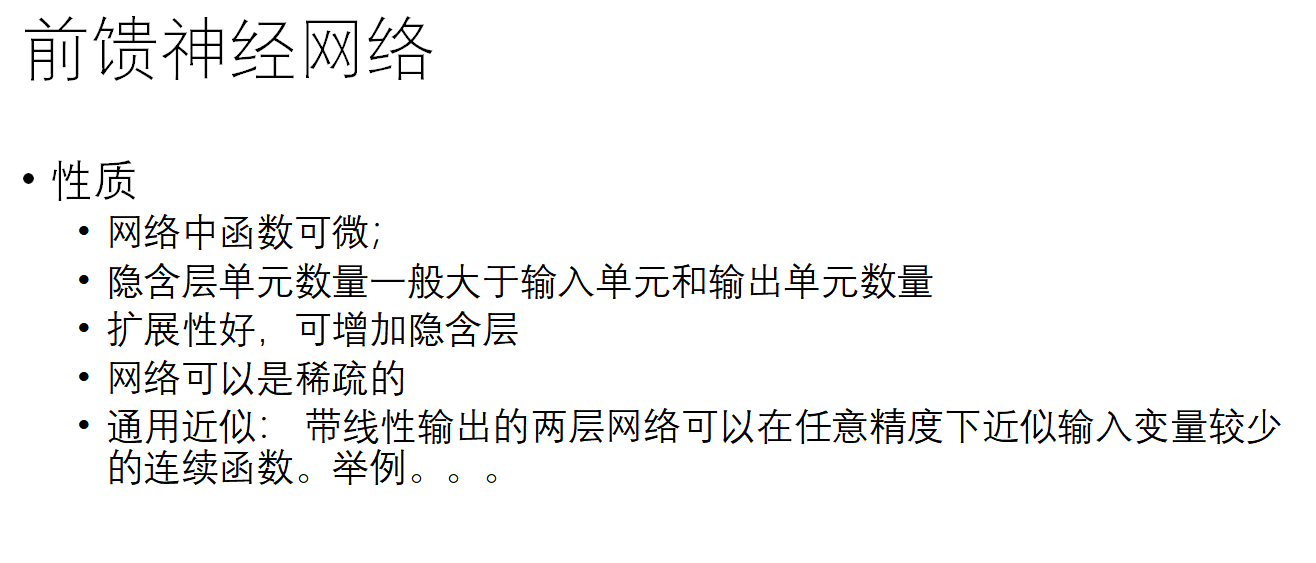


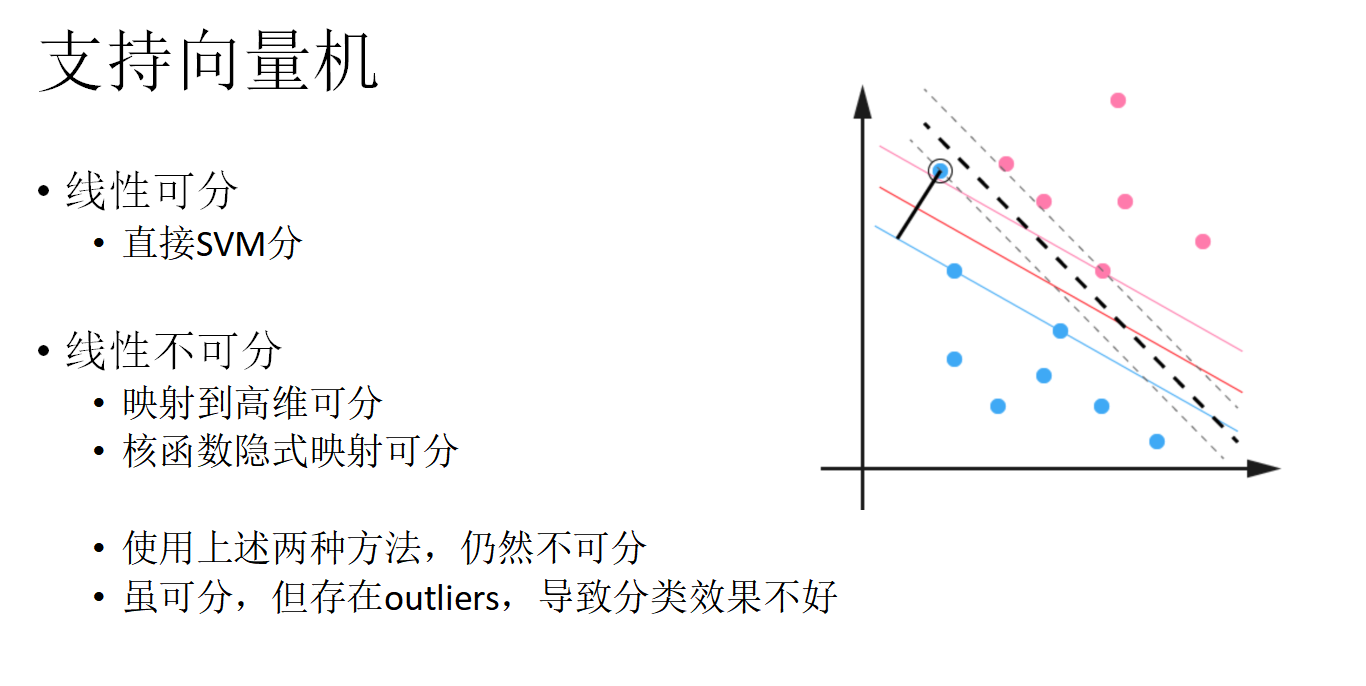












核函数本质

实际中，我们会经常遇到线性不可分的样例，常用做法是把样例特征映射到高维空间中去，相关特征便被分开了，也就达到了分类的目的；

但如果线性不可分的样例映射到高维空间的维度大小高到可怕的程度，就必须使用核函数；

核函数的价值在于它虽然也是将特征进行从低维到高维的转换，但核函数是在低维上进行计算，而将实质上的分类效果表现在了高维上，也就避免了直接在高维空间中的复杂计算。

