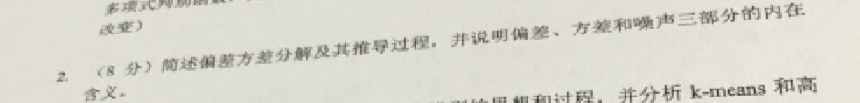


1. 线性判别函数一般是 其中是特征向量的增广形式， 是权重系数。根据的取值进行分类，这个函数在几何上一般表现为直线（高维空间的超平面） ,所以称之为线性判别函数。如果是低纬向高维投影后的特征向量，那么就是广义线性判别，理论上广义线性判别可以模拟任意复杂的函数。
2. 虽然广义线性判别可以拟合非线性，但是会面临参数爆炸问题，假如采用核技巧，虽然参数不会爆炸，但是面临kernel函数形式有限和无法度量哪个kernel函数更有效的问题，因此，假如经过先验知识，可以设计更合适的非线性的模型，是必要的。
3. 参数数目
   1. 线性需要（3+1）= 4
   2. 二次需要 ( 3(一次) + 3（二次） + 3(混合) + 1 （偏移） ) = 10
   3. 或者直接用：公式
      1. 线性
      2. 二次



1. 偏差-方差分解的推导过程

由于,因此第三项为0

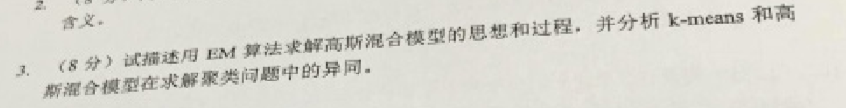
由于,因此第四项为0

是方差

是偏差

是误差

1. 内在含义
   1. 偏差：偏差是训练所使用的模型和模式之间的差异导致的错误
   2. 方差：方差是相同模型在不同采样数据下训练带来的误差。
   3. 噪音：是采样过程中采样误差（噪声）导致的训练结果的错误。



1. EM 算法求解高斯混合模型
   1. 思想： 假定存在M个独立的高斯分布，数据按照 的概率从第个高斯进行采样获取的。
   2. 过程
      1. 首先初始化混合高斯的参数
      2. 迭代直到收敛
         1. E步骤，计算
         2. M步骤，更新

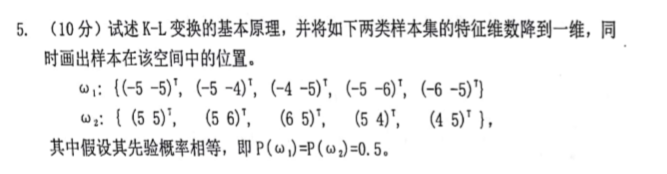
参数迭代公式（背过吧~）

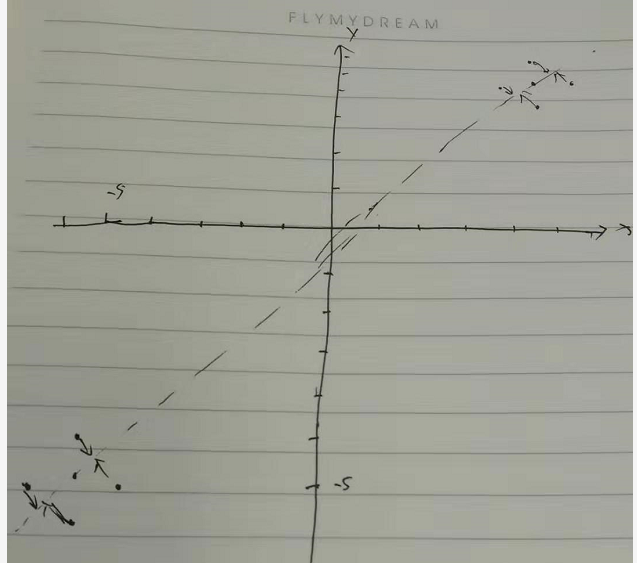
1. k-means 和高斯混合的异同
   1. 不同：
      1. k-means 的损失函数是最小平方距离， 混合高斯是负对数似然函数
      2. k-means 是硬划分，混合高斯是软化分
      3. k-maans 假设类别是概率相同且是球簇，混合高斯可以处理非球形，类别概率不同
   2. 相同：
      1. 混合高斯的E步骤其实就是软化分的k-means
      2. 当类别概率相同，且 的时候，混合高斯退化为k-means。

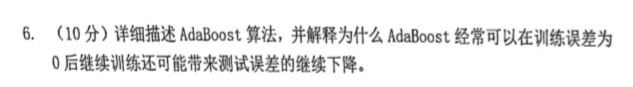


迭代直到全部可以分类:

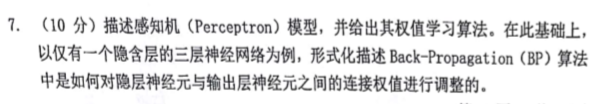
刚好四个点都需要添加进去。



1. K-L 变换的基本原理：
   1. K-L的关注问题杀是在均方误差最小的条件下获得最佳降维变换。
   2. 算法步骤是：
      1. 将特征减去均值
      2. 计算协方差矩阵
      3. 进行特征值分解，获得的特征向量按照特征值大小排序，取其前个作为转移矩阵
      4. 就是降维后的特征
2. 对样本进行降维
   1. 符合最佳K-L 变换需求
      1. 特征向量
   2. 降维：
      1. {
   3. 



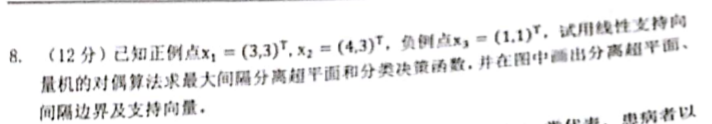
1. AdaBoost算法
   1. 预设样本为 第m个弱分类器为 ，分类器性能为 ，分类器权重为 , 第m轮样本权重为
   2. 初始化样本权重
   3. 迭代
   4. 根据当前权重 训练弱分类器 ，要求性能优于随即猜想
   5. 计算 ,
   6. 更新 其中是归一化因子
   7. 最终的分类器是
2. 继续训练类似于增大margin , 虽然训练正确率100% , 但是泛化能力增加。



1. 感知器模型：
   1. 描述： 感知器模型是一种赏罚模型，分类正确就不处罚，分类错误就处罚，直到全部样本都分类正确为止。
   2. 公式解释： 期望对于所有 有 ， 因此其损失函数为 其中 是全部错分 样本。 因此其权重更新公式为 , 其中是超参数“步长”。
2. BP 算法需要误差 的反向传播。 预先定义： 每一层的输入向量分别是 ，真实标签是 , 每一层的输出是 , 转移函数是 ， 权重是 和
   1. 在输出层，误差 , 对应的输入的误差是
      1. 隐含层🡪 输出层的连接权值更新:

( 如果题干要输入🡪 隐含层的权重更新)

* 1. 在隐含层，输出的误差是输出层输入的加权求和
     1. 对应的输入的误差是
     2. 对于输入层🡪 隐含层的权重更新



1. 对偶问题算法

显然

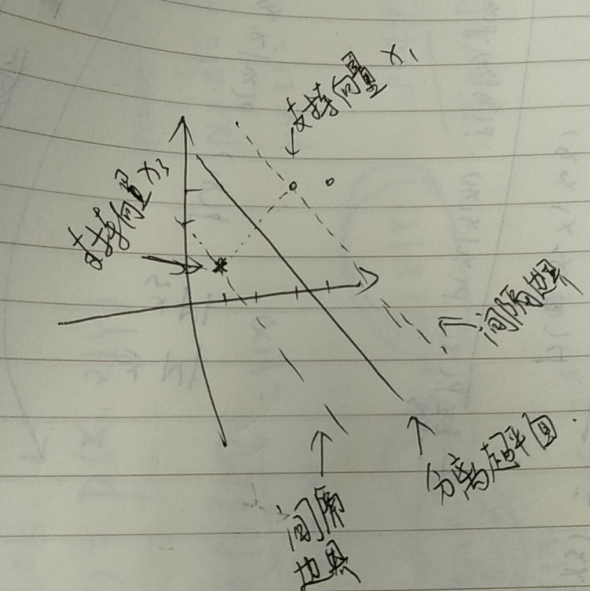
支持向量是 ,因此

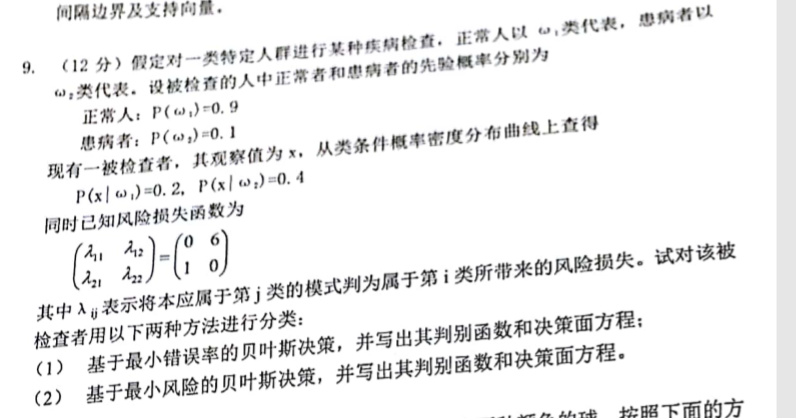
同理

无论从哪个点计算都是一样的，因此：

🡨 分类面。二维坐标系下 ：

1. 草图





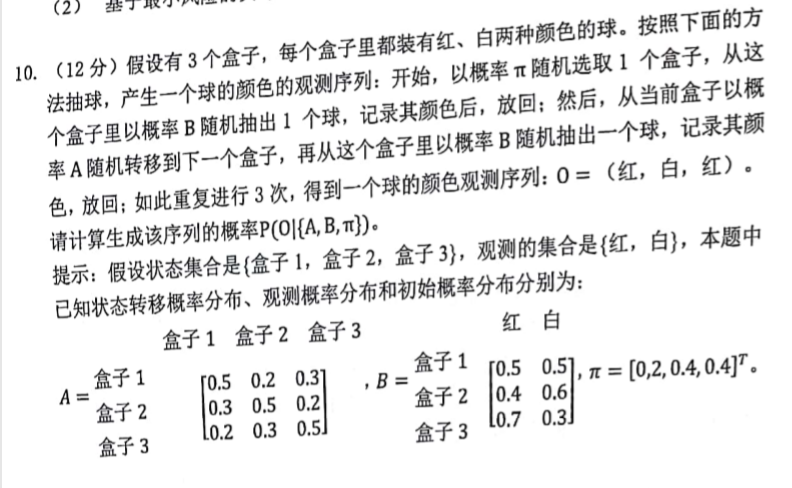
1. 最小错误率贝叶斯：
   1. 判别函数 ：
   2. 决策面方程
   3. 决策

判决属于。

1. 最小风险贝叶斯：
   1. 判别函数

* 1. 决策面方程
  2. 决策：

故判决属于



1. 前向计算需要的公式
2. 计算

-------------------------------------------------------

------------------------------------------------------------

因为没有仔细检查，上述计算可能存在错误！关键是知道前向计算的迭代公式！