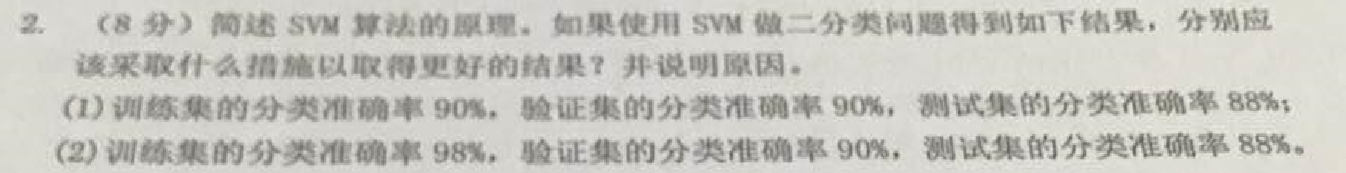


1. 试阐述线性判别函数的基本概念，并说明既然有线性判别函数，为什么还需要非线性判别函数？
   1. 线性判别函数的一般函数形式是,其中是特征的增广向量，则是权重系数，一般根据的取值来进行类别判定，比如2类问题可以定。因为这个函数的几何形态往往是一条直线（或者多维下的超平面），所以称为线性判别。如果是经过低维向高维投影的特征，则是广义线性判别函数。
   2. 虽然广义线性判别函数可以达到非线性判别的效果，但是随着模型复杂度的提升，往往会遇到参数爆炸的问题，采用核技巧虽然可以避开参数爆炸，但是也会遇到kernel形式有限和没有kernel是否合适的评估机制的弊端。因此如果能够基于先验知识确定一个合适的非线性判别函数，还是会避开很多问题而取得较好效果的。
   3. 包括6个4维不同的模式（样本？） ，则线性权向量至少多少？二次权向量至少要多少？
      1. 线性权向量至少要5个（d+1）
      2. 二次权向量至少15个 (
      3. 公式
         1. 线性
         2. 二次



1. SVM的算法的原理
   1. 一言以蔽之：最大化分类margin 。 在soft margin的情况下，其实是求解下面问题的最优解：

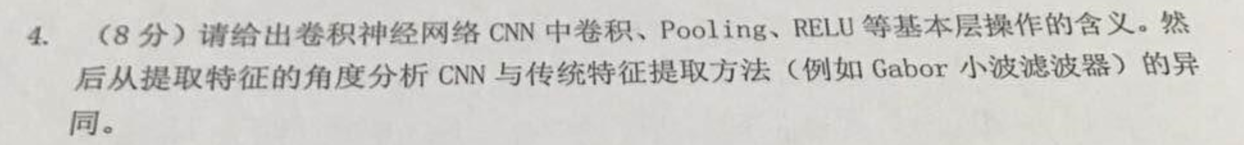
使用Lagranger 函数处理再取其对偶问题是：（得到的的就是支持向量，分类面在支持向量正中间。

1. 训练集合，测试集合，验证集合的准确性都大约是90% ，可以适当的增大小C值再训练，因为此时的模型尚未出现过拟合，同时准确率没有非常高，说明模型对错误的容忍过于宽泛，margin宽余实际需求。
2. 训练集98% ，测试和验证集合都越90% ，可以适当的减小C值，因为感觉已经过拟合，训练集合对错误过于严苛，margin太小导致泛化能力差。

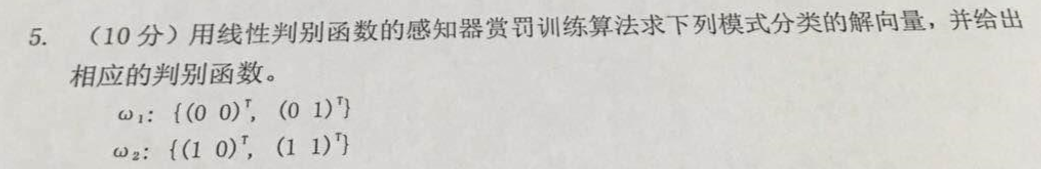


预设是变换矩阵，是原始特征向量，是变化后的向量。不失一般性的假定样本中心是坐标原点。

1. 最大化映射后的样本方差角度
2. 最小重建误差角度
   1. 求最佳
      1. 这个公式可以推导成最大方差公式

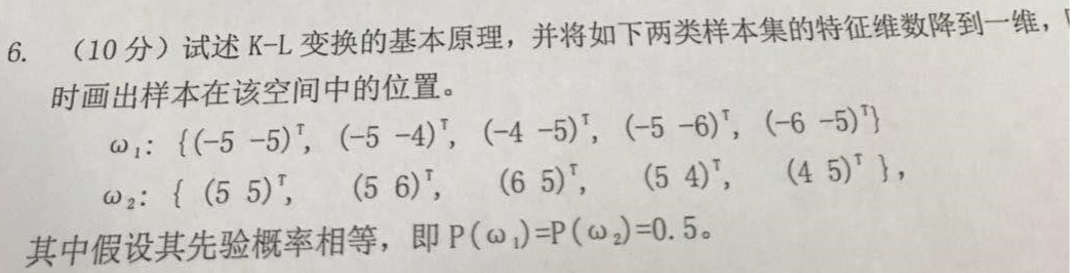


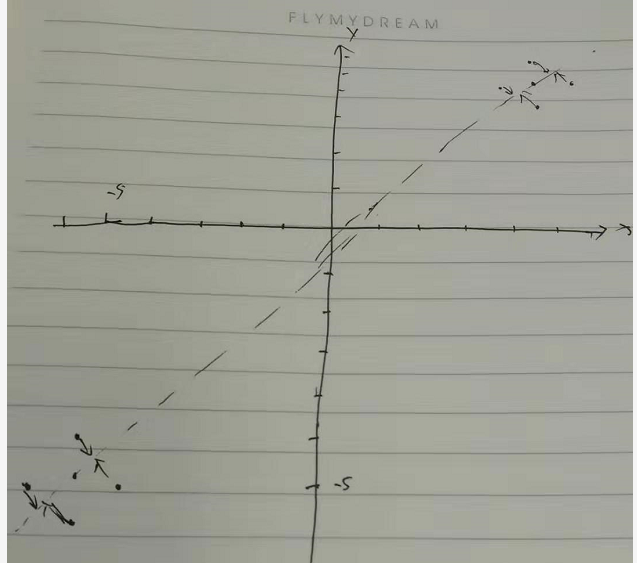
1. 基本层操作
   1. 卷积 ： 部分特征与滤波器做矩阵乘（相乘后求和为卷积）的操作，是一种局部特征提取的手段。
   2. Pooling ： 池化，将局部特征压缩（比如2x2 🡪 1）的手段。 池化是逐步扩大卷积的范围有效手段，从而使得在计算量不显著上升的情况下得到卷积也能获得更加全局的特征。
   3. ReLU ： 神经元非线性转移的一种。 。 是一种可以解决梯度消失的转移函数。不过会带来神经元死亡的问题。
2. 异同
   1. 相同： 不同的特征之间的权值共享
   2. 不同： CNN 的权值是学习获得的，Gabor的权重是预设的

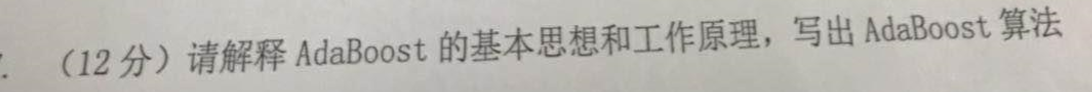


1. 使用批处理感知器
   1. 获得规范增广矩阵

* 1. 初始化向量 , 步长1
  2. 迭代
     1. ;
  3. 判别函数

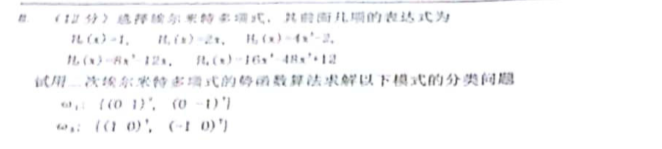


1. K-L 变换的基本原理：
   1. K-L的关注问题杀是在均方误差最小的条件下获得最佳降维变换。
   2. 算法步骤是：
      1. 将特征减去均值
      2. 计算协方差矩阵
      3. 进行特征值分解，获得的特征向量按照特征值大小排序，取其前个作为转移矩阵
      4. 就是降维后的特征
2. 对样本进行降维
   1. 符合最佳K-L 变换需求
      1. 特征向量
   2. 降维：
      1. {
   3. 



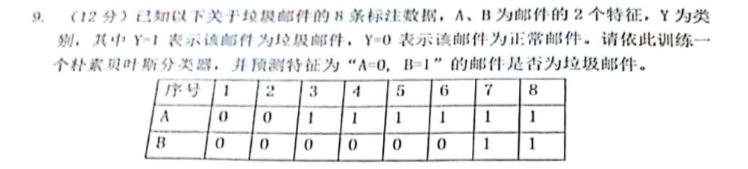
1. 基本思想：
   1. 构造强学习往往难度较大，构造弱学习器则不难，如果能构造多个弱学习器并使得他们能够互补的话，就能够组合出好性能。
   2. adaboost 采用在弱学习器1上失败的样本训练弱学习器2
      1. 确保弱学习器1在其训练集上误差<0.5
   3. 调整样本权重，使得弱学习器1在样本上表现等于随机猜想。
      1. 然后用调整过权重的样本来训练弱学习器2
2. Adaboost算法
   1. 给定训练集合： 其中 表示类别标签
   2. 初始化样本权重
   3. 迭代
      1. 对训练样本采用权重 训练弱分类器
      2. 计算当前权重下误差
      3. 更新权重 其中
         1. 是归一化因子
   4. 最终的强分类器是

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



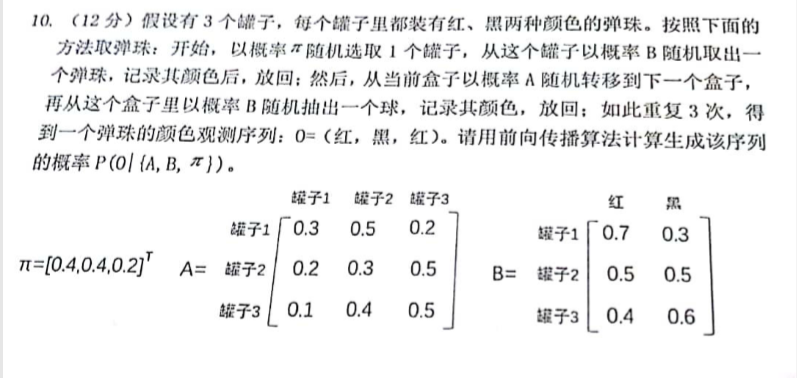
1. 构造正交函数集合，根据题干要求需要二次项，因此取和构建即可：
2. 构造核函数
3. 训练

迭代直到全部可以分类:



手动修正题干

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| B | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Y | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |



1. 参考公式
2. 计算过程

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

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

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

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

假设我们需要计算最佳状态序列

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

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

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

… 显然 最大

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

显然 最大

= 0.133492

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

可见 最大

最佳状态序列是 1🡪2🡪3 , 由于3个在千分位保持一致。