1. （必填）自己提出的问题的理解（罗列全部）：
2. **提出的问题1：**

P78提到的要怎么理解呢？混合概率的意义不太能理解，代表某个成分的参数，那么这里的参数包含什么呢？P79写道混合权重是类别先验概率，即在条件下是类别的概率，可否直观的理解为在所有数据中类别占总数的比例呢？

讨论后的理解：

包含了每一个成分的混合概率和每一个成分的分布参数，但是在实际计算过程中，我们并不用关心，只需要计算出成分（类）的先验概率即可，因为像成分的分布参数这些指标是体现在数据中的。成分的参数可以理解为分布参数，就像如果是分布参数，就是和。在计算先验概率时可以直接计算该类在所有数据中所占的比例。

1. **提出的问题2：**

P86公式61提到的i的取值范围是支持向量的下标，那么如何找到这些支持向量呢？是通过解出参数之后，令约束函数等于0的数据吗？

讨论后的理解：

根据Kuhn-Trucker条件，满足约束条件公式55和公式56的点即是支持向量。但是在实际计算过程中，我们是先找到支持向量并用其来求解未知数b。因此在找支持向量时应该先找满足条件>0的向量，这些向量就作为支持向量然后用于求解未知数b。

1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：
2. **问题3**：

p83式41下面说使用超平面上的任一点与边缘超平面的距离来作为d+，而不是x+到超平面的距离，为什么会这样计算？两种方式有什么本质上的区别？如果没有的话书上为什么会强调这一点？

自己的理解：

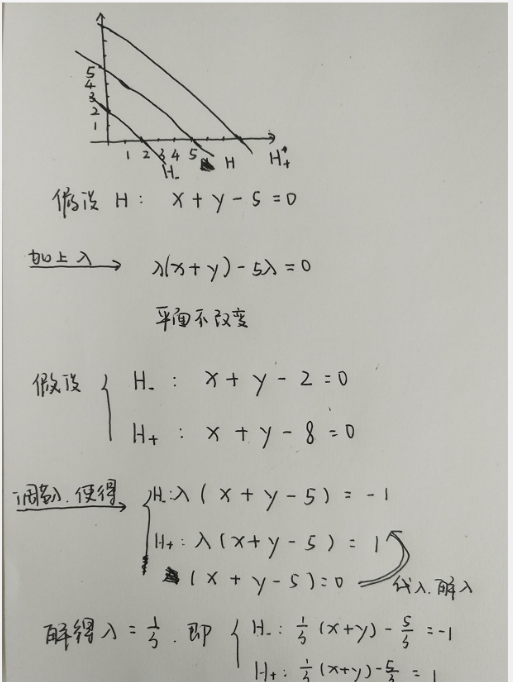
根据点到平面的距离公式，如果采用计算x+到超平面的距离方式计算，则根据公式40计算，其中由于x+在其中一个边缘超平面上，所以代入公式40之后，分子的值应该为-1或者1，能够得到与公式42相同的结果。但是这仅仅是针对x+刚好在边缘平面的情况，事实上，由于数据并非完全规范导致可能x+并非完全在边缘超平面，所以采用该方法就会产生误差。而采用计算面到面之间的距离，由于本身边缘超平面就已经是经过一次误差（平滑）处理了，就能减小误差。

1. **问题4：**

83的第三行，通过加入参数，我们可以调节超平面，并且不改变函数，如何理解，参数为何一定是正实数

自己的理解：

相当于整个等式左右两边同时乘以参数lambda ，超平面不改变。但是通过这个参数，我们可以找到公式38和公式39两个边缘平面。如果没有这个参数，两个边缘平面的式子极大可能不会写成=-1或=+1的情况，但是我们可以通过调节参数lambda，使得两个边缘平面的表达式成立。按照我的理解，我认为要求是正实数只是一种约束而已，因为我们想要得到在超平面的下方的边缘平面始终=-1，而超平面上方的平面始终=1，就需要保持系数的正负号不变，否则就需要交换边缘平面的-1和1。



1. **问题5：**

SVM的核方法把输入样本映射到高维的特征空间，在高维空间中还能保证算法的有效性吗？怎么评估这个误差是不是合理？？

自己的理解：

算法的有效性不是根据映射来的，而是根据映射之后通过对训练集的训练来的，即我们需要找到一个最好的映射方式（核函数）来将地位空间的数据映射到高位空间，使得训练数据线性可分。并不是说任意映射到高位空间就一定能达到可分的效果，因此需要根据不同的场景和数据特征选择不同核函数。因为数据是带有类标签的，所以我们可以直观的获取到分类结果以验证算法的有效性。采取Cross-Validation方法，即在进行核函数选取时，分别试用不同的核函数，归纳误差最小的核函数就是最好的核函数。如针对傅里叶核、RBF核，结合信号处理问题中的函数回归问题，通过仿真实验，对比分析了在相同数据条件下，采用傅里叶核的SVM要比采用RBF核的SVM误差小很多。

1. （必填）读书计划

1、本周完成的内容章节：第三章（3.6-3.9）

2、下周计划：第三章（3.9-3.10）

四、（选做）读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1、读书摘要及理解（选做）

摘要：本周完成了第三章监督学习的朴素贝叶斯分类、SVM和KNN算法的学习。

**贝叶斯算法**是一种典型的监督学习中的生成模型。在此书中，主要介绍了朴素贝叶斯的思想和文本分类模型，其基本思想找到一个具有最大后验概率的类，将这个类赋予该数据。基于此思想，要对文本分类，首先要考虑如何将文本内容以计算机可识别并统计的方式表达出来，书中采取的方法是将文档看成是由单词的多项式分布生成。然后给定类别计算一篇文档的生成概率，找到具有最大概率的类别作为文本的类别。为了解决0估计问题，加入了参数，在实际操作中，可以通过多次交叉验证选择一个最优参数。

**SVM算法**一般用于两类的分类问题，支持向量机选择最大化正例和负例之间的边距的那个超平面作为分界。对于数据现在线性可分的情况，用标准拉格朗其乘子来解决，由于其计算过程中由于求导会出现对偶问题，所以采用求解其相对容易的对偶问题的方法来求解原问题。而对于数据不可分情况，有两种解决方案。第一种是添加一个误差代价修改目标函数，这种方法在求解对偶问题时是原来线性可分的情况是一样的，区别在于增加了一条关于C(一个是先指定的参数)的约束；第二种是将原本的低维数据通过核函数映射到更高维的空间，让原本在低维空间线性不可分的数据在高维空间的线性可分。关于核函数的选择，可以采取Cross-Validation方法，尝试多个核函数，最终选择归纳误差最小的核函数。

（PS: KNN算法写在下周的读书报告中）

2、伪代码的具体实现(选做)

实现了KNN算法：

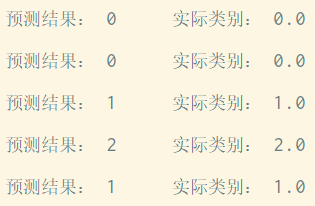
算法准备：



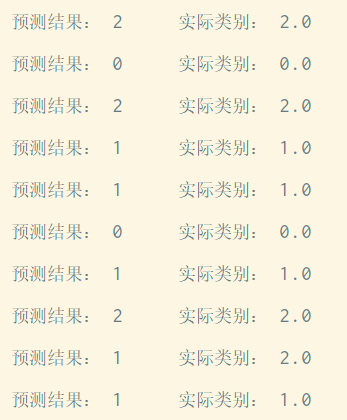
算法实现：



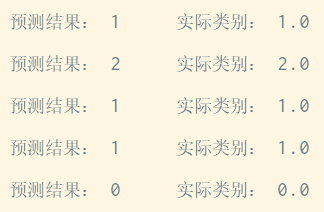
测试结果1：k=5，n=5，正确率100%



测试结果2：k=5，n=10，正确率90%



测试结果3：k=10，n=5，正确率100%



测试结果4：k=10，n=10，正确率100%

