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

Boosting算法中会将之前分类器错分的样本赋予更高的权重，会造成对异常样本敏感，降低了强学习器的准确性，那么除了去掉异常值有没有其他可能来解决这个问题呢？

讨论后的理解：

在迭代过程中，赋予样本权重的计算公式时可以考虑到异常值可能会带来的影响，即不过大化本轮分类结果与实际类别不相同的对应样本权重。但AdaBoost算法对异常值敏感是由算法本身的设计思想决定的，所以虽好的方法还是对数据进行预处理，去掉异常值和噪声数据再执行算法。

1. **提出的问题2：**

AdaBoost算法中弱学习器参数问题应该怎么考虑呢？

讨论后的理解：

考虑弱学习其的参数问题，算法最终的目的是完成分类，为了使得分类效果最好，需要使得损失函数最小，所以这是一个求解极值问题（前提是已经确定好数据权重分布），即找到一个系数最小化损失函数。

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**：

书中的两种集成分类器方法中每个基本分类器都是使用的相同的分类算法，如果使用不同的分类算法有没有可能实验结合每种算法各自的优点的可能？

自己的理解：

书上只是举出了弱分类器（基分类器）可以完全一样的情况，同理，我们也可以采用不同的基分类器集成成为一个强分类器，成为同质强分类器；而由不同分类算法分类器构成的继承分类器也成为异质强分类器，这样的分类器会结合各分类器的优点，但是数据集的权重设置和各个弱分类器的系数也需要考虑在内。在算法设计过程中，基分类器需要一定的可靠性，同时基分类器需要具有多样性，然而很多情况下可靠性和多样性难以兼得。

1. **问题4：**

选择k的时候采用验证集合、交叉验证的方法，每次重新划分训练集和测试机都需要重新计算一下所有点的距离，时间复杂度会很大，有没有什么其他选择k的方法？

自己的理解：

为了解决每一次都需要计算测试数据与样本点之间的距离造成时间复杂度为n的问题，可以用kd-tree（k-dimensional树的简称），是一种分割k维数据空间的数据结构来解决，其主要应用于多维空间关键数据的搜索（如：范围搜索和最近邻搜索），采用此数据结构可以让时间复杂度下降至logn。关于如何选择一个更合适的k，可以先检查样本的分布，通过样本的分布参数确定样本密度，选取一个合适的k的范围，再不断测试逐渐缩小得到一个最合适的k。

1. **问题5：**

adaboost生成的一系列基本分类器的权重是相同的吗？是否可以认为越后面生成的基本分类器越好，应该赋予更高的权重呢？

自己的理解：

不是相同的，分类器的权重是通过最小化损失函数求解得到的，只与本轮的误差率有关。所以其系数不是固定不变的，同时也不是越后面的基分类器就赋予更高的权重。某种程度上，可以认为经过越多次的迭代，越后面的强分类器会趋向更好，但并不是基分类器更好。

1. （必填）读书计划

1、本周完成的内容章节：第三章（3.9-3.10）

2、下周计划：第四章（3.1-3.5）

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KNN算法和AdaBoost算法的学习与实现。

**KNN算法**原理是当预测一个新的值x的时候，根据它距离最近的K个点是什么类别来判断x属于哪个类别。所以KNN 算法有两个很重要的参数，一个是距离的计算公式，另一个是K值的选取。在算法实现过程中和实际运用中主要也是考虑这两个问题，KNN 算法很简单，且结果也相对准确，可以和一些很复杂的算法相媲美。但KNN算法的缺点也很明显，即其并不能给出一个可以理解的模型，且在计算过程中每次都需要遍历样本集进行距离计算（此过程可以通过KD树优化）。

**AdaBoost算法**全称叫Adaptive Boosting自适应增强算法，其自适应在于：前一个基本分类器错分的样本的权值会增大，而正确分类的样本的权值会减小，并再次用来训练下一个基本分类器。算法的核心在于解决四个主要的问题：第一，如何计算学习误差率e，书中采用将分类错误的样本权重加和作为本轮的误差率；第二，如何得到弱学习器权重系数分类正确的样本，书中采用使得损失函数最小化的参数作为权重系数；第三，如何更新样本权重，书中采用分类正确的样本乘以上一轮的权重，分类错误的样本沿用上一轮的权重的方法；第四，使用何种结合策略，书中采用基分类器乘以分类器的系数加和作为强分类器，在实际操作中，为了防止过拟合，一般还会再加一个参数v用于调整权重。

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：

