Web data mining部分：

第三章讨论部分：

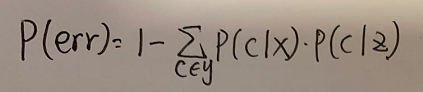
读书报告内容：

1. （必填）自己提出的问题的理解（罗列全部）：
2. 提出的问题1：

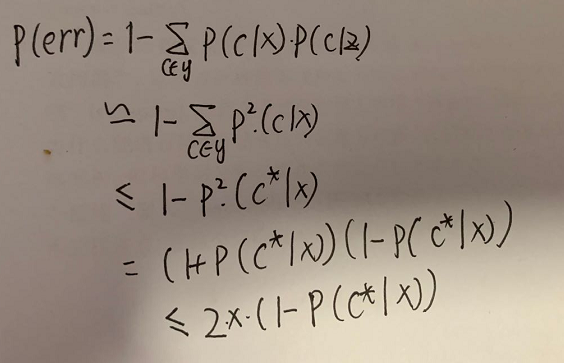
knn算法看起来精确度似乎得不到保证，但是书上却说精确度非常高，是否有数学证明可以证明knn算法精确度存在某个确定的下限？

讨论后的理解：

给定测试样本x，若其临近样本为z，则最近 分类器出错的概率就是x与z类别标记不同的概率，即为：



那么对任意测试样本，总能在任意近的范围内找到训练样本z，那么则有：

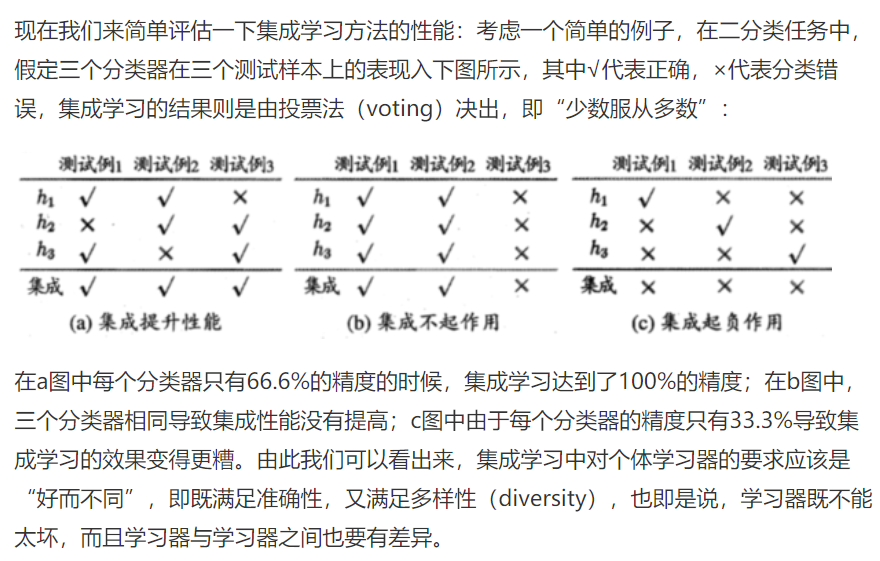


那么可得knn算法虽然简单，但是他的泛化错误率不超过贝叶斯最优分类器的两倍。

1. 提出的问题2：

集成学习选取多个学习器然后综合评价为什么可以获得更好的分类结果，换句话说，分类器中必定有好的有坏的，最终结果从生活经验来看不是应该趋向于最终的平均值嘛？

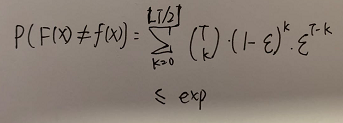
讨论后的理解：



所以这也就说明了为什么bagging算法要求算法不那么稳定。

我们假定基分类器的错误率为e，那么有。

假设集成通过简单的投票法结合T个基分类器，若有超过半数的基分类器正确，那么集成分类器就正确：

那么集成的错误率为。

上述显示出，随着集成中个体分类器的数T的增大。集成的错误率讲指数下降，最终趋向于0.

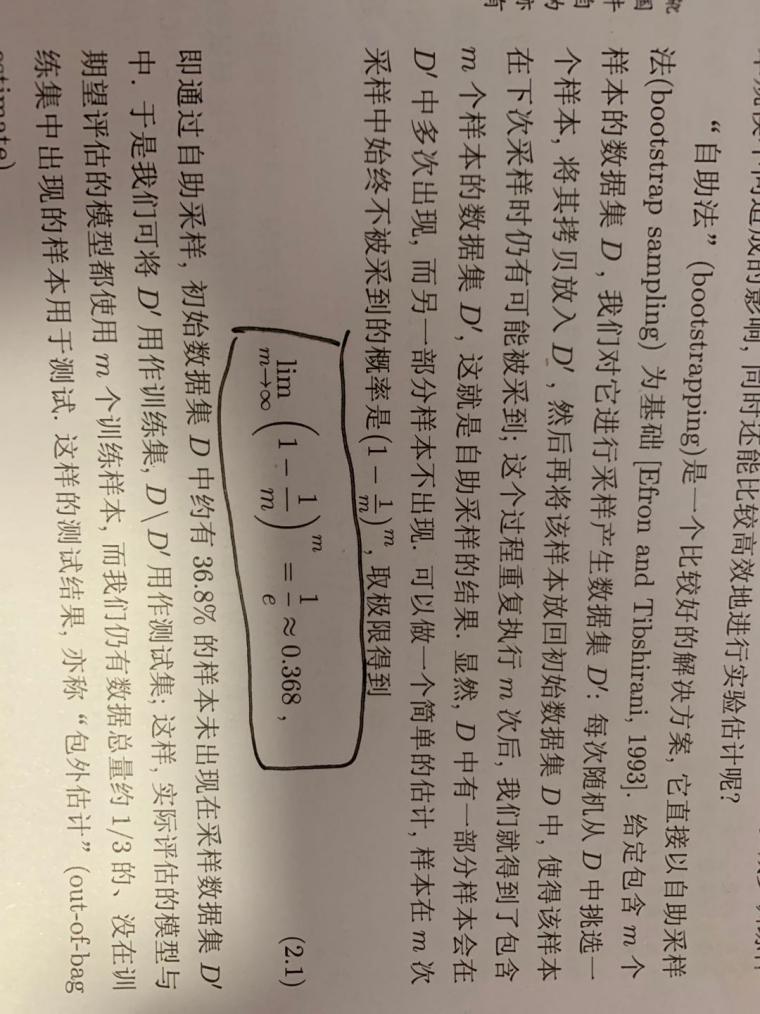
1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：
2. 问题3：knn的k个近邻其实还是有相对远近的，在决策时应该如何进一步优化？

自己的理解：

Knn算法其实是对样本的密度有要求的，比如说我们设置lamda的阈值为0.001，那么其实是要求一个单位空间的样本超过1000个，这在多维空间中的数据样本要求会达到一个天文数字，所以我们要先进行数据的降维，然后再根据相应的函数，比如说范式距离这些再进行距离计算，确定相对远近。

1. 问题4：为什么Bagging算法取一个自展复制的时候每个样本平均有63.2%的原始样本？

自己的理解：

这个其实源于bagging抽样的方法，他采用了自助采样法，由下式可得，初试训练集有63.2%的样本出现在采样集中。并且我们还可以通过采样形成的自然分类，用从来没用训练过的数据集作为验证集或者调参使用。

1. 问题6：knn在样本不平衡时会有缺陷，例如有一类样本容量很大，其他类样本容量小，导致大容量的样本在k个邻居中占多数。有无改进的办法？

自己的理解：

这个问题我和牛兄弟都提出了一些思路。

我的想法是，可以参考NLP里面的逆文档数，对于大容量的样本除以逆文档数，对大容量的数据进行加权投票，这样可以有效的减少大容量数据产生的误差。

牛作尧的想法是，在采样的时候改变权重，比如说大容量样本10个里面采样1个，小容量的样本十个里面采样4个，这样都可以有效减少大容量数据产生的误差。

1. （必填）读书计划
2. 本周完成的内容章节：

（1）3.9到3.10完成并且梳理知识点，寻找问题，自己思考并且于小组会之前完成了自己的思考。

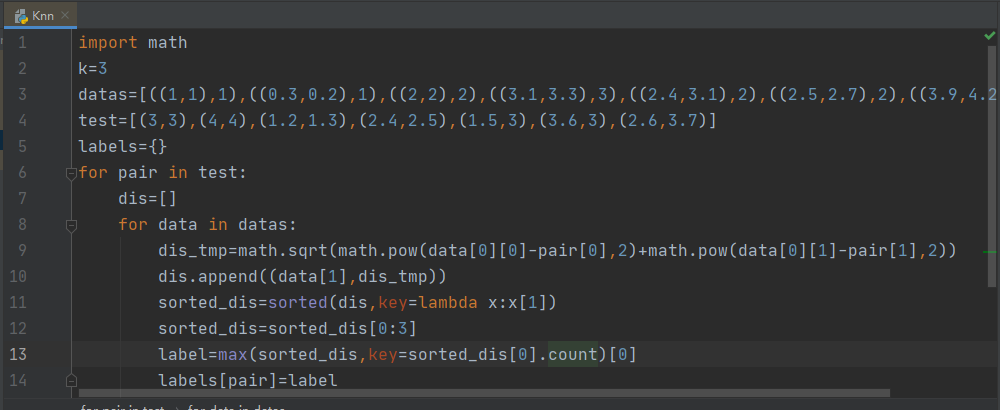
（2）为配合第三章的阅读，同时参考并且阅读了西瓜书的第8章，第10章，参考了视频《白板推导》和视频《一起啃书》，并且放到一起综合学习，adaboost算法可以参考知乎《手把手教你adaboost》和李航《统计学习方法》的第八章，k近邻算法可以参考李航《统计学习方法》的第三章

2、下周计划：2.1-2.5的阅读.

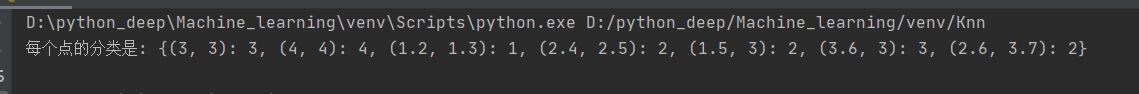
四、（选做）读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1.代码实现

Knn算法的代码实现：



输出结果：



1、读书摘要及理解（选做）。

（1）阅读了李航统计学习的第三章，knn算法最简单的实现是线性扫描，但是这个训练集合很大的时候非常消耗时间，所以为了提高knn算法的效率，可以考虑特殊的存储结构训练数据 ，比如kd树方法。

Kd树是一种便于对k维空间中的数据进行快速检索的数据结构。Kd树是二叉树，表示对k维空间的一个划分，其每个节点对于k维空间划分中的一个超矩形区域。利用kd树可以省去对大部分数据点的搜索，从而减少搜索的计算量。

（2）adaboost算法的特点是通过迭代每次学习的一个基分类器。每次迭代中，提高那些被前一轮分类器错误分类数据的权重，而降低那些被正确分类的数据的权重，最后组成一个强分类器，这样在前面投票的时候高权重的对于那些低权重，即使分类错误也无关紧要。

知识图谱部分：

## DBpedia - A Crystallization Point for the Web of Data

DBpedia 是一个很特殊的语义网应用范例，它从维基百科(Wikipedia)的词条里撷取出结构化的资料，以强化维基百科的搜寻功能，并将其他资料集连结至维基百科。

通读了论文感觉主要是介绍DBpedia，涉及到type inference部分的其实不多，用到了type inference的其实主要是用的infobox的template做的mapping，感觉和之前看的本科生的毕业论文的基于规则的type inference的部分是比较像的。这一部分内容主要是在：这一部分。

Infobox因为不同的编辑者编辑属性键值对有不同偏好，所以我们用两种方法来改善这个问题，A generic approach which aims at

wide coverage and a mapping-based approach which aims at high data quality

第一种：主要缺点是没有解决同义词问题和由于Wikipedia属性没有显式定义数据类型，另一个问题是用于确定属性值数据类型的启发式算法的相对较高的错误率。

第二种：

主要是为了克服上面的缺点，但是是手工完成的，现在这种数据，显然是不实际的，果然是十几年前的东西了。。。

并且cover的datatye和数据量也是较少的。

# Knowledge graph construction from multiple online encyclopedias

这篇文章提出了一个新的框架来解决多个在线非英文的百科全书的知识图构建。 核心模块是知识提取和知识链接。 知识提取包括规则提取，即周期性地提取整个在线百科全书中的目标文章内容，以及live提取，只提取新实体和更新实体的内容。 知识链接利用启发式轻量级实体匹配策略和半监督学习方法来寻找重复的实体和属性不同的在线百科全书。

Section5介绍了知识链接的两种方法一个是轻量级实体匹配的三种策略。还有一个是 半监督实体匹配，以轻量级实体匹配结果为种子，寻找更多的等价实体。

下面的还没继续仔细看

下周计划：

1. 阅读sdtype论文系列的2013年版本和2017年版本。
2. 继续阅读上周没看完的Knowledge graph construction from multiple online encyclopedias
3. 跟进实现sdtype代码。