**机器学习及应用实验报告**

**——自实现朴素贝叶斯分类器**

实验编号 : 三

实验名称 : 自实现朴素贝叶斯分类器

实验室 : 电光学院105

姓名 : 刘嘉洋

学号 : 1412620

班级 : 软件学院14级二班

实验日期 : 2017年3月14日

至 : 2017年3月21日

评分教师 实验报告成绩

评分日期 年 月 日

**一、实验目的**

* 深入掌握朴素贝叶斯分类器实现原理
* 对Weka朴素贝叶斯源码进行阅读剖析
* 理解拉普拉斯平滑及实现逻辑
* 完成朴素贝叶斯分类器的编程操作
* 对比自实现分类器与Weka的表现结果

**二、实验环境**

* macOS Sierra Version 10.12
* weka-3-8-1
* Eclipse Mars Release 4.5.0

**三、实验报告**

1. 朴素贝叶斯分类器原理

1.1. 分类基础—贝叶斯定理

其针对性地解决如下问题：已知某条件概率，如何得到两个事件交换后的概率，也就是在已知P(A|B)的情况下如何求得P(B|A)。

条件概率：P(A|B)表示事件B已经发生的前提下事件A发生的概率，其基本求解公式为：

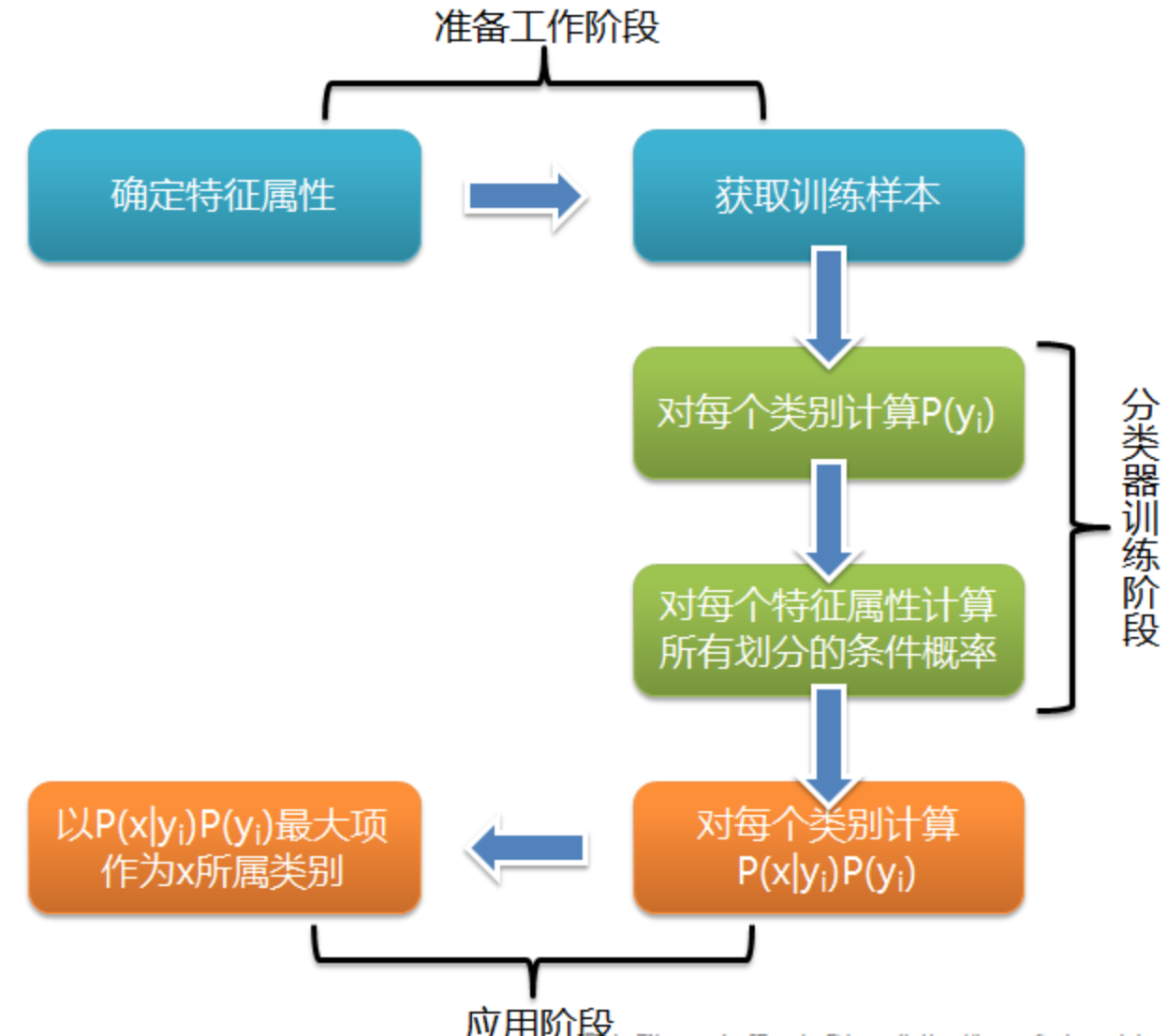
贝叶斯定理之所以有用，是因为我们在生活中经常遇到这种情况：我们可以很容易直接得出P(A|B)，P(B|A)则很难直接得出，但我们更关心P(B|A)，贝叶斯定理就为我们打通从P(A|B)获得P(B|A)的道路。

不加证明地直接给出贝叶斯定理：

极大后验(MAP)假设：用贝叶斯公式计算每个候选假设的后验概率，寻找给定数据D时可能性最大的假设h：

1.2 朴素贝叶斯分类

朴素贝叶斯分类是一种十分简单的分类算法，叫它朴素贝叶斯分类是因为这种方法的思想真的很朴素。朴素贝叶斯的思想基础是这样的：对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。



可以看到，整个朴素贝叶斯分类分为三个阶段：

第一阶段—准备工作阶段，这个阶段的任务是为朴素贝叶斯分类做必要的准备，主要工作是根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征属性和训练样本。这一阶段是整个朴素贝叶斯分类中唯一需要人工完成的阶段，其质量对整个过程将有重要影响，分类器的质量很大程度上由特征属性、特征属性划分及训练样本质量决定。

第二阶段—分类器训练阶段，这个阶段的任务就是生成分类器，主要工作是计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。其输入是特征属性和训练样本，输出是分类器。这一阶段是机械性阶段，根据前面讨论的公式可以由程序自动计算完成。

第三阶段—应用阶段。这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是机械性阶段，由程序完成。

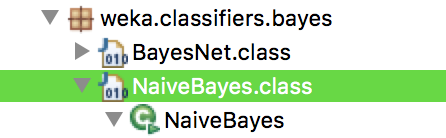
2. 拉普拉斯平滑原理

一个值得讨论的问题就是当P(a|y)=0怎么办，当某个类别下某个特征项划分没有出现时，就是产生这种现象，这会令分类器质量大大降低。

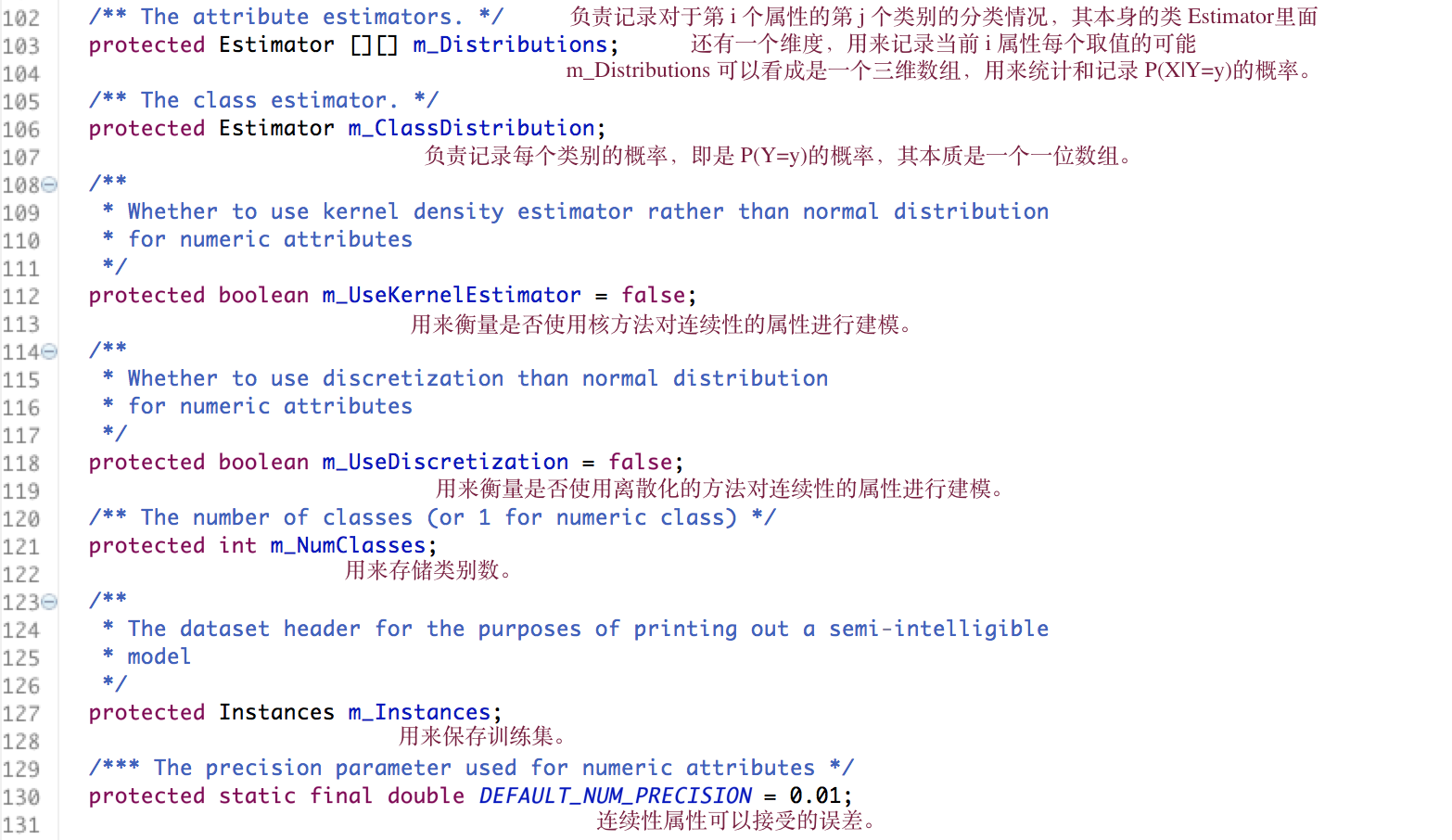
为了解决这个问题，我们引入Laplace平滑校准。它的思想非常简单，就是对没类别下所有划分的计数加1，这样如果训练样本集数量充分大时，并不会对结果产生影响，并且解决了上述频率为0的尴尬局面。

3. Weka朴素贝叶斯源码剖析

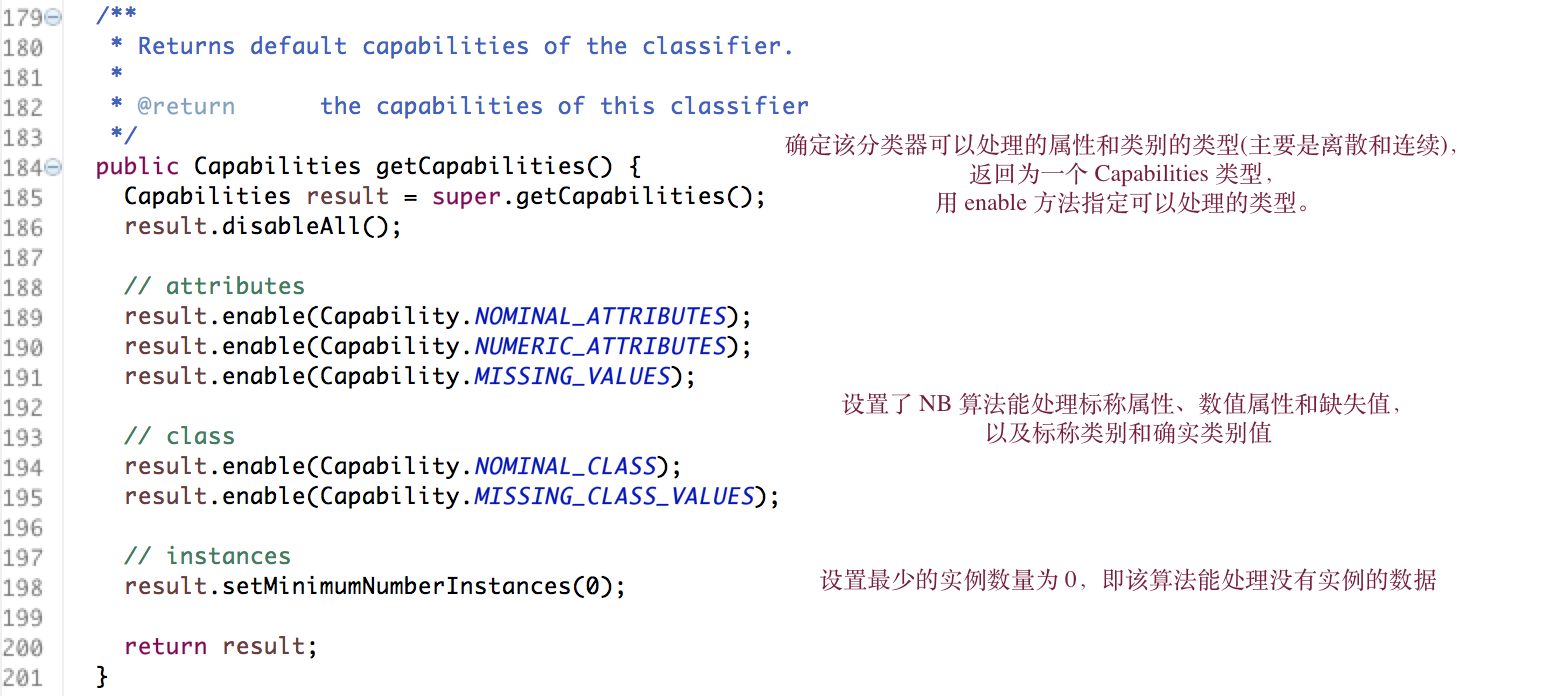
在Eclipse中引入weka.jar，并在其中找到实现朴素贝叶斯的源码。



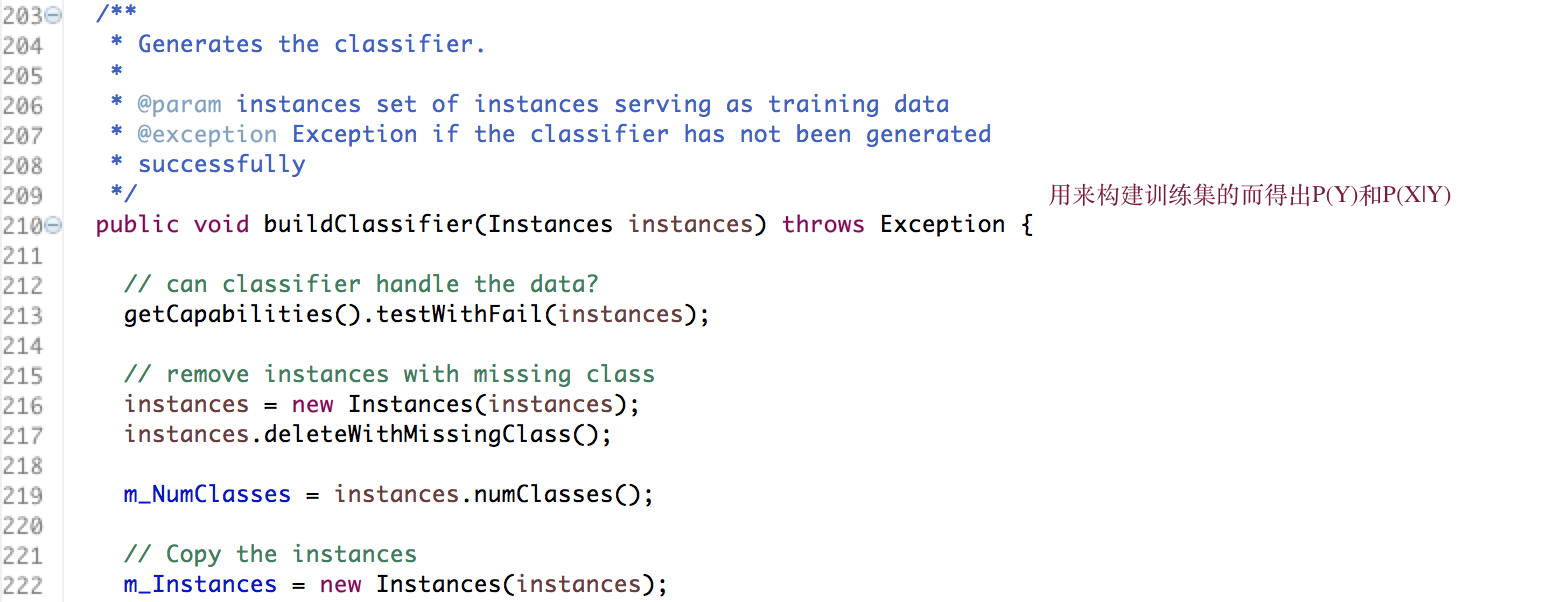
对其使用到的核心变量进行分析

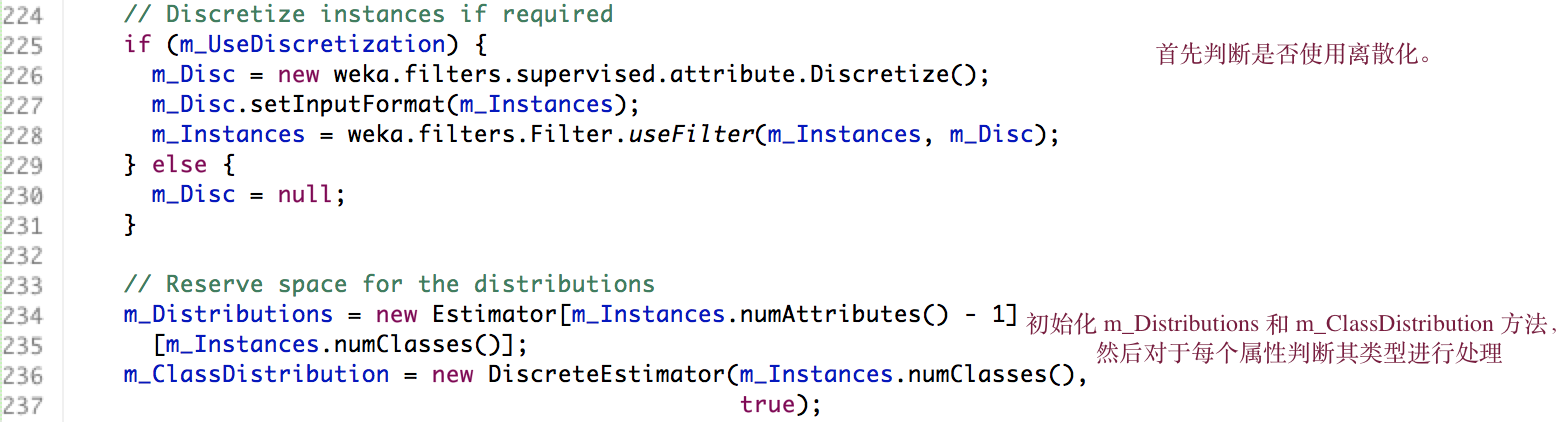


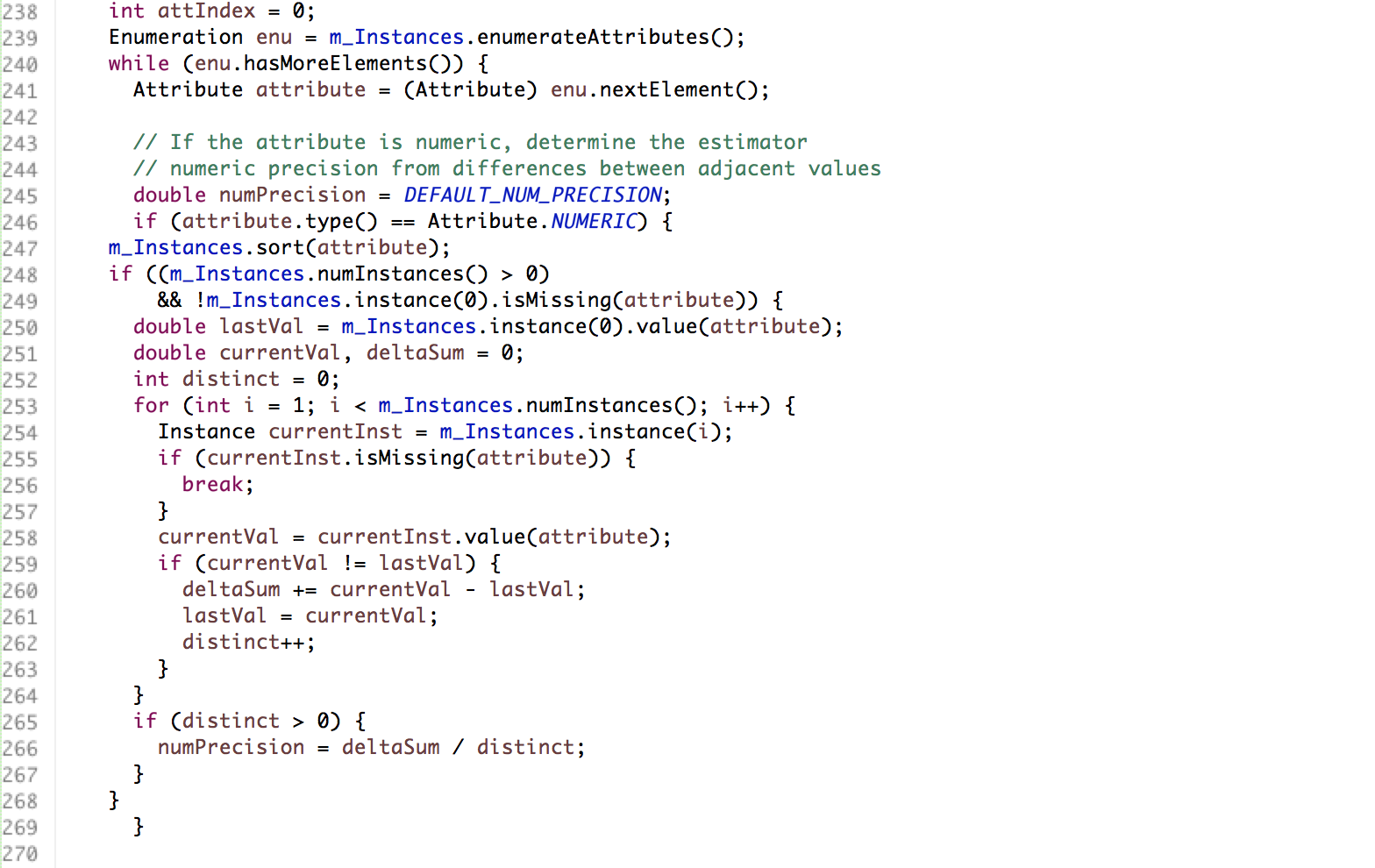
getCapabilities方法

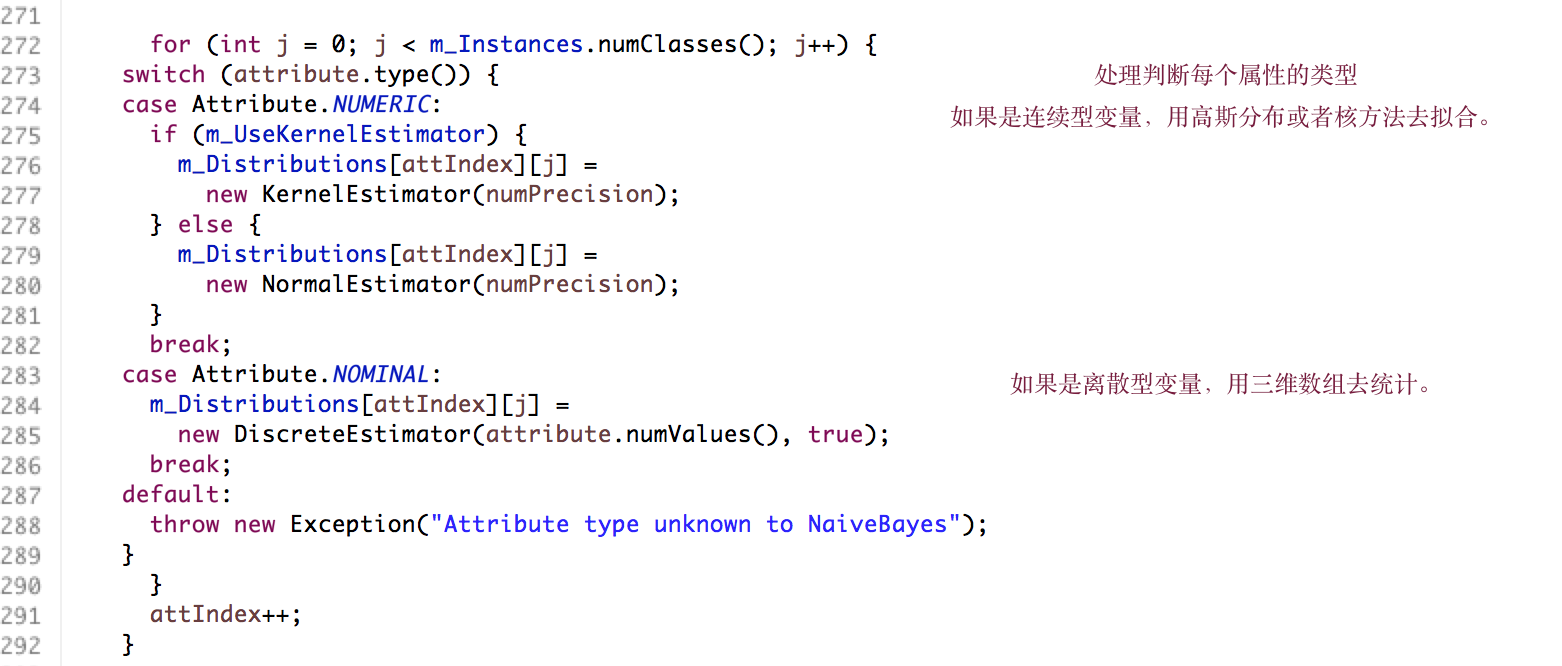


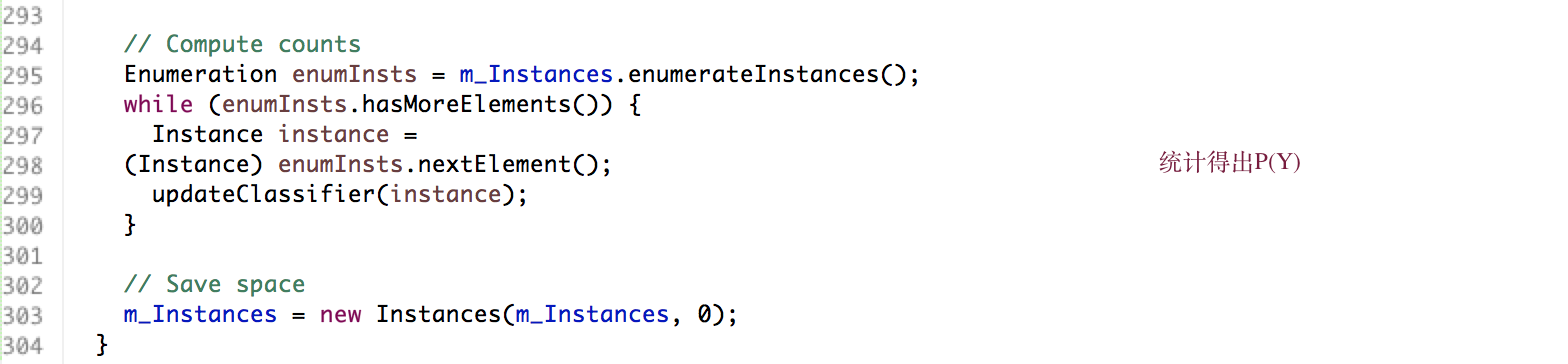
buildClassifier方法



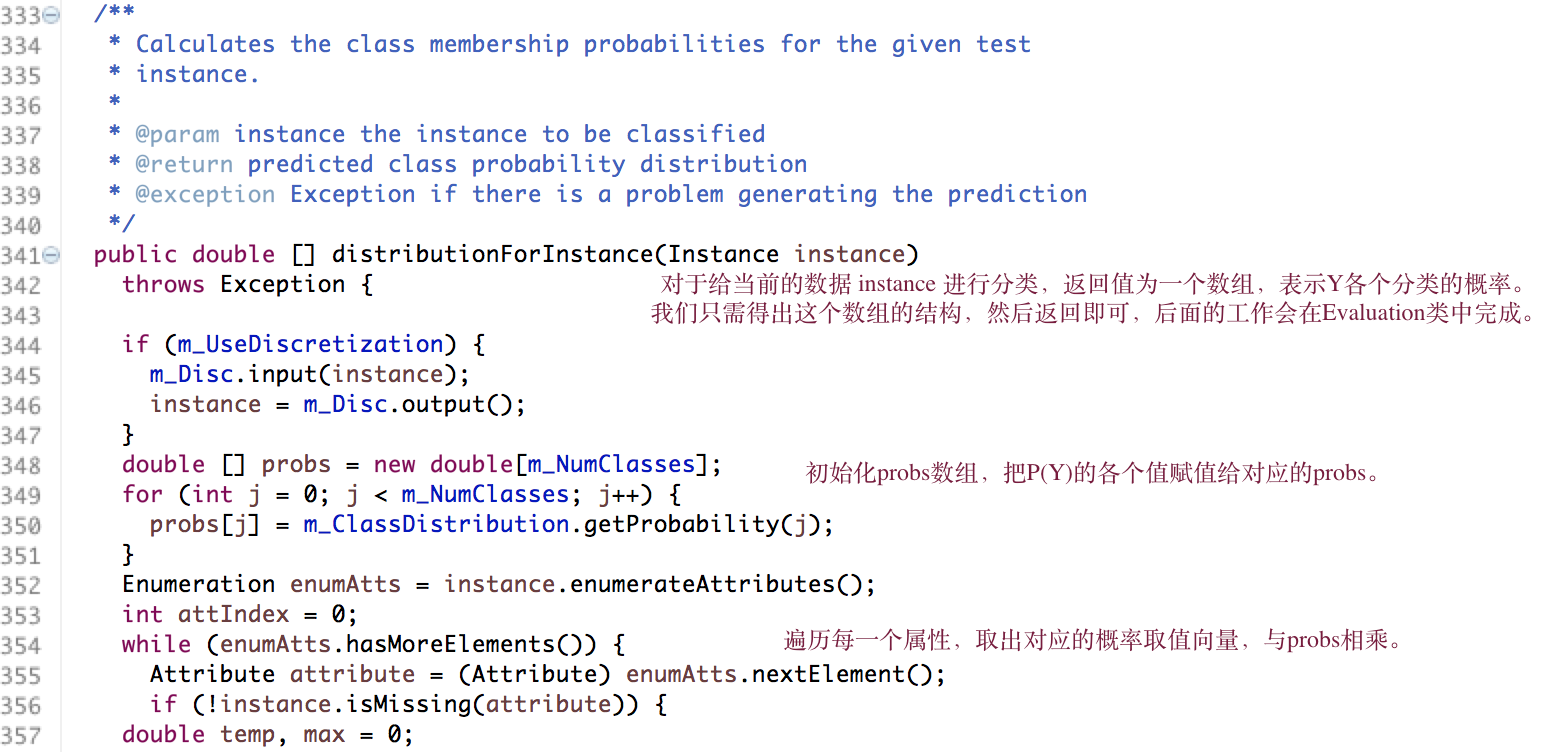


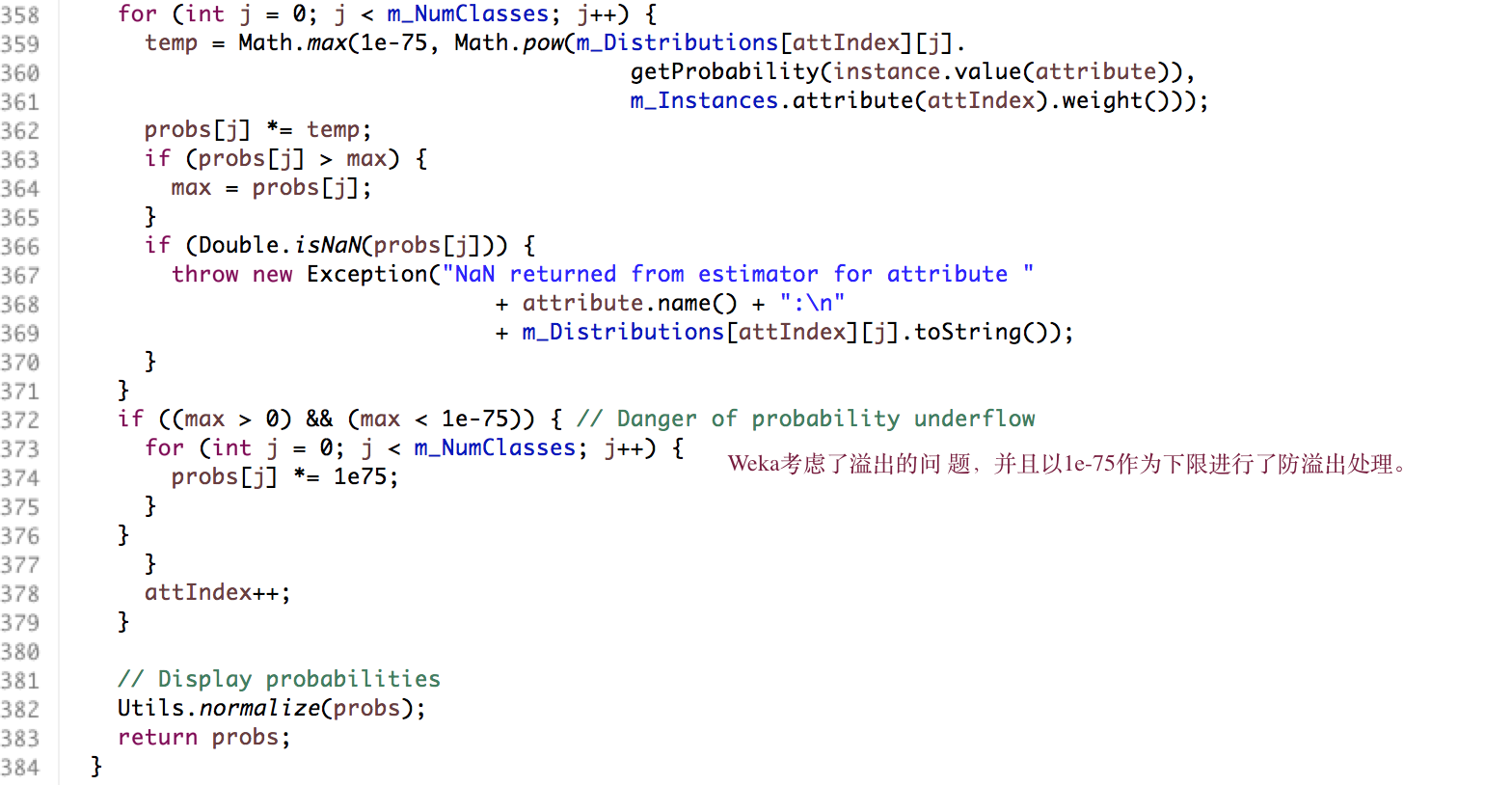






distributionForInstance方法



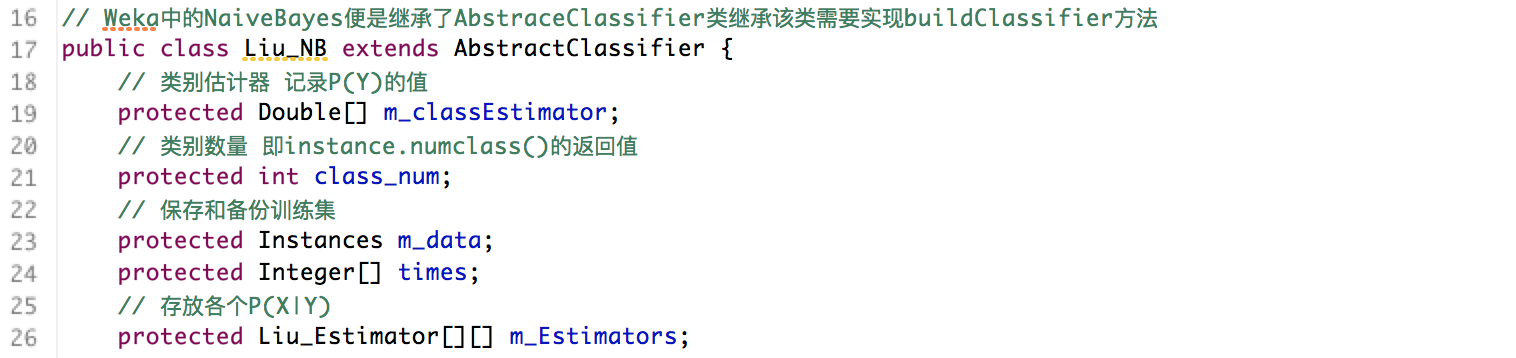


4. 自实现朴素贝叶斯分类器

在剖析了Weka的朴素贝叶斯的核心实现逻辑之后，便可以模仿着这些功能模块一个一个地动手实现。

4.1 分类器类

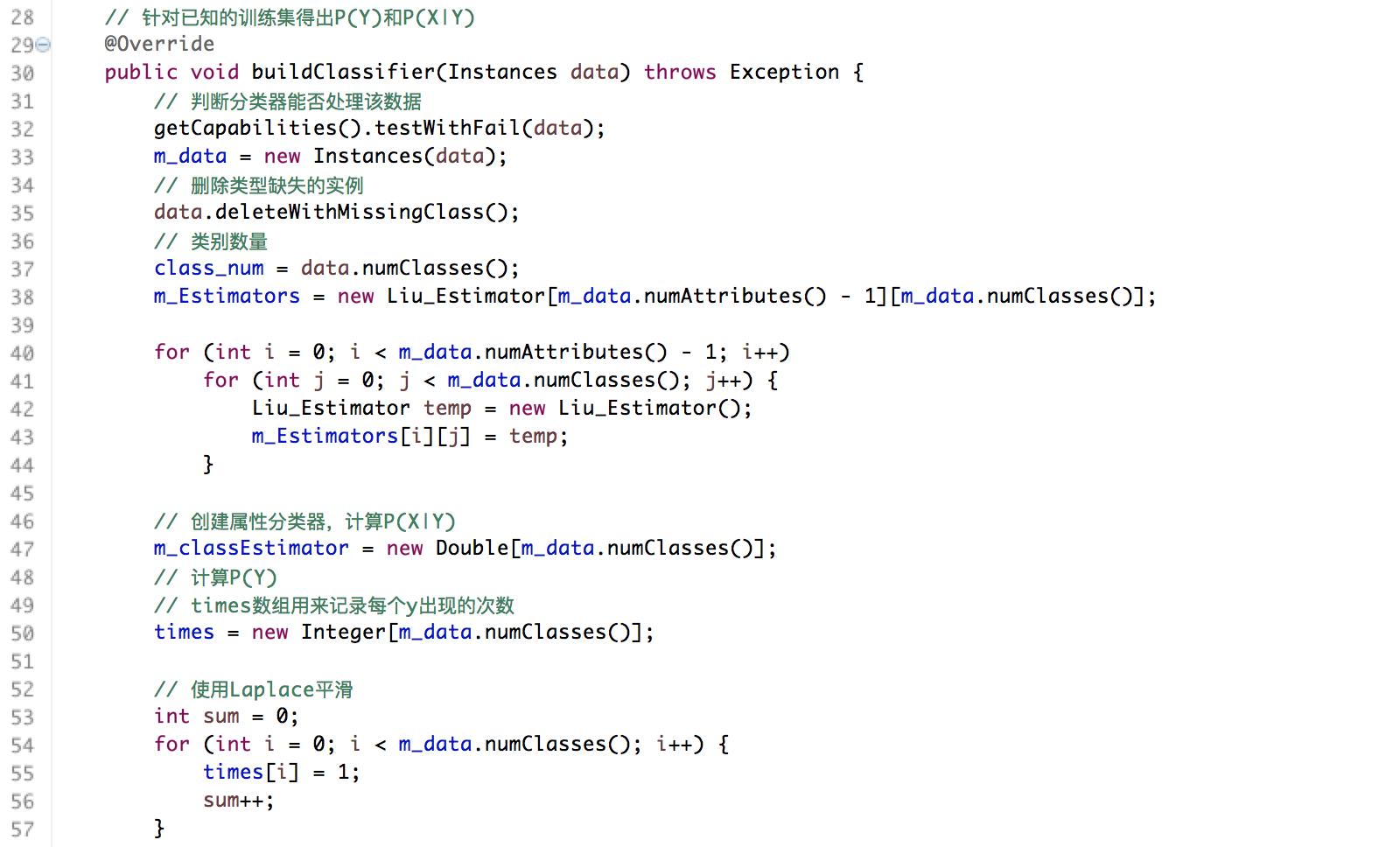
构建类，继承AbstractClassifier类



模仿Weka实现getCapabilities方法，使可以处理离散型和连续型属性



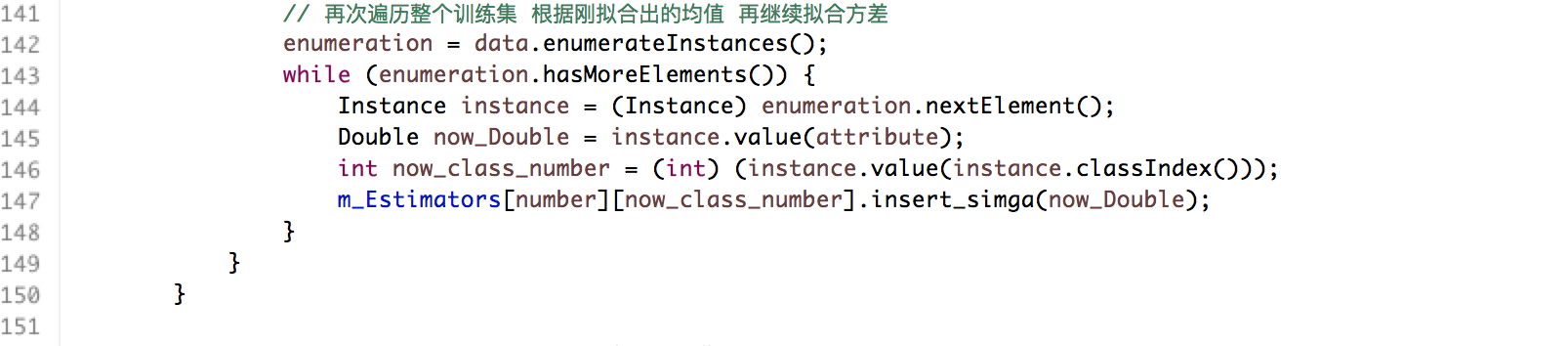
实现核心的buildClassifier方法，用来针对已知的训练集得出P(Y)和P(X|Y)

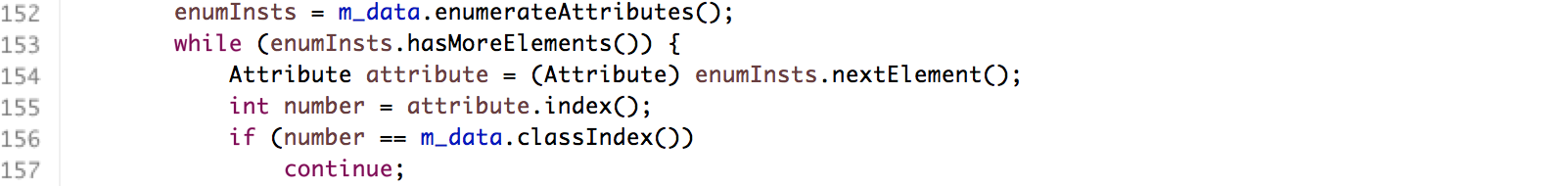


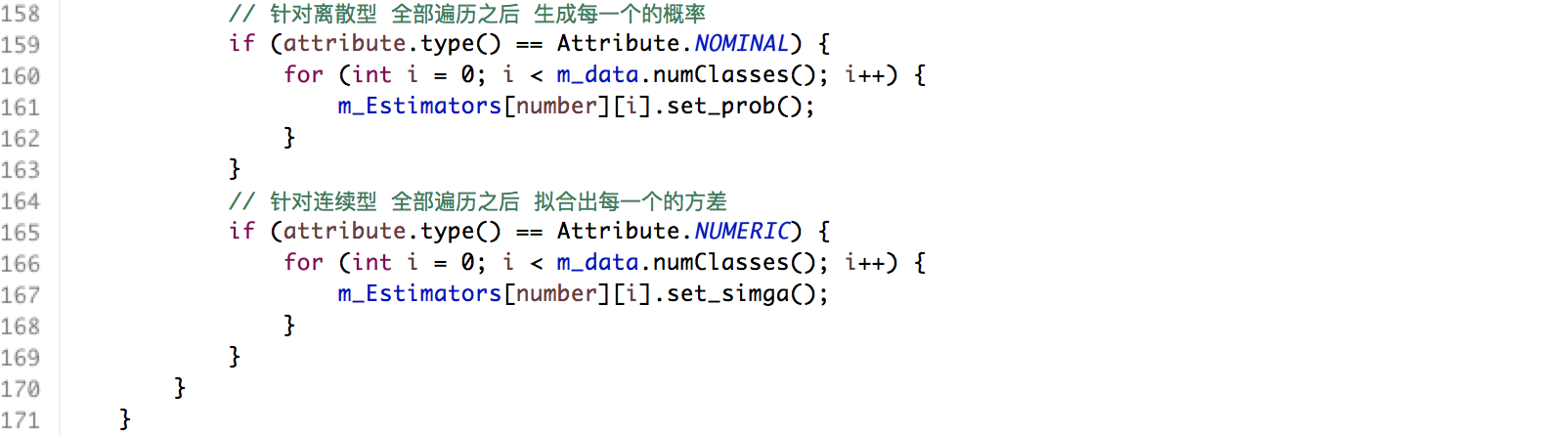












4.2 估计类

估计类的目的是协助分类器类进行各类数值计算，其中定义的参数、方法可被分类器对象使用。

在估计类中存放各个概率的值或是分布拟合的参数，从逻辑上总体分为离散型和连续型两种，分别定义不同的初始化函数、传值方式以及其他实现方法。

对于离散型：



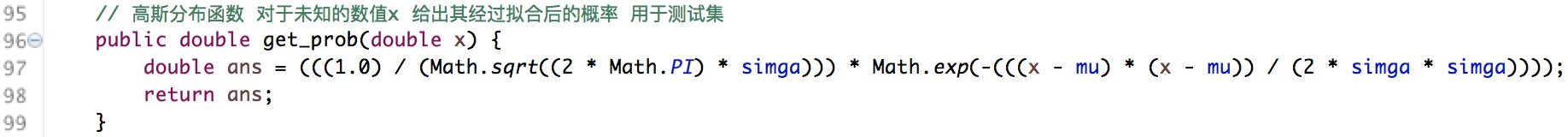


对于连续型：



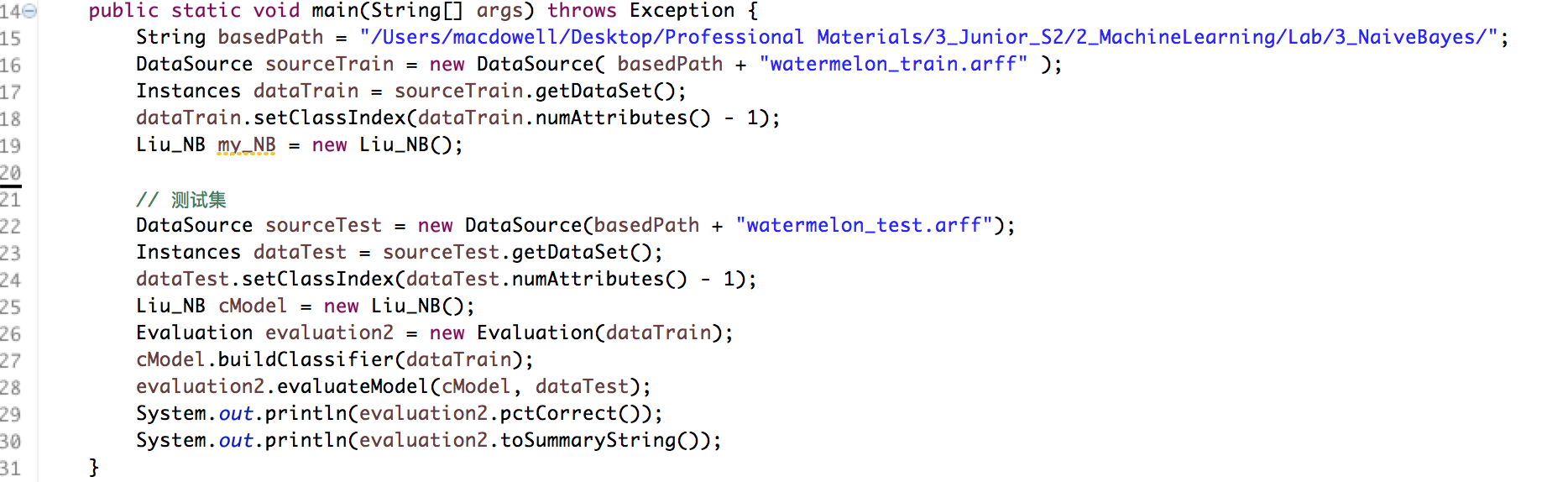






4.3 测试类

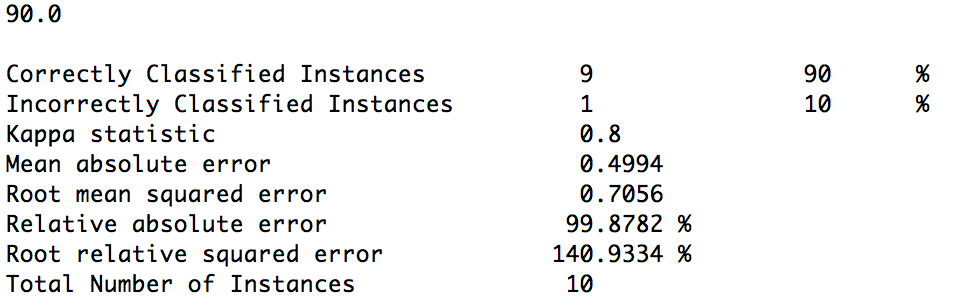
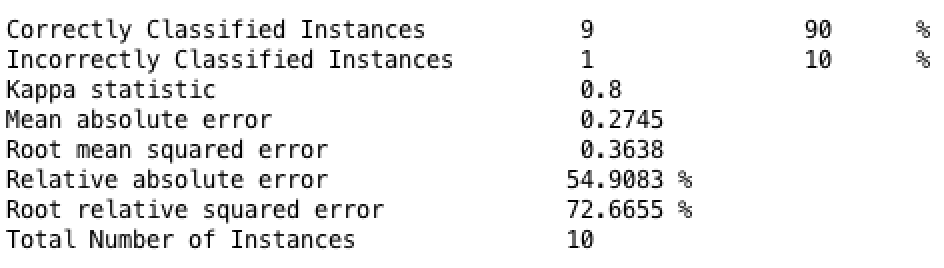
在完成自定义朴素贝叶斯后，编写测试类，并使用先前定义的分类器类，利用训练集与测试集进行模型构建与表现测试。



至此，自实现朴素贝叶斯分类器编写完成。

5. 自实现分类器表现分析

5.1 控制台运行输出

a. 自实现分类器 b. Weka分类器

图1 两种朴素贝叶斯分类器在watermelon训练集与测试集下的表现结果

5.2 结果分析

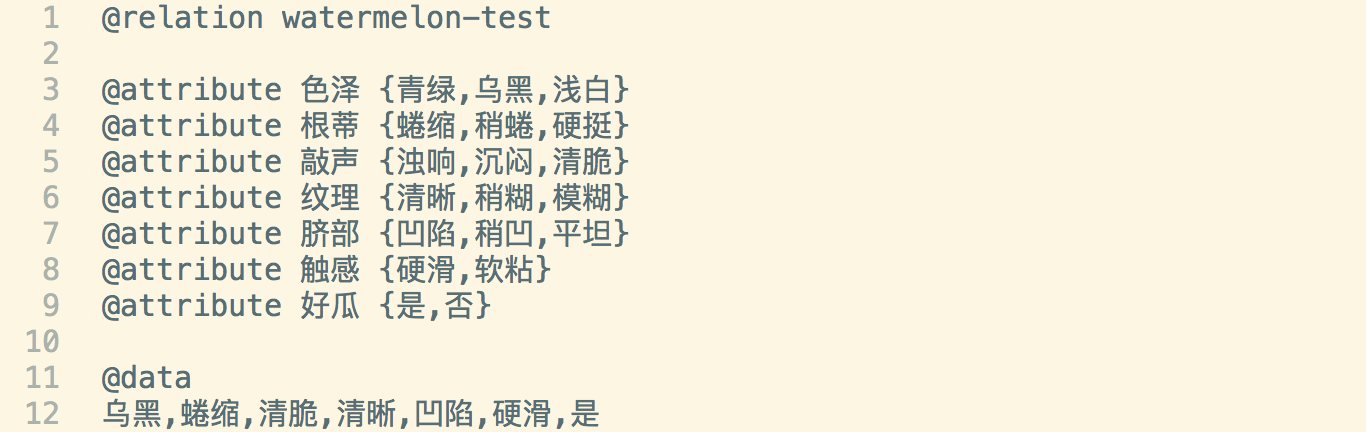
分析可知，自实现分类器在准确率上达到了和Weka原生分类器持平的能力，不过同时后续的几种错误率都明显比Weka分类器更高。

原因在于，自实现分类器囊括了朴素贝叶斯算法的整体实现框架，与Weka的实现逻辑相似，所以能得到同样的准确率。但Weka原生分类器中同时包含了诸多优化方法以提升多指标正确率。在后续还有很多对自实现分类器的优化空间。

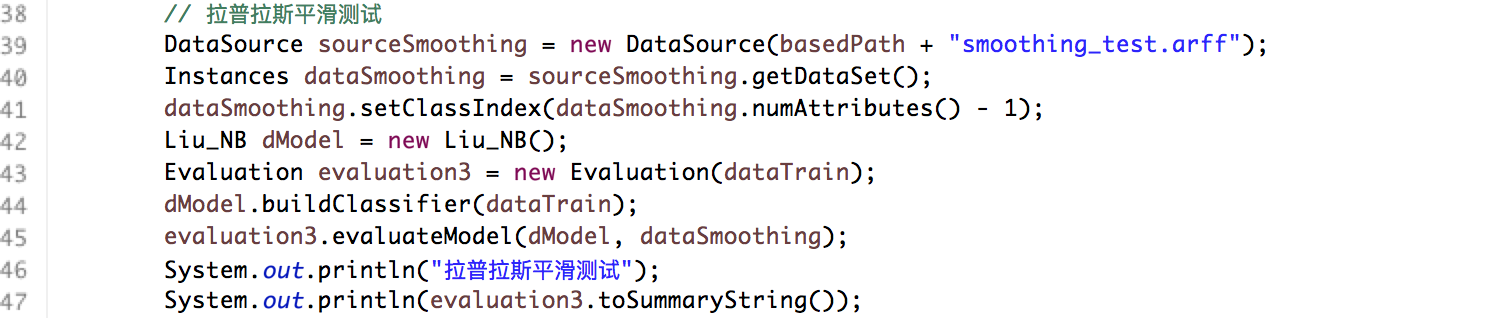
6. 拉普拉斯平滑验证

6.1 测试

将测试实例编入arff文件



在测试类中进行测试



控制台运行结果



结果分析

可见，虽然在训练集中没有实例与该测试实例前六个对应，但仍可见其被成功分类了，证明拉普拉斯平滑生效。

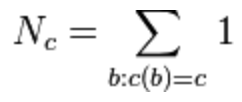
6.2 常用平滑算法对比与思考

Good-Turing算法

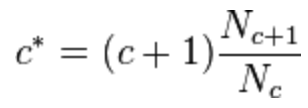
其基本思想是：用观察计数较高的N元语法数重新估计概率量的大小，并把它指派给那些具有零计数或者较低计数的N元语法。涉及的符号含义为:

c：某个N元语法出现的频数。

Nc：出现次数为c的N-gram 词组的个数，是频数的频数。



c\*：Good-Turing平滑计数



Laplace Smoothing算法

在前文中对原理已经提及，对本学期最后的关于垃圾邮件处理的课程设计有相应的一些联想：

垃圾邮件的情境中，有一个单词在字典之中，但是它从未在训练集中出现，那我们利用极大似然估计计算出来的朴素贝叶斯分类器前两个参数的估计值都是0，即不管是垃圾邮件还是正常邮件中这个单词都没出现过， 那么我们利用贝叶斯公式计算p（y=1|x）的时候，分子因为乘上了一个参数为0的值变成0，分母也一样变成0，那么就变成0/0不可预知，为了解决这个问题，也同样需要使用拉普拉斯平滑。

**四、写在最后**

感谢卫金茂老师以及助教学长学姐对本次实验的细致指导和悉心解答。

报告中如出现理解偏差也望指正。

刘嘉洋

NKCS, 1412620

[macdowellliu@163.com](mailto:macdowellliu@163.com)

2017.3.16