机器学习第一次作业-朴素贝叶斯分析报告

姓名:王龙涛

学号:2014011406

Goal:

* Implement a Naïve Bayes classifier and test it on a real dataset
* Have basic ideas about:
  + How to implement and apply a machine learning algorithm on a practical dataset
  + How to evaluate its performance
  + How to analyze your results

Naïve Bayes classifier:

Details:

* The Adult Data set: (<http://archive.ics.uci.edu/ml/datasets/Adult> )
* Aims at determining whether a person’s income exceed a certain amount (50k)
* Each record (a line in the file) corresponds to a person, and has 15 attributes
* The last attributes, (<=50k or >50k), is the class label
* There are continuous and categorical attributes, and some of the values are missing (denoted by “?”)

Evaluate the Performance:

* Train my classifier on training set and test it performance test set.
* The accuracy is calculated by .

Issue 1, the impact of the size of training set:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 采样率 | test1 | test2 | test3 | test4 | test5 | minimum | maximum | average |
| 0.001 | 0.80075 | 0.79031 | 0.77354 | 0.7694 | 0.76973 | 0.7693631 | 0.800749 | 0.780738 |
| 0.01 | 0.80837 | 0.8322 | 0.81715 | 0.8249 | 0.80941 | 0.8083656 | 0.832197 | 0.818402 |
| 0.05 | 0.84098 | 0.84276 | 0.83564 | 0.8405 | 0.83797 | 0.8356366 | 0.842762 | 0.839568 |
| 0.5 | 0.84325 | 0.83994 | 0.84116 | 0.8398 | 0.84307 | 0.8397518 | 0.843252 | 0.841434 |
| 1 | 0.83895 |  |  |  |  | 0.8389533 | 0.838953 | 0.838953 |

从上述实验结果可以看出，随着采样率的不断降低，实验预测结果大体上是呈下降趋势的。这说明训练集越大，预测结果也就越好。

Issue 2, zero-probabilities:

如果测试集中出现了训练集中没有出现的属性，我采用了平滑概率的方法，

对应的程序代码如下：

Double n\_xi\_y = *accNum*.get(idx).get(key).doubleValue() + bias;

Double n\_y = *accNum*.get(14).get(pos).doubleValue() + *cateNum*[idx] \* bias;

entry.setValue(n\_xi\_y / n\_y);

其中，cateNum为手动设置的M参数，大概估计各种属性的可能值。当样率为1时，bias为0.4的预测结果最好。

Issue 3, continuous and missing attributes:

Continuous attributes:

对于连续属性，我将其进行了区间化，不同的连续属性分段的区间不同，通过估计 和手动调整参数，得到较优的结果。

不同属性分段区间如下，对应代码为QuickNaiveBayes.java 207行，decreNum函数：

|  |  |
| --- | --- |
| Attribute | bucket size |
| Age | 6 |
| Fnlwgt | 99000 |
| education-num | 5 |
| capital-gain | 4000 |
| capital-loss | 220 |
| hours-per-week | 9 |

Missing attribute:

对于缺失属性，同不做处理相对比，把缺失当做一种全新的属性的预测结果更优。

所以，代码中，我把缺失当做了一种属性，进行分析和预测。

Program Details:

本程序使用java语言编写，安装环境为jdk8，IDE推荐eclipse，

程序代码文件为(其他文件没有使用，无需注意)：

src/ --

|--- QuickNaiveBayes.java 朴素贝叶斯代码相关

|--- MainDriver.java 程序入口

data/ --

|--- adult.names 数据说明文件

|--- adult.train 训练集

|--- adult.test 测试集

Design Ideas:

Class QuickNavieBayes :

quickInit() 设计用于程序的变量的初始化；

quickInfoReader() 设计用于从训练集文件中读入数据并解析与统计。

quickParseInfo() 设计用于数据的解析和统计，

decreNum() 设计用于连续数据的离散化，程序参数为实际调整至最优。

quickPosCal() 设计用于计算不同属性的概率，便于朴素贝叶斯使用。

quickNavieBayes() 设计用于预测不同属性结合的结果。

Class MainDriver :

main() 程序入口，调用朴素贝叶斯类，实现程序的调用。

Design Results:

最开始的情况下，即没有加入任何的优化，预测成功率在72%左右，之后加入了连续属性区间离散化的优化之后，成功率提升至了80%左右，在之后，考虑到测试集中空属性的问题，加了平滑概率的优化之后，成功率提高到了83.0%。

但是同其他人做出的NavieBayes数据统计结果进行对比，发现83.0%的成功率同还是存在不小的差距的，所以，在做出了上述的成果之后，我进一步深入分析了数据中的特征，对于程序参数进一步的优化。

Futher Analyse:

在不断调整的同时，我发现朴素贝叶斯的成功率一直在83.0%左右，并不能够进一步的提升。而连续函数离散区间化我已经尽可能调整到最优了，那么剩下的就是测试集中的空词条问题，我在平滑概率的程序中是通过估计和手动调整M的值的，对应的变量代码如下：

**public** **static** Integer[] *cateNum* =

{

12, 8, 100, 16, 10, 7, 14, 6, 5, 2, 100, 50, 100, 30000

};

最初的时候，cateNum的值是用于调整测试集中空词条的概率的，但是随着一次偶尔的参数调整，我发现了一个很奇怪的现象。当cateNum的最后一个参数，即对应native-country的M值越大时，预测准确率有着明显的提升。

按照正常的估算，native-country的值应该在84左右，但是从下面的统计数据中可以看出，随着native-country的值逐渐增大，预测的效果是越好的。

|  |  |
| --- | --- |
| native-country | |
| M Value | Accuracy |
| 40 | 0.830354401 |
| 80 | 0.830477243 |
| 200 | 0.830661507 |
| 500 | 0.830722929 |
| 1000 | 0.831030035 |
| 10000 | 0.834223942 |
| 30000 | 0.838953381 |
| 100000 | 0.843130029 |
| 1000000 | 0.84509551 |

我初步认为是由于随着M值的增大，国家属性对于最终的预测结果的影响会不断降低，所以可以推出国家对于市民的年薪是否超过50K的影响并不是很大。

但是后来的研究结果表明，同样，对于其他的属性进行类似的提升，发现预测成功率也是随着提升的，这推翻了我之前的结论，因为这不仅仅是native-country属性才导致的这种现象。

经过几天的思考，我仍然没有解决这个问题，希望老师和助教能够给予我帮助和解答。