机器学习概论朴素贝叶斯分类器实验报告

2016011359 计65 曾军

实验目的

本次实验的主要目的是训练出一个可以用于过滤垃圾邮件的朴素贝叶斯分类器，并且在训练分类器的过程中学会怎么将学到的机器学习算法应用到实际的数据集中去解决实际问题，学会如何评价模型的效果和分析模型的输出，并且了解一些机器学习中常用的概念和方法。

实验知识

1. 贝叶斯公式： $P(B|A)=\frac{P(A|B)P(B)}{P(A)}$

   贝叶斯公式为我们打通了先验概率和后验概率之间的关系，让我们可以用通过经验得到的先验概率去计算和预测难以估计的后验概率。在机器学习中应用这种先验概率和后验概率之间的关系，我们就有了一系列的贝叶斯算法，而本次实验实现的是贝叶斯算法中最简单的一类——朴素贝叶斯分类算法。

2. 朴素贝叶斯分类算法：
   - 设$x=\{a_1,a_2,\dots,a_m\}$为一个待分类项，而每个a为x的一个特征属性，

     且有类别集合$C=\{y_1,y_2,\dots,y_n\}$
   - 计算$P\left(y_1|x\right),P\left(y_2|x\right),\dots,P\left(y_n|x\right)$，如果
     $P\left(y_k|x\right)=\max\left\{P\left(y_1|x\right),P\left(y_2|x\right),\dots,P\left(y_n|x\right)\right\}$，则$x\in y_k$
   - 计算后验概率实际上是预知未来，实际上是不可实现的，但是有了贝叶斯公式的帮助，我们可以通过统计和计算先验概率，用先验概率去预测后验概率，最终实现待分类项的分类。

问题分析

垃圾邮件分类是一个典型的朴素贝叶斯分类问题，待分类项为一封邮件，而邮件的类别集合$C=\{spam,ham\}$，我们可以通过选取一定的特征来实现对一封邮件进行垃圾邮件或正常邮件的分类，而最简单且明显的特征就是邮件中出现的中文词语和单字。

选出特定的特征之后，我们运用贝叶斯公式$P\left(y_i|x\right)=\frac{P(x|y_i)P(y_i)}{P(x)}$计算后验概率，其中分母是一个固定的常数而我们的目标是求集合中的最大值，因此我们可以忽略掉分母进一步求$P(x|y_i)P\left(y_i\right)$。而在朴素贝叶斯分类中，我们假设各个特征属性条件独立，有：
$P(x|y_i)P\left(y_i\right)=P\left(a_1|y_i\right)P\left(a_2|y_i\right)\dots P\left(a_m|y_i\right)P\left(y_i\right)=P\left(y_i\right)\prod_{j=1}^{m}P\left(a_j|y_i\right)$，为此我们只需要统计各个特征在各个类别下出现的频率再将它们相乘即可。

对应到邮件分类问题，我们要统计的就是垃圾邮件和正常邮件的数目以及各个特征分别在垃圾邮件和正常邮件中出现的频数，最终用这些数据计算出对应的条件概率，通过贝叶斯公式算出后验概率，实现邮件分类。

代码实现

本次实验使用python3实现垃圾邮件分类，整个程序设计训练，验证，控制和画图四个部分：

`shell.py`：shell类似一个控制程序执行的脚本，他组织了程序的各个模块，形成了过滤器的统一入口。在shell程序的入口，可以选择训练（划分测试集和验证集），测试训练出来的模型的准确率，判断某个特定的邮件是否是垃圾邮件，五折交叉验证。

`model_training.py`：对shell中划分出来的训练集进行训练。用正则表达式提取邮件中的特征信息，在基本的要求中，我们只需提取中文词语和单字，因此使用 `re = u"[\u4E00-\u9FA5]+"` 作为提取的正则表达式，在增加特征时的做法也是如此，统计方法类似，将在后面的问题讨论中提出。我们记录下在垃圾邮件和正常邮件中每个特征出现的次数，以及我们训练的邮件中垃圾邮件和正常邮件的出现的次数。

`validation.py`：本模块提供了两个接口，一个是判断一封邮件是否是垃圾邮件的接口，一个是给定测试集计算所训练的模型的正确率的接口，第二个接口是在第一个接口的基础上做了进一步的封装。判断邮件的算法就是使用贝叶斯公式，通过计算每个特征在每个类别下出现的概率，最后将这些概率相乘并乘以类别概率得到一个“相对的”后验概率值，通过$\hat{y} = \operatorname{argmax}_y P(y) \prod_{i=1}^{n} P(x_i|y)$找出最可能的分类。其中我们要注意到其实每个特征的概率都是很小的，如果连乘的话很有可能因为浮点精度的问题导致很大的误差，因此对每个概率取对数将原本的连乘转换为连加，减小了误差，提高了准确率。

`plot.py`：用于绘制数据结果分析图

问题讨论

问题一：训练集的大小和正确率的关系

1. 使用随机采样的方式分别采样5%，50%，100%的训练集进行训练并汇报结果（考虑了其他特征的实现）：

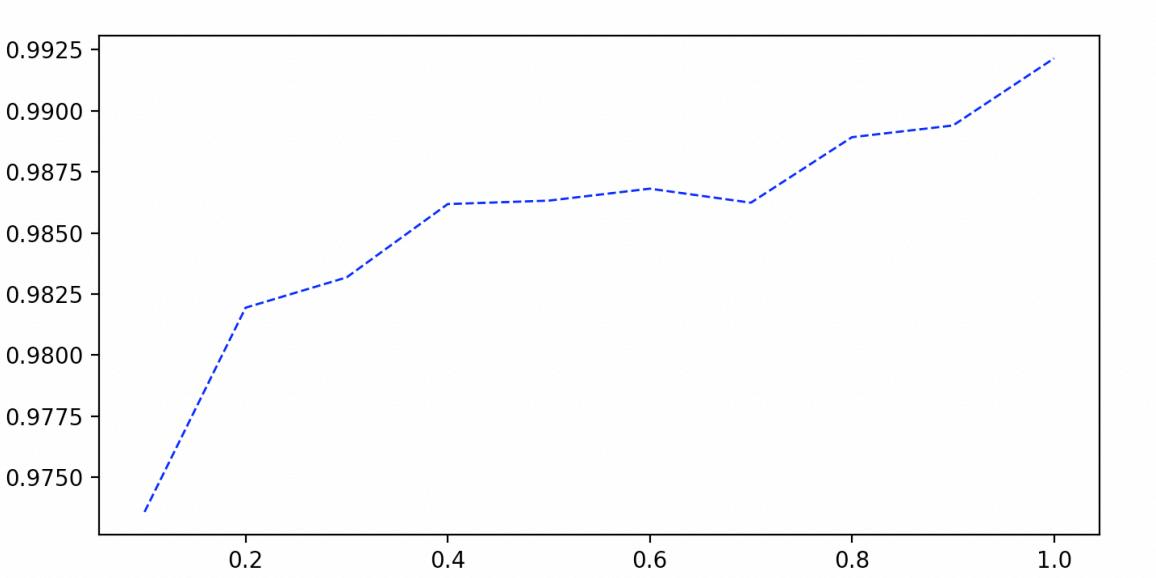
```
最后得到的结果为:
5%:
[0.9663416898792944, 0.9625502940266172, 0.9678118229650263,
0.9667285670071185, 0.9658000619003404]
min: 0.9625502940266172
max: 0.9678118229650263
average: 0.9658464871556793

50%:
[0.9866914268028474, 0.9873104302073661, 0.9855307954193748,
0.9851439182915506, 0.9873878056329309]
min: 0.9851439182915506
max: 0.9873878056329309
average: 0.9864128752708139

100%:
[0.990792324357784, 0.9889353141442278, 0.9896316929743113, 0.9887031878675333,
0.9891674404209223]
min: 0.9887031878675333
max: 0.990792324357784
average: 0.9894459919529558
```

根据结果可以看出，即使使用5%的训练集（五折交叉验证），我们也可以达到很高的分类正确率，但是随着训练集的规模增大，正确率也是在不断提升。

2. 为了研究正确率变化的趋势，我们分别计算各个训练集采样率下训练和测试的正确率并作图：



通过我们绘制的折线图可以看出，随着训练集使用比例（即规模）的增加，模型准确率不断增加，但是在50%左右的区间内正确率平缓甚至呈下降趋势。针对这一现象，猜测是混入的噪声的比例增大导致正确率不变甚至下降。具体原因没有深入研究，当然也有可能是比较偶然的原因。

问题二：零概率问题（注：图像横坐标均取对数处理）

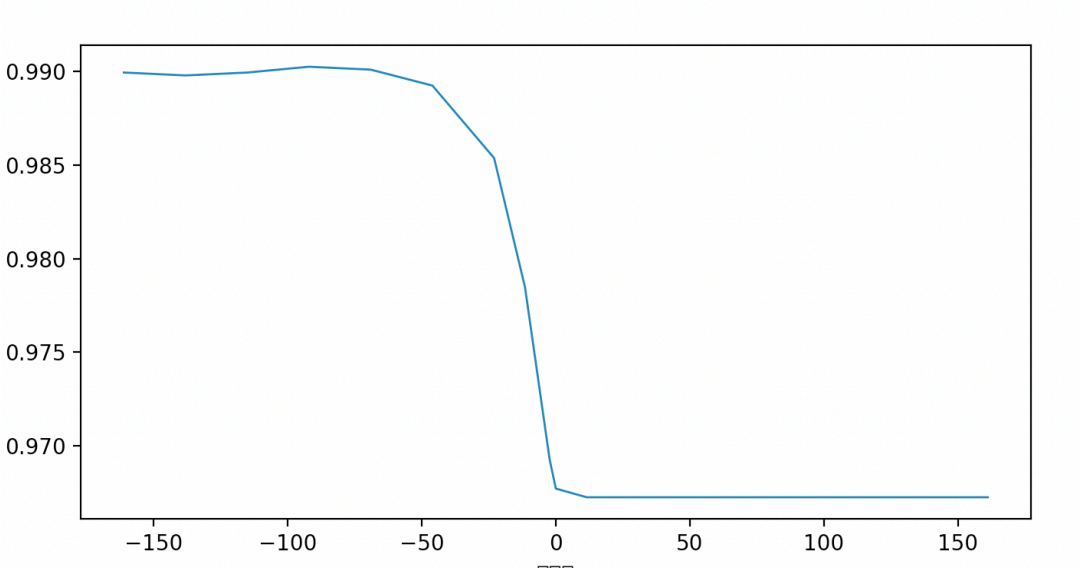
1. 零概率问题是指由于我们使得我们在通过连乘求后验概率的时候由于训练时特征统计的不完全导致待分类邮件中出现了我们没有统计到的特征，这一特征在我们的模型中的概率为零。根据公式$\widehat{P}(y=c|x_1,\ldots,x_i=k,\ldots,x_n)=0$该分类的最终概率为零，这显然是错误的并且会造成很大误差，因此考虑进行平滑操作或直接将该特征抛弃。
2. 研究各种方法对正确率的影响：
    1. 不进行平滑操作的正确率（进行五折交叉验证）：

```
[0.7726709996904984, 0.7778551532033426, 0.7828071804394924,
0.7778551532033426, 0.7756886412875271]
min: 0.7726709996904984
max: 0.7828071804394924
average: 0.7773754255648406
```

    从得到的结果来看不进行平滑操作直接丢掉零概率特征会造成较大的误差，因此平滑是本模型需要做的优化之一。

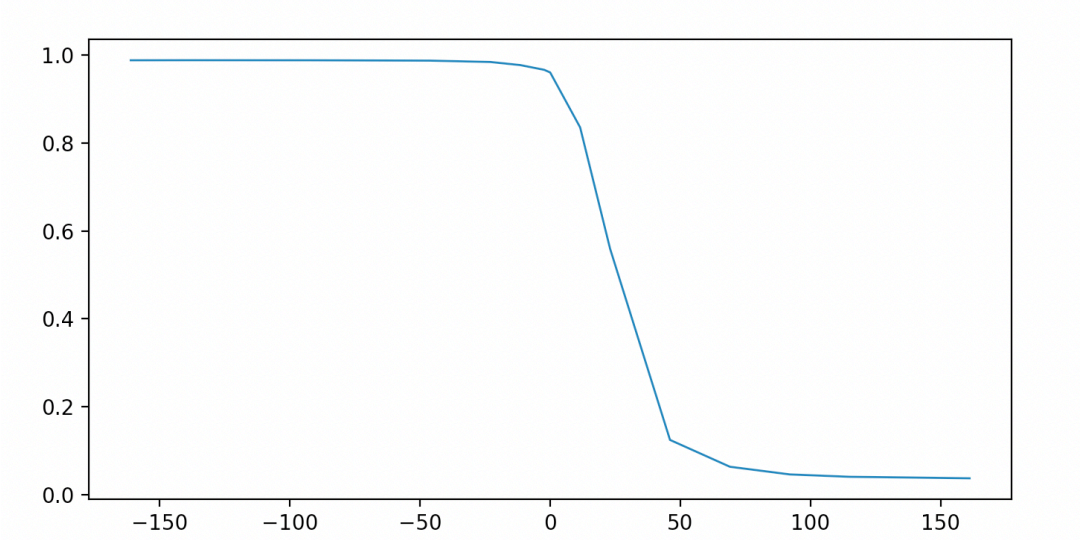
    2. 进行拉普拉斯平滑操作：$\widehat{P}\left(x_i=k|y=c\right)=\frac{\#\{y=c,x_i=k\}+\alpha}{\#\{y=c\}+M\alpha}$

    拉普拉斯平滑操作就是对于零概率现象，我们在分子和分母上分别加一个常数偏移，来平滑零概率现象造成的概率突变现象。如何取这个常数是拉普拉斯平滑的重点，下面分别对不同的拉普拉斯系数的平滑效果进行分析：



3. 使用一个常数进行偏移操作：

   其实拉普拉斯平滑就是通过加一个偏移来避免零概率问题的影响，想到其实我们可以更简单的通过加一个常数来避免零概率问题，对所加常数的大小进行研究：



   发现在e取合适大小时，对于正确率的贡献是几乎稳定不变的，但是这个常数必须比较小，和拉普拉斯平滑类似。

问题三：选取其他特征，提高分类效率

1. 只使用中文词语和单字进行训练得到的正确率为（五折交叉验证）：

```
[0.9868461776539771, 0.9867688022284122, 0.98785205818632, 0.9845249148870319,
0.9868461776539771]
min: 0.9845249148870319
max: 0.98785205818632
average: 0.9865676261219436
```

   发现正确率已经很高，但是其实我们还有很多可以用的特征没有提取。

2. 考虑三个特征：邮件的 `X-mailer` ， `X-priority` 以及发件邮箱

   经过统计发现，没有邮件客户端的邮件大概率是垃圾邮件，并且客户端大多是firefox和outlook，邮件的发送客户端也可以作为一个新的特征。

邮件的优先级表示了邮件的重要性，分1-5等，邮件的优先级也可以作为一个过滤特征。

识别垃圾邮件很重要的一点就是识别源，因此我们将每封邮件的发件邮箱提取出来作为第三个过滤特征。

提取特征所用的正则表达式分别为： `u"(X-Mailer): ([a-zA-Z ]+)"` ， `u"(X-Priority): ([1-5])"` ， `u"(From.*)@(.*)>"`

增加三个特征之后，再次进行五折交叉验证有：

```
[0.9895543175487466, 0.9906375735066543, 0.990328071804395, 0.9904054472299597,
0.9880068090374498]
min: 0.9880068090374498
max: 0.9906375735066543
average: 0.9897864438254411
```

因此增加上述特征之后，邮件分类的正确率确实有了一定的提升。参考往届学长做法，尝试增加上述三个比较有代表性的特征的权重，进行五折交叉验证有：

```
[0.9924172082946456, 0.9924945837202105, 0.9925719591457753,
0.9935778396781182, 0.993190962550294]
min: 0.9924172082946456
max: 0.9935778396781182
average: 0.9928505106778088
```

发现有比较大的提升，说明这种方法有一定的效果，可以被借鉴。

实验总结

本次实验是机器学习概论的第一次实验，实现的是比较简单的朴素贝叶斯分类算法，但是正确率也达到了99.8%，这也体现了概率的美妙之处。本次实验让我感触最深的不是算法的实现，因为算法本身并没有什么难以理解的地方，我感触最深的是参数的调节以及实现上的一些细节如平滑处理等。其实收获最大的也是这个部分，正是因为有了这些分析我们才会有各种突破，因此今后的学习生活中要多学会分析研究，发现新知识。