朴素贝叶斯分类器 实验报告

实现原理

在朴素贝叶斯分类器中，根据条件独立性假设，对于某个样本$\mathbf{x}$，要最大化的后验概率为

$$p(y|\mathbf{x}) \propto p(y)\prod_{i=1}^{n}p(x_i|y)$$

在实际实现中，由于数值稳定性问题，所有概率均取对数后相加。

训练的过程为根据训练集估计$p(y)$和$p(x_i|y)$的过程。基础的特征为bag-of-words，即使用各个单词出现的频数估计概率，特征分量x_i为样本中依次出现的单词。某个类别的频数为此类别中所有样本单词的个数之和，某类别下某单词的频数为此类别中所有样本中此单词出现的次数之和。这样

$$p(y)=\frac{\text{类别}y\text{的频数}}{\text{所有类别的频数之和}},\quad p(x_i|y)=\frac{\text{类别}y\text{下单词}x_i\text{的频数}}{\text{类别}y\text{的频数}}$$

评估方法

对于训练和测试，采用5折交叉验证，即将数据集均匀划分为5份，每次取4份训练，1份测试，5次的结果进行平均。

采用Accuracy、Precision、Recall、F1 Score这4种评价指标，计算方式如下：

$$Accuracy=\frac{tp+tn}{tp+tn+fp+fn}$$

$$Precision=\frac{tp}{tp+fp}$$

$$Recall=\frac{tp}{tp+fn}$$

$$F1=\frac{2\cdot Precision\cdot Recall}{Precision+Recall}$$

其中tp, tn, fp, fn分别指代True Positive、True Negative、False Positive、False Negative。

问题与讨论

数据去噪与预处理

提供的数据集是十分Noisy的，体现在以下几个方面：

- 没有将元数据和正文分开
- 单词间可能包含杂乱的标点符号
- 可能含有html标签
- 可能含有某种编码的图片

对于第二、第三条，我使用BeautifulSoup4对html标签进行处理，之后将标点符号替换，再进行文本分割后使用pyenchant检查每个连续的字母块是否为英文单词。第一条和第四条没有很好的处理方式，这可能是影响性能的一个因素。

训练集大小

选取训练样本比例为5%、50%、100%，不实现平滑时，结果如下：

选取比例	Accuracy	Precision	Recall	F1 Score
5%	0.4296	0.5511	0.6126	0.5802
50%	0.7133	0.7521	0.8271	0.7878
100%	0.7762	0.8012	0.8675	0.8330

在平滑系数为1e-5时，结果如下：

选取比例	Accuracy	Precision	Recall	F1 Score
5%	0.9577	0.9870	0.9468	0.9665
50%	0.9816	0.9906	0.9808	0.9857
100%	0.9847	0.9897	0.9865	0.9881

可以看出

- 当不进行平滑时，训练集增大会较多地增加准确率，且训练集较小时，准确率甚至低于50%。这是由于不平滑时，若遇到未在训练集中出现的单词，将会直接忽略，这使得测试样本的特征被极大削减了。朴素贝叶斯是通过频率估计概率，因此需要较大的训练集来保证估计的准确性
- 进行平滑时，训练集增大也会增加准确率，但增加的幅度较小，这是由于未在训练集中出现的单词被赋予了一个较小的概率，其影响也被纳入在内了

零概率问题

对于某类别下未在测试集中出现的样本，会导致$p(x_i|y)=0$，这样乘积的后验概率为0导致无法预测。计算中，对0取对数也会造成数值问题。一种方法是忽略未在训练集中出现的单词，即不平滑。另一种方法是采用如下公式

$$p(x_i|y)=\frac{\text{类别}y\text{下单词}x_i\text{的频数}+\alpha}{\text{类别}y\text{的频数}+M\alpha}$$

其中M为类别个数。采取不同的平滑系数，结果如下：

α	**Accuracy**
1e-50	0.9819
1e-45	0.9820
1e-40	0.9822
1e-35	0.9822
1e-30	0.9822
1e-25	0.9825
1e-20	0.9826
1e-15	0.9831
1e-10	0.9841
1e-5	0.9847
1e-4	0.9845
1e-3	0.9844
1e-2	0.9844
1e-1	0.9830
1e0	0.9819
1e1	0.9797
1e2	0.9701
1e3	0.9105
1e4	0.6816
不平滑	0.7762

从中可以看出，随着α从1e-50到1e4变化，准确率先上升后下降，在1e-5处达到极值，这说明要选择一个适中大小的平滑系数。同时，平滑比不平滑的效果好很多。

其它特征的选取

通过对邮件元数据的观察，我选择了提取"From:"后邮箱的域名（如果有）。由于邮件头十分Noisy，只能尽可能进行匹配、替换特殊符号以去噪，但聊胜于无。如在某种数据集的划分中，"lists2.u.washington.edu"在非垃圾邮件中出现了169次，而在垃圾邮件中没有出现（或过于Noisy以至于提取失败），这可以作为一个重要的特征。

将bag-of-words和mail独立看待，将他们的后验概率相乘，并给mail一个权重以计算总的概率用以比较。结果如下：

权重	Accuracy	Precision	Recall	F1 Score
0.1	0.9848	0.9898	0.9865	0.9882
0.5	0.9853	0.9907	0.9865	0.9886
1.0	0.9856	0.9915	0.9861	0.9888
2.0	0.9852	0.9924	0.9846	0.9885
3.0	0.9846	0.9931	0.9829	0.9880
5.0	0.9814	0.9941	0.9768	0.9854

在权重合适时，发件邮箱的特征对准确率有微小提升，但没有想象中的大，这可能是因为提取算法对噪声鲁棒性比较差。同时注意到，随着邮箱信息权重的增大，Precision会增加，而Recall会降低，这说明更少的非垃圾邮件被误判为垃圾邮件，而更多的垃圾邮件被误判为非垃圾邮件。