实验目的

- 实现NaïveBayes分类器并在真实数据集上进行测试
- 学会如何在实际数据集上实现和应用机器学习算法
- 学会如何评价模型的效果
- 学会如何分析实验结果

实验原理

分类模型中有M个样本，每个样本有N维，输出的类别有C类。样本例如$(X_1^1, X_2^1, X_3^1, \ldots, X_n^1, Y_1)(X_1^2, X_2^2, X_3^2, \ldots, X_n^2, Y_2)$，从样本中我们可以得到先验概率$P(Y)$(K=1,2...C)及条件概率$P(X|Y)$，然后得到联合概率为：$P(XY)$，定义联合概率为

$$P(XY) = P(Y = C_k) * P(X = x|Y = C_k)$$

$$= P(Y = C_k) * P(X = (x_1, x_2, \ldots, x_n)|Y = C_k)$$

根据**朴素贝叶斯假设**，假设X的n个维度之间互相独立，得到

$$P(X = (x_1, x_2, \ldots, x_n)|Y = C_k) = \sum_{i=1}^{n} P(X_i = x_i|Y = C_k)$$

得到朴素贝叶斯原理

$$C_{result} = argmaxP(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}$$

其中，$P(Y = C_k) = \frac{m_{c_k}}{m}$是类别在训练集中中出现的频数。上式中由于分母都一样都是$P(X)$，那么只要计算分子最大化即可。即

$$C_{result} = argmaxP(Y|X) = argmaxP(Y) * \prod_{j=0}^{n} P(X = x_j|Y = C_k)$$

在本问题中，由于是离散的特征，所以采用离散型的拉普拉斯平滑：

$$P(X = x_j|Y = C_k) = \frac{x_j + \lambda}{m_k + n\lambda}$$

，其中λ是拉普拉斯平滑参数。

所以模型训练的过程中需要得到的是对于$P(y)$和$P(x_i|y)$的估计值。模型测试阶段需要输出

$$argmax_y P(Y) * \prod_{j=0}^{n} P(X = x_j|Y = C_k)$$

也即使上式值最大的y的值

数据分析

实验所提供的数据来自于中文邮件数据集，我直接使用了已经经过分词的版本。共有64620个邮件，邮件包含两个部分:前一部分是邮件头，其中包含了该邮件的相关信息，包括发送方、接收方、发送时间、邮件主题等等，后一部分是邮件正文。两部分用空行隔开，利于分割提取。label/index文件是数据集的标签。共有42854个垃圾邮件，21766个非垃圾邮件。

数据处理部分在代码实现部分进行阐述。

代码实现

data_process/process.py

这部分是数据处理的部分，我将训练的部分也一并写在了这里。

- `ProcessEmail` 类

  此类是构建一个email对象，通过输入路径，将文件读入，由于数据集的特征可以很方便的将正文和邮件头分开，通过正则表达式等进行匹配中文、发件者、发件时间（对 `issue 3` 的探究）等提取的特征

- `ProcessLabel` 类

  此类是将label文件读入，构建一个文件对象，提供了方便的接口可以查询一封邮件是否是垃圾邮件、计算先验概率 `P(y)` 、得到(非)垃圾邮件总数的功能

- `GetBagOfWords` 类

  此类是训练类。进行的操作包括：将数据分为5-fold，设置 `sampling_rate` （满足 `issue 1` 的要求进行对训练集规模的探究）。采样运用蒙特卡洛采样的方法，设置 `sampling_rate` 为0.05, 0.5, 1来进行对训练集规模的控制。当采样率为1时，训练集为4折，测试集为1折。

  对一个email对象的特征进行处理，将中文词语存到dict中去等操作。对于训练集中的数据，分别获取他的label，分别对spam和ham邮件中的词语建立dict存储，最后可以得到四个文件作为训练结

果：`words_bag_train`，`words_bag_test`，`words_ham`，`words_spam`， 四个训练结果用于测试的过程。

nb/nb.py

这部分是利用模型计算测试集邮件分类结果和测试的部分。

- `Nb` 类

  这部分分别读入测试集邮件，用 `ProcessEmail` 类实例化为一个对象，得到该邮件的词语 `word_list`，按照朴素贝叶斯公式分别计算被分为spam和ham的概率，哪个概率大则最终的分类结果就是哪个类。

  为了避免小浮点数连乘带来的精度损失，所有概率取了对数 `log`

- `Evaluation` 类

  对所有测试集的邮件进行正确率、精确率、召回率、F1指数统计。

模型的评价指标

本次实验采用了Accuracy，Precision，Recall，F1来评价。他们各自的意义如下：

	被判为spam	**被判为ham**
实际为spam	True Positive	False Negative
实际为ham	False Positive	True Negative

以下简记为TP, FN, FP, TN.

- Accuracy

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

- Precision

  表示被判为垃圾邮件的样本占实际为垃圾邮件的比率

$$Precision = \frac{TP}{TP + FP}$$

- Recall

  对实际为垃圾邮件的样本被正确找出的比率

$$Recall = \frac{TP}{TP + FN}$$

- F1-Measure

  兼顾两方面的性能

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

实验结果分析

实验用五折交叉验证的方式进行模型的测试。测试结果如下：

以下数据是用全部训练集训练得到的结果：

用普通平滑的词袋模型

折数	Accuracy	Precision	Recall	F1-Measure
1	0.872325581395	0.898455551501	0.900975609756	0.899713815989
2	0.889689922480	0.9384598012646	0.904450973990	0.9211415904682
3	0.8803100775193	0.9362884466270	0.8733124397299	0.9037046276662
4	0.8877519379844	0.9361419604056	0.8940511874567	0.9146125722372
5	0.8935483870967	0.9383401159205	0.8957033549146	0.9165261382799
平均值	0.884725181295	0.9295371751437	0.8936987131694	0.9111397489281

用拉普拉斯平滑的词袋模型（lambda = 1e-40）

折数	Accuracy	Precision	Recall	F1-Measure
1	0.956899224806	0.96014928967	0.972560975609	0.966315279292
2	0.9722480620155	0.9886024123049	0.9722494286647	0.980357730714
3	0.981937984496	0.9889630078835	0.9828833172613	0.9859137899764
4	0.9764341085271	0.9861756505576	0.9786718930136	0.982409443351
5	0.9694316436251	0.9874774232390	0.9653914067098	0.9763095238095
平均值	0.97139020469	0.9822735567310	0.9743514042516	0.9782611534285

可以从数据看出，普通平滑模型和拉普拉斯的平滑模型之间的差距很大，拉普拉斯平滑模型在正确率、精确率、召回率、F1值方面都有很大的提升。说明拉普拉斯的性能比较好。

包含发件人、诱导信息等信息的词袋模型

折数	Accuracy	Precision	Recall	F1-Measure
1	0.9637209302325	0.9666827619507	0.9765853658536	0.971608832807
2	0.9789147286821	0.9922711714695	0.9780171944716	0.9850926230406
3	0.9862015503875	0.9921193016488	0.9863789778206	0.9892408123791
4	0.9831007751937	0.9893518518518	0.9854738298362	0.9874090331523
5	0.9784178187403	0.9893946615824	0.9773984696880	0.9833599810505
平均值	0.9780711606472	0.98596394970	0.980770767534	0.983342256485

在增加了辅助信息之后，准确率提到了0.007个百分点，其他几个指标都有较大的提升，模型准确率有了进一步的提升。

issue讨论

issue 1 训练集大小对结果的影响

`sampling_rate` 作为GetBagOfWord对象的初始化参数给出，运用蒙特卡洛算法，如果调用 `random.random()` 的结果小于 `sampling_rate` 则将其选入训练集。分别用5%, 50%, 100%的训练集对普通平滑的词袋模型进行了测试，测试结果(Accuracy)如下：

折数	100%	50%	5%
1	0.8723255813953489	0.8627131782945736	0.7751937984496124
2	0.8896899224806202	0.8827906976744186	0.7622480620155039
3	0.8803100775193798	0.8630232558139534	0.7445736434108527
4	0.8877519379844961	0.872093023255814	0.7468992248062015
5	0.8935483870967742	0.8843268746238238	0.7519201228878648
MIN	0.8723255813953489	0.8627131782945736	0.744573643410852
MAX	0.8935483870967742	0.8843268746238238	0.7751937984496124
平均值	0.884725181295324	0.8729894059325167	0.7561669703140071

可以看出普通平滑模型中训练集越大时，朴素贝叶斯的准确率越高，但是训练集规模从50%增长到100%时正确率并没有增加很多，可能是模型的参数（普通平滑设置的零概率对应的概率值）不好，导致正确率不能增长的很高。

下面是没有加入其他信息（只包含拉普拉斯平滑）的模型随着训练集规模增加的测试值的平均值($\lambda = 1e-40$):

	100%	50%	5%
Accuracy	0.9780711606472	0.9744239631336	0.9745007680491
Precision	0.9822735567310	0.986644407345	0.972559916637
Recall	0.9743514042516	0.9739846968805	0.9888169511477
F1	0.9782611534285	0.9802736804691	0.9806210600046

对于拉普拉斯平滑则没有出现因为训练集规模的增加而大幅度改进的情况，这也体现了拉普拉斯平滑性能的稳定性。

issue 2 零概率

零概率事件在如下可能的情况下发生:

- 某些特定的词汇只在某个label中出现，在另一个label中该词出现的概率是0。
- 当测试集中有词汇不属于训练集时，在两个label中的概率都是0。

在朴素贝叶斯公式里，只要有一个项是0，就会导致连乘的概率为零，导致失败预测。平滑处理可以实现简单的平滑处理，也可以实现拉普拉斯平滑。

1. 普通平滑:

   查阅资料可知，在之前的研究中即有相关研究，最合适的参数是：对只出现在一个集合里面的词，赋值概率为0.01，如果两个集合都没有出现相关的词语，则给这个词的概率赋值为0.4

   相比于不实现平滑的模型而言（即对所有零概率的直接跳过该词），普通平滑给零概率的词提供了一个初始概率，使得降低了因为数据规模不够而出现的词汇分布不均匀导致对概率计算带来问题。

   模型一（用普通平滑的词袋模型）即是这样处理的模型。

2. Laplace平滑

$$P(X = x_j|Y = C_k) = \frac{x_j + \lambda}{m_k + n\lambda}$$

   其中λ是拉普拉斯平滑参数。

本实验中λ值取1e-40，可以看到。拉普拉斯平滑的效果要明显优于普通平滑的方法。

以下是对lambda参数的讨论：

	1e-50	1e-40	1e-10	1	10
Accuracy	0.97255294	0.97139020	0.9693579	0.95630232	0.914922480

可以看到，随着λ的增大，拉普拉斯平滑处理的效果会逐渐下降。因此，λ应该取较小的值。

issue 3 其他可以用来识别的特征

- 发件邮箱信息

  由于垃圾邮件经常来自特定邮箱，163邮箱尤为多，所以可以作为一个特征。匹配的正则表达式是 `regex = u'From.*@.*'` ，将邮箱地址也加入词袋模型中，有助于正确率的进一步提升。

- 邮件内容

  翻看垃圾邮件，可以看到比较明显的是垃圾邮件经常有推销电话和网址，但是电话有的是座机有的是手机，并且正常邮件里也有可能留下电话，更容易匹配的是推销网址，于是将正文里的所有网址字段提取出来加入词袋，有助于进一步提升正确率。

加入辅助信息的效果见实验结果分析部分，准确率提到了0.007个百分点，其他几个指标都有较大的提升。

可能的进一步提升

- 没有用到停用词表，用了之后性能应该会进一步上升