机器学习 实验一 Naive Bayes

个人信息

- 姓名：顾逸宏
- 学号：2015011249
- 班级：计52

模型设计

- 使用Naive Bayes模型，设随机变量Y表示是否是垃圾邮件，$Y=1$表示是垃圾邮件，$Y=0$表示不是垃圾邮件。对于Baseline模型，观测值$\mathbf{X}=[X_1,X_2,\cdots,X_n]$是一个$n$维的随机向量，其中$X_1,X_2,\cdots,X_n$相互独立（当$Y$固定时），即

$$P(X_1,\cdots,X_n|Y)=\prod_{i=1}^{n}P(X_i|Y)$$

- 在Baseline的模型中，仅仅考虑以下feature：X_i表示序号为i的token是否在这封信件中出现过：$X_i=1$表示出现过，$X_i=0$表示没出现过。
- 现在数据已有N个观测，即$\{\mathbf{X}^{(k)},Y^{(k)}\}_{k=1}^{N}$。

- 那么最后我们总结一下这个概率模型：

$$Y\sim \text{Bern}(\alpha)$$

$$X_i|Y\sim \text{Bern}(\beta_i)$$

- 我们使用以下的方法估计α和β:

$$\alpha=\frac{\#\{Y=1\}}{N}$$

$$\beta_i=\frac{\#\{Y=y,X_i=x_i\}}{\#\{Y=y\}}$$

Baseline 实现

- 首先，在建模的时候，关于一封信的内容，我们仅仅把它看成是一个n维的0/1向量，而忽略上下文甚至忽略每个词出现的个数。
- 我们把数据集按照8:2分成了training set和test set。
- 由于在初期就遇到了*zero probabilities*的问题，所以我们在Baseline中就实现了smoothing，即

$$P(X_i = x_i|Y = y) = \frac{\#\{X_i = x_i, Y = y\} + \gamma}{\#\{Y = y\} + 2\gamma}$$

- 关于γ的选取我们在之后的部分讨论。

实验结果和分析

主要结果，数据集大小的影响

- [**Issue 1**] 查看数据集的大小(分别保留训练集的5%，50%和100%)对结果的影响，主结果如下表所示，其中每个数据我们均random shuffle了5次，并给出平均值和标准差。

Train Size	Accuracy	Precision	Recall	F1 Score
5%	0.9598 ± 0.0038	0.9620 ± 0.0056	0.9781 ± 0.0026	0.9699 ± 0.0028
50%	0.9819 ± 0.0017	0.9904 ± 0.0015	0.9823 ± 0.0017	0.9863 ± 0.0013
100%	0.9839 ± 0.0019	0.9949 ± 0.0011	0.9808 ± 0.0021	0.9878 ± 0.0014

- 分析：通过以上数据我们可以得到以下结论
  - 随着training set的size变大，各个metric的mean performance都严格上升，尤其是5%至50%这一段上升比较明显。
  - 随着training set的size变大，各个metric的performance的std趋向于变小。
  - F1 score明显好于Accuracy，总体来看，Precision均严格大于Accuracy，Recall总体来说比Accuracy好。
  - 通过以上数据，我们可以发现模型主要犯错在于把Spam分成了Ham。

零概率

- [**Issue 2**] 尝试用smoothing来解决零概率问题
  - 首先，通过数据集我们发现，zero probability非常常见，即有的词只在 spam or ham中出现但是没有在另外一边出现。
  - 如果不去考虑零概率问题，我们可以发现，我们计算的$P(Y = 1, \dot{\mathbf{X}})$和$P(Y = 0, \dot{\mathbf{X}})$都是0（$\dot{\mathbf{X}}$是需要预测的特征向量），无法做出相应的判断。
- 我们通过加入γ解决这个问题。同时，通过实验，我们列出了γ的大小对结果的影响（见下表，只跑了一轮）

γ	Accuracy	Precision	Recall	F1 Score
1	0.9143	0.9642	0.9034	0.9328
1e-1	0.9253	0.9697	0.9151	0.9416
1e-10	0.9704	0.9889	0.9658	0.9772
1e-100	0.9857	0.9958	0.9823	0.9891

- [关于γ对结果的影响、解释和讨论]
    - 可以发现γ对最后的结果影响非常大，其中，accuracy和recall从中收益最高，由于减少γ的值实质上是给那些仅在spam or ham的词了更大的权重，
    - 从结果来看，实际上减少γ使得False Negative的量减少了，即对于那些本来是spam的但是误分成ham的数量减少了。
    - 由此可见，在spam邮件中有一些只有在spam中才出现的词/从来没有的词，让这些词对最后的结果有更大的权重有利于增大预测准确度。

额外的feature

- 我们选取了以下feature
    - `X-Mailer` 的种类 (共158种)，标记为Z，一共有158种取值。
    - 发送时间：包括小时和是星期几，标记为W，一共有7×24种取值，但是我们标记为$W = [W_1, W_2]^T$，W_1表示星期几，W_2表示小时。
- 使用了之后模型具体的提升如下：

Model	Accuracy	Precision	Recall	F1 Score
Z, W	**0.9919**	**0.9960**	**0.9917**	**0.9938**
Z	0.9919	0.9960	0.9917	0.9938
W	0.9857	0.9958	0.9824	0.9891
Baseline	0.9857	0.9958	0.9823	0.9891

- 关于结果的解释和讨论
    - 我们发现，Z能对最后的结果产生比较显著的提升。
    - 同时我们发现，引入W对最后的结果基本没有影响。

讨论和总结

我们总结工作如下：

- 我们使用了Naive Bayes模型来进行垃圾邮件的判断，我们基于bag of words提出了baseline的模型，获得了0.9839的Accuracy和0.9878的F1 Score。
- 我们分析了training size大小对最后结果的影响，发现对最后结果还是有较大的影响的。
- 我们使用了四个不同的评价参数，最后发现我们模型主要的问题是把spam分成了ham。
- 我们使用smooth解决了zero prob问题，在关于参数γ对最后模型的非常显著的影响的讨论中，我们得出了结论：存在一些feature只在spam or ham中出现，加大这些feature的影响力有利于提升模型的预测能力。
- 我们新添了两个额外的feature，并且讨论了这两个feature对最后的模型的影响。