机器学习概论 实验一 垃圾邮件识别 实验报告

- 作者：计 76 王聿中

1. 任务与方法

1.1 实验任务

使用朴素贝叶斯（Naive Bayes）模型完成垃圾邮件识别算法。

1.2 垃圾邮件识别问题的形式化描述

对于一封邮件，我们使用向量 $\boldsymbol{x}$ 来描述其邮件内容，其每一位 $\boldsymbol{x}_i$ 为邮件的一个特征（可能是其中出现的一个单词，也可能是其他特征）。同时，用变量 y 描述其是否为垃圾邮件（$y=1$ 表示是，$y=0$ 则表示不是）。这是一个典型的二分类问题。

垃圾邮件识别模型需要以 $\boldsymbol{x}$ 为输入，输出结果 $\hat{y}$：若 $\hat{y}=1$，则表示算法预测该邮件为垃圾邮件；若 $\hat{y}=0$ 则表示算法预测该邮件为普通邮件。算法需要使预测结果 $\hat{y}$ 尽可能与实际结果 y 相同。

输入向量的设置（Issue#3）

在本文中，输入向量 $\boldsymbol{x}$ 取邮件正文中出现的词，以及发件方域名[1]。

1.4 方法

朴素贝叶斯识别垃圾邮件的方法如下：

我们假设 $P\left(y|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\right)\propto P\left(y\right)\prod_{i=1}^{n}P\left(\boldsymbol{x}_i|y\right)$，其中 n 为向量 $\boldsymbol{x}$ 维数，

则我们可以通过对训练集拟合得到 $P\left(y\right)$ 和 $P\left(\boldsymbol{x}_i|y\right)$，

并对于任意输入 $\boldsymbol{x}$，通过式子 $\hat{y}=\operatorname{argmax}_y P\left(y\right)\prod_{i=1}^{n}P\left(\boldsymbol{x}_i|y\right)$ 进行预测即可。

拉普拉斯平滑（Issue#1）

对于训练集中未出现过的词，其 $P\left(\boldsymbol{x}_i|y\right)$ 为 0，因此若训练集中包含这样的词，将使结果变为 0。

为了避免这样的情况出现，我们引入拉普拉斯平滑：令 $P\left(\boldsymbol{x}_i|y\right)=\frac{\#\{y,x_i\}+\lambda}{\#\{y\}+M\lambda}$ 。其中，λ 为拉普拉斯平滑系数，M 为标签数。

2. 实验

2.1 数据集

使用了课程提供的英文邮件数据集 https://plg.uwaterloo.ca/~gvcormac/treccorpus06/，共 37691 条数据[2]。

2.2 实验设定

在本实验中，我们使用了 k 折交叉验证，即：将数据随机平均分成 k 组，并进行 k 次训练-测试：其中第 i 次使用第 i 组数据进行测试，剩余部分用作训练。最终取 k 次测试结果的平均值作为实验结果。

在本文中，若未特殊说明，k 均取 5。

在本文中，我们将使用 Accuracy、Precision、Recall、F1 四项指标来评价算法的预测。

2.3 实验一：基础实验

k 次训练-测试和最终结果汇总如下表所示（其中，Mean 表示平均，Std 表示标准差[3]）：

	1	2	3	4	5	Mean	Std
Accuracy	0.938	0.941	0.941	0.932	0.938	0.938	0.0031
Precision	0.925	0.926	0.926	0.917	0.924	0.924	0.0033
Recall	0.949	0.951	0.952	0.945	0.949	0.949	0.0025
F1	0.934	0.936	0.936	0.927	0.934	0.933	0.0032

可以看到的是，准确率和 F1 都达到了 93% 以上，表现尚可。

除此之外，数次实验的标准差也较小，初步说明实验随机性不大。

实验二：训练集大小对实验的影响（Issue#2）

为了探究训练集大小对实验结果的影响，我们取 $k=2,3,4,5$，分别进行了 k 折交叉验证。很显然地，由于训练集大小为 $\frac{k-1}{k}$，因此 k 越大训练集也将越大。

实验结果如下[4]：

k	2	3	4	5
Accuracy	0.929744	0.935184	0.935608	0.938075
Precision	0.914866	0.920617	0.921125	0.923733
Recall	0.942576	0.946777	0.947192	0.949166
F1	0.924782	0.930389	0.930852	0.933432

观察上表，不难得出结论：训练集越大，实验结果越好。

为了进一步探究训练集大小对实验的影响，我们在 5 折交叉验证中，只取前 5% 的训练数据进行训练，并取 3 个不同的随机种子，测试结果如下：

Seed	100000007	123456789	998244353	**Mean**	**Std**
Accuracy	0.911491	0.885119	0.867609	0.8881	0.0221
Precision	0.895901	0.871786	0.860577	0.8761	0.0181
Recall	0.92325	0.905577	0.89001	0.9063	0.0166
F1	0.9054	0.879339	0.862222	0.8823	0.0217

作为对比，同样使用此 3 个随机种子，正常训练（使用 100% 训练集数据进行训练）得到的结果如下：

Seed	100000007	123456789	998244353	**Mean**	**Std**
Accuracy	0.938075	0.936776	0.929744	0.9349	0.0045
Precision	0.923733	0.922368	0.914866	0.9203	0.0048
Recall	0.949166	0.948011	0.942576	0.9466	0.0035
F1	0.933432	0.932058	0.924782	0.9301	0.0046

可以看到的是，上表中各项指标的均值更小，但方差更大。这表明：更小的训练集不但会在各项指标上影响模型的表现，还会使模型变得更不稳定。

实验三：拉普拉斯平滑系数 λ 对实验结果的影响

我们取了 $\lambda = 0.01, 0.1, 0.25, 0.5, 1.0$ 分别进行实验，并得到如下结果：

λ	0.01	0.1	0.25	0.5	1.0
Accuracy	**0.968374**	0.959194	0.952721	0.946592	0.936776
Precision	**0.959278**	0.947991	0.940297	0.933206	0.922368
Recall	**0.972868**	0.965592	0.960506	0.95581	0.948011
F1	**0.965416**	0.955609	0.94875	0.942315	0.932058

在上表中，λ 越小，实验结果越出色。但我们并不确定这是否对于所有范围都成立。若需验证这一结论，还需更深入的实验。

3.总结

总的来说，我们不但较好地完成了基于朴素贝叶斯的垃圾邮件识别模型，获得了 96% 以上的 Accuracy 和 F1；并且通过丰富的实验，得到了一些有趣的结论。

我们认为，朴素贝叶斯在这个问题上表现较好是自然的。根据经验，二分类问题是难度非常低的一类问题，对于绝大多数不刁钻的二分类问题，传统的统计机器学习模型可以很容易地达到 95% 以上的 Accuracy 和 F1，而深度学习神经网络模型在这两项指标上达到 98% 以上则是更轻而易举。而若遇到更复杂的多标签分类（Multi-label Classification）任务或是回归（Regression）任务，则统计机器学习模型（如 NaiveBayes、SVM、LGB 等）则更多时候效果不如深度学习算法。

致谢

感谢张老师的精彩授课和精心的作业安排。

感谢助教的指导和批阅。

感谢陈果、邢健开同学（排名不分先后）与我交流给予我的帮助和启发。

附录

代码使用说明

详见 `README.md`

一些细节

- 由于编码原因，我们扔掉了少许数据，因此数据总数是 37691 条。
- get_data.py 非常非常非常的慢，因此你可以使用参数 `--utf8-only` 来让脚本只收集 UTF-8 编码的数据。这将使得脚本速度变快许多，但会额外扔掉 5000 条左右的数据。
- 在数据预处理阶段，我们进行了去停用词的处理，这部分我们使用了 `nltk` 库提供的停用词表（这个包并不是必须安装的）。
- 我们选择了调用 `sklearn` 中的库函数进行指标的计算，尽管实现它们是非常容易的。
- 由于直接计算概率会使得结果由于浮点数精度问题变得极不精确，因此实际实现中，我们对概率取对数后相加来替代使用概率相乘。很显然地，这将不会改变结果的大小关系。
- 在实验中，我们将邮件域名同时加入了输入的特征向量中，为分类器提供了更多的依据。

参数

如无特殊说明，实验中参数默认如下：

- λ：拉普拉斯平滑参数，默认取 1.0
- k：k 折交叉验证参数，默认取 5
- 随机种子：默认取 100000007

---

1. 实验显示加入发件方域名作为输入特征后，整体效果可以提升 1% 左右↩
2. 由于编码原因，我们扔掉了少许数据，因此数据总数是 37691 条。↩
3. 由于小数舍入原因，表中展示的标准差与直接按表中元素计算得到的标准差有所出入↩
4. 从这部分起，我们将不展示 k 折交叉验证中 k 次训练-测试的结果，而只展示它们的平均值作为最终实验结果↩