

数据科学与工程学院

统计方法与机器学习课程实验报告

基于 Naive Bayes 算法的文本分类

[GITHUB.COM/QIUSHISUN](https://github.com/QIUSHISUN)

2021 年 6 月 13 日

摘要

文本分类是是一个机器学习领域经典的话题。朴素贝叶斯 (Naive Bayes) 是一种构建分类器的机器学习算法，朴素指该分类算法假定样本每个特征与其他特征都不相关。朴素贝叶斯分类器的一个优势在于只需要根据少量的训练数据估计出必要的参数，且计算量相较其他机器学习算法而言较小。本实验所采用的数据集为卡内基·梅隆大学 Text Learning Group 的 20newsgroup 数据集，共涉及 20 个网络新闻话题。本实验在 Python3.7(Jupyter-Lab) 环境下进行，并且对一些重要指标与分类精度，以及交叉验证结果（误差对比）其进行了可视化展示。本报告中主要展示可视化分析结果和最优模型性能。

关键词: 文本分类，朴素贝叶斯，向量化，误差曲线

目录

摘要	I
第 1 章 项目概述与数据概览	1
1.1 朴素贝叶斯算法	1
1.2 交叉验证	1
1.3 20 newsgroups 数据集	2
第 2 章 算法	4
2.1 朴素贝叶斯算法流程	4
第 3 章 实验过程与结果	5
3.1 使用朴素贝叶斯算法	5
3.2 交叉验证	5
3.3 使用最优超参数进行朴素贝叶斯文本分类	6
第 4 章 参考资料	8

第 1 章 项目概述与数据概览

本实验的目标在于使用朴素贝叶斯分类器对 20 newsgroups 数据集进行文本分类，并使用交叉验证方法找到最优超参数，再进行后续分析。

1.1 朴素贝叶斯算法

朴素贝叶斯算法是一种基于贝叶斯定理的机器学习算法，它在文本分类任务和垃圾邮件（信息）检测任务

朴素贝叶斯最基本的假设即每个样本特征与其他特征不相关，表示如下

$$P(x | y) = P(x_1, x_2, \dots | y) = P(x_1 | y) P(x_2 | y) \dots = \prod_{i=1}^n P(x_i | y)$$

朴素贝叶斯算法实际上学习到生成数据的机制（上一个实验中所使用的 KNN 算法则不然，它仅仅是记忆数据，而非学习），所以朴素贝叶斯算法属于**生成模型**。

朴素贝叶斯法分类时，对给定的输入 x ，通过学习到的模型计算后验概率分布 $P(Y = c_k | X = x)$ ，将后验概率最大的类作为 x 的类输出。后验概率计算根据贝叶斯定理进行，后验概率如下所示

$$P(Y = c_k | X = x) = \frac{P(X = x | Y = c_k) P(Y = c_k)}{\sum_k P(X = x | Y = c_k) P(Y = c_k)}$$

1.2 交叉验证

交叉验证（Cross Validation）是在机器学习建立模型和辅助模型超参数选择时常用的技巧。交叉验证指的就是重复使用数据，把得到的样本数据按一定的准则进行切分，组合为不同的训练集和测试集，用训练集来训练模型，用测试集来评估模型预测的好坏。在此基础上可以得到多组不同的训练集和测试集，某一轮的训练集中的样本在下一轮可能成为测试集中的样本，这就是所谓的“交叉”。



图 1.1: 训练集，验证集和测试集

交叉验证用在数据量不充足时可以方便的帮助我们进行超参选择。交叉验证的过程中一般随机把数据分成三份，如图 1.1 所示，一份为训练集（Training Set），一份为验证集（Validation Set），最后一份为测试集（Test Set）。用训练集来训练模型，用验证集来评估模型预测的好坏和选择模型及其对应的参数。

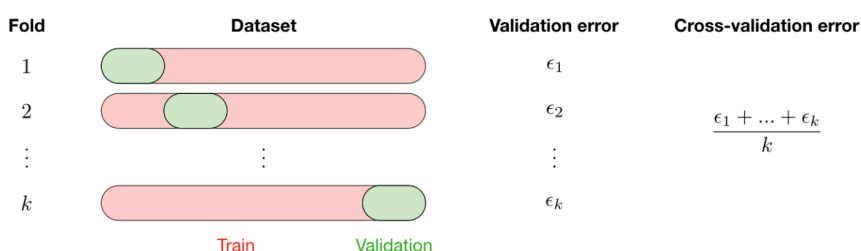


图 1.2: K-Folder 交叉验证

在本次实验中我们使用 5 折交叉验证 (5-Folder Cross Validation)。和第一种方法不同, 5 折交叉验证会把样本数据随机的分成 5 份, 每次随机的选择 $5-1=4$ 份作为训练集, 剩下的 1 份做测试集。当这一轮完成后, 重新随机选择 4 份来训练数据, 最后进行模型评估。

1.3 20 newsgroups 数据集

本次实验所用的数据集为卡内基·梅隆大学 Text Learning Group 的 20newsgroup 数据集, 数据集内包含了 20 个网络新闻话题, 如下所示

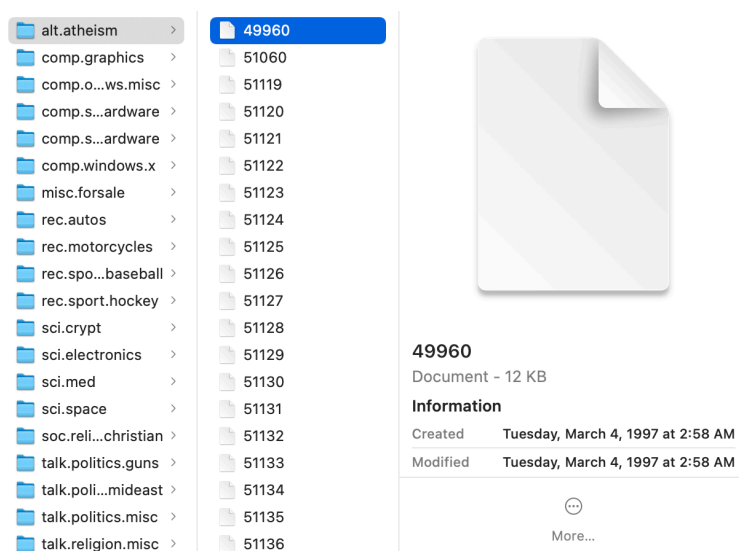
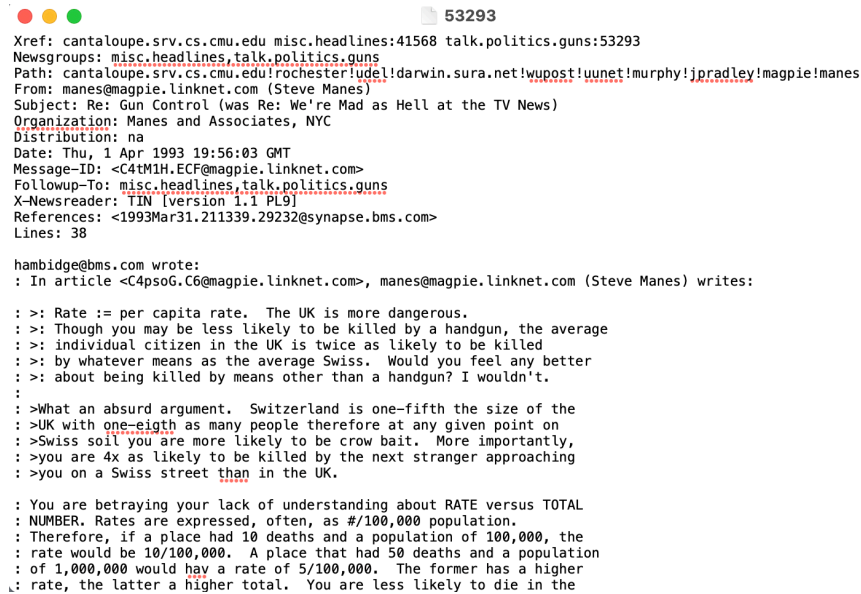


图 1.3: 20 newsgroups

每份新闻数据都包含长短不一的文本内容, 如下所示, 这个示例新闻数据是关于枪支控制的政类新闻。



```
Xref: cantaloupe.srv.cs.cmu.edu misc.headlines:41568 talk.politics.guns:53293
Newsgroups: misc.headlines,talk.politics.guns
Path: cantaloupe.srv.cs.cmu.edu!rochester!udel!darwin.sura.net!wupost!uunet!murphy!jpradley!magpie!manes
From: manes@magpie.linknet.com (Steve Manes)
Subject: Re: Gun Control (was Re: We're Mad as Hell at the TV News)
Organization: Manes and Associates, NYC
Distribution: na
Date: Thu, 1 Apr 1993 19:56:03 GMT
Message-ID: <C4tM1H.ECF@magpie.linknet.com>
Followup-To: misc.headlines,talk.politics.guns
X-Newsreader: TIN [version 1.1 PL9]
References: <1993Mar31.211339.29232@synapse.bms.com>
Lines: 38

hambidge@bms.com wrote:
: In article <C4ps0G.C6@magpie.linknet.com>, manes@magpie.linknet.com (Steve Manes) writes:

: >: Rate := per capita rate. The UK is more dangerous.
: >: Though you may be less likely to be killed by a handgun, the average
: >: individual citizen in the UK is twice as likely to be killed
: >: by whatever means as the average Swiss. Would you feel any better
: >: about being killed by means other than a handgun? I wouldn't.
:
: >What an absurd argument. Switzerland is one-fifth the size of the
: >UK with one-eighth as many people therefore at any given point on
: >Swiss soil you are more likely to be crow bait. More importantly,
: >you are 4x as likely to be killed by the next stranger approaching
: >you on a Swiss street than in the UK.

: You are betraying your lack of understanding about RATE versus TOTAL
: NUMBER. Rates are expressed, often, as #/100,000 population.
: Therefore, if a place had 10 deaths and a population of 100,000, the
: rate would be 10/100,000. A place that had 50 deaths and a population
: of 1,000,000 would hav a rate of 5/100,000. The former has a higher
: rate, the latter a higher total. You are less likely to die in the
```

图 1.4: news content

该数据集可在以下地址获得:

<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html>

第 2 章 算法

2.1 朴素贝叶斯算法流程

首先，我们把独立性假设扩展到条件独立性假设，条件概率分布的参数数量为指数级增长，这在应用中是行不通的，我们需要以下假设：

$$P(X = x | Y = c_k) = P(X^{(1)}, \dots, X^{(n)} | Y = c_k) = \prod_{j=1}^n P(X^{(j)} = x^{(j)} | Y = c_k)$$

算法的核心是最大化后验概率，朴素贝叶斯法将当前的样本分类到后验概率最大的类中，这步等价于期望风险最小化。

$$P(X|Y)_{\text{posterior}} = \frac{\overbrace{P(Y|X)}^{\text{likelihood}} \overbrace{P(X)}^{\text{prior}}}{P(Y)_{\text{evidence}}} = \frac{\overbrace{P(Y|X)}^{\text{likelihood}} \overbrace{P(X)}^{\text{prior}}}{\underbrace{\sum_x P(Y|X)P(X)}_{\text{evidence}}}$$

平滑贝叶斯估计如下，当 $\lambda = 1$ 时，这个平滑方案叫做 Laplace Smoothing。拉普拉斯平滑可视作给未知变量给定了先验概率，也防止了出现分母接近于 0 的情况。

$$P_\lambda(X^{(j)} = a_{jl} | Y = c_k) = \frac{\sum_{i=1}^N I(x_i^j = a_{jl}, y_j = c_k) + \lambda}{\sum_{i=1}^N I(y_j = c_k) + S_j \lambda}$$

值得注意的是，平滑参数 λ 的选取因任务的不同而不同，在本次实验中，我们将使用交叉验证方法确定对文本分类最佳的超参数值。

第 3 章 实验过程与结果

本实验的过程主要包括数据预处理，训练模型、结果分析和数据的可视化展示等几个主要步骤。这部分代码请见 *Project2 – code.ipynb* 文件。

3.1 使用朴素贝叶斯算法

将文本文件解析成词条向量后，直接调用朴素贝叶斯算法进行文本分类，算法流程已在 2.1 中表明。由于代码部分和输出结果较为冗长，请直接参考附带的 *Project2 – code.ipynb/pdf* 文件（我已将 ipynb 转换为 pdf 格式方便查看），在本报告中主要展示可视化分析结果以及最优模型性能。

3.2 交叉验证

通过随机打乱数据集进行交叉验证，辅助选取超参数

在以下超参数空间进行搜索

$\lambda_list = [0.0001, 0.001, 0.01, 0.05, 0.125, 0.25, 0.5, 0.75, 1, 2]$

交叉验证的搜索结果如下所示（数值为精确度）

λ	<i>Fold1</i>	<i>Fold2</i>	<i>Fold3</i>	<i>Fold4</i>	<i>Fold5</i>
$\lambda = 0.0001$	8.92E – 01	8.98E – 01	8.89E – 01	8.97E – 01	8.94E – 01
$\lambda = 0.001$	8.94E – 01	8.99E – 01	8.97E – 01	8.81E – 01	9.01E – 01
$\lambda = 0.01$	8.88E – 01	8.98E – 01	8.92E – 01	8.75E – 01	8.93E – 01
$\lambda = 0.05$	8.84E – 01	8.78E – 01	8.82E – 01	8.92E – 01	8.80E – 01
$\lambda = 0.125$	8.88E – 01	8.83E – 01	8.82E – 01	8.75E – 01	8.77E – 01
$\lambda = 0.25$	8.85E – 01	8.68E – 01	8.78E – 01	8.86E – 01	8.78E – 01
$\lambda = 0.5$	8.77E – 01	8.62E – 01	8.79E – 01	8.65E – 01	8.76E – 01
$\lambda = 0.75$	8.72E – 01	8.72E – 01	8.56E – 01	8.65E – 01	8.66E – 01
$\lambda = 1$	8.46E – 01	8.61E – 01	8.62E – 01	8.53E – 01	8.49E – 01
$\lambda = 2$	8.15E – 01	8.19E – 01	7.99E – 01	8.02E – 01	8.21E – 01

完成上述统计后，计算每组参与交叉验证的数据的均值和方差，使用误差曲线进行可视化，如图 3.1 所示，该曲线反映了不同的超参数选择的情况下，文本分类任务的验证集性能差异。

图中的蓝色曲线为分类准确度均值的曲线，每个超参数选择所对应的短横线为其标准差区间。可以看到，不同超参数对模型的性能有一定的影响，在该参数搜索空间内，验证集精度和性能大致呈负相关。

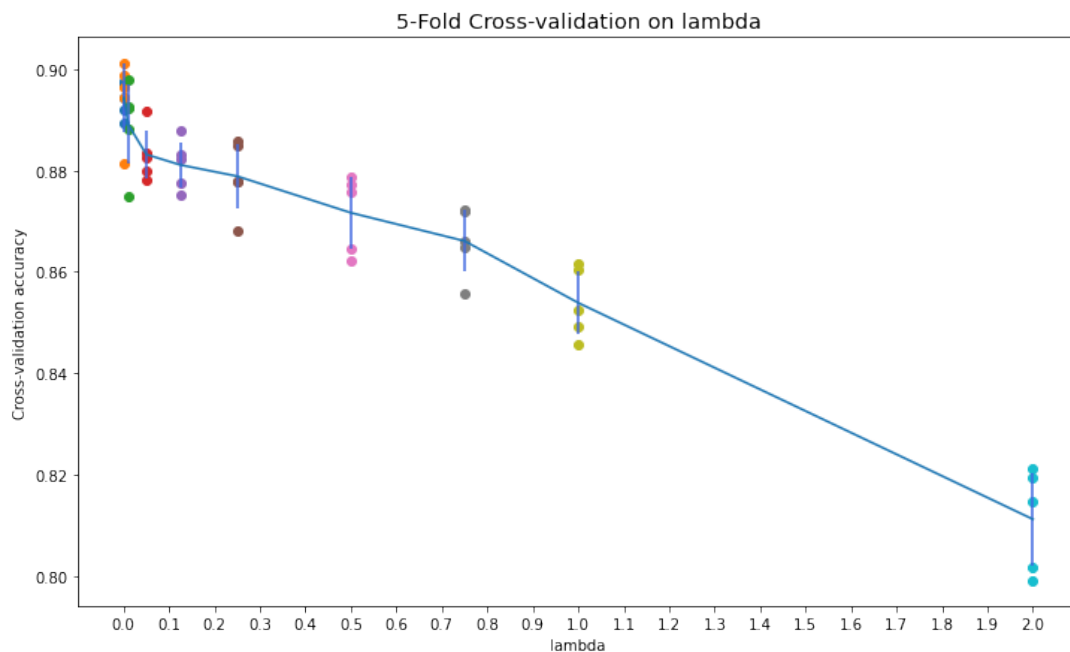


图 3.1: 交叉验证 ErrorBar

随后比较各超参数所对应的组均值，选择 $\lambda = 0.0001$ 为平滑参数，验证集准确率为 0.896，文本分类均值比较如图 3.2 所示。

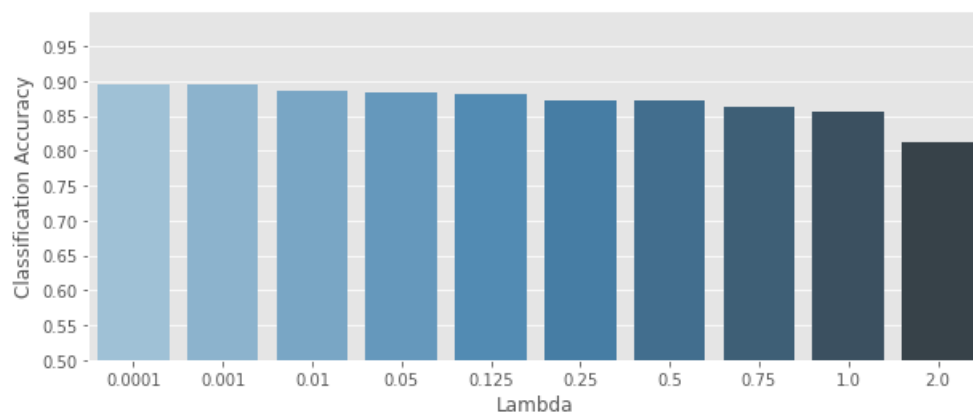


图 3.2: 各超参交叉验证均值比较

3.3 使用最优超参数进行朴素贝叶斯文本分类

我们在最优超参数下进行朴素贝叶斯文本分类实验
分类结果如下所示

class	precision	recall	f1-score	support
<i>alt.atheism</i>	0.90	0.93	0.92	122
<i>comp.graphics</i>	0.73	0.84	0.78	144
<i>comp.os.ms – windows.misc</i>	0.95	0.48	0.64	152
<i>comp.sys.ibm.pc.hardware</i>	0.69	0.90	0.78	158
<i>comp.sys.mac.hardware</i>	0.85	0.85	0.85	144
<i>comp.windows.x</i>	0.85	0.86	0.85	146
<i>misc.forsale</i>	0.86	0.75	0.80	146
<i>rec.autos</i>	0.92	0.92	0.92	153
<i>rec.motorcycles</i>	0.98	0.98	0.98	155
<i>rec.sport.baseball</i>	0.98	0.96	0.97	148
<i>rec.sport.hockey</i>	0.98	0.99	0.98	175
<i>sci.crypt</i>	0.90	0.98	0.94	153
<i>sci.electronics</i>	0.80	0.86	0.83	127
<i>sci.med</i>	0.92	0.93	0.92	135
<i>sci.space</i>	0.94	0.94	0.94	147
<i>soc.religion.christian</i>	0.97	0.93	0.95	164
<i>talk.politics.guns</i>	0.90	0.93	0.91	138
<i>talk.politics.mideast</i>	0.99	0.99	0.99	140
<i>talk.politics.misc</i>	0.88	0.89	0.89	113
<i>talk.religion.misc</i>	0.83	0.78	0.80	67
accuracy			0.89	2827
macro avg	0.89	0.88	0.88	2827
weighted avg	0.89	0.89	0.89	2827

在上表中可见，该超参数选择下，分类的精确度达到 89%，在绝大多数的话题上都有不错的表现。

个人对实验结果的理解：如图 3.1 所示，Laplacian Smoothing 越小，文本分类模型准确率越高，可能的原因是 $\lambda = 0$ 时的极大似然估计是理论上的最优解，Laplacian Smoothing 只是为了排除概率接近或等于 0 的情况下导致计算出错，所以理论情况下，只要使概率不为 0，超参数 λ 取值越小越逼近理论最优值。

第 4 章 参考资料

- (1) 《统计学习方法（第二版）》李航. 清华大学出版社
- (2) [Stanford CS229 Maching Learning: Deep Learning Cheatsheet](#)
- (3) [Stanford CS221 Artificial Intelligence](#)