机器学习实验报告

实验名称：朴素贝叶斯文本分类

学生姓名：蒋雨初

学生学号：58121102

完成日期：2023/6/14

任务描述

用两种方式实现朴素贝叶斯方法：

- 手动实现基于多元伯努利分布的朴素贝叶斯方法（简称：伯努利朴素贝叶斯方法）以及基于多项式分布的朴素贝叶斯方法（简称：多项式朴素贝叶斯方法）；
- 使用 sklearn 库简洁实现多项式朴素贝叶斯方法。

并应用于文本分类任务。

本实验使用的文本分类任务是一个二类分类问题，将文本按情感分类为”积极”和”消极”两类。每条文本为一个 txt 文档，其中训练集共包含 1400 个文档（正类和负类各 700 个），测试集共包含 600 个文档（正类和负类各 300 个）。性能度量方式为准确率。

将文本通过 bag-of-words 方法转化为高维向量（布尔向量或词频向量）进行表示。从数据集中取出部分词语组成词典，词典信息已在代码中给出：['love', 'wonderful', 'best', 'great', 'superb', 'still', 'beautiful', 'bad', 'worst', 'waste', 'boring', 'UNKNOWN']。

实验原理

Bag of words

词袋模型（Bag of Words）是一种文本表示方法，用于将文本转换为数值向量的形式，以便计算机可以处理和分析文本数据。在词袋模型中，文本被看作是一组词汇的无序集合，忽略了单词在文本中的顺序和语法结构，只关注单词的出现频率。

词袋模型的基本思想是，将文本中的每个单词看作一个特征，并为每个单词分配一个索引。然后，通过统计每个单词在文本中的出现次数或出现与否（0 或 1），将文本表示为一个向量。向量的维度等于词汇表中的单词数量，每个维度对应一个单词，向量中的元素表示对应单词在文本中的频率或存在与否。

实现思路十分简单，关键是要获得词根，例如把 loving, loved 都归为 love。为此，可以使用 nltk 的 PorterStemmer。

展示数据集如图 1。

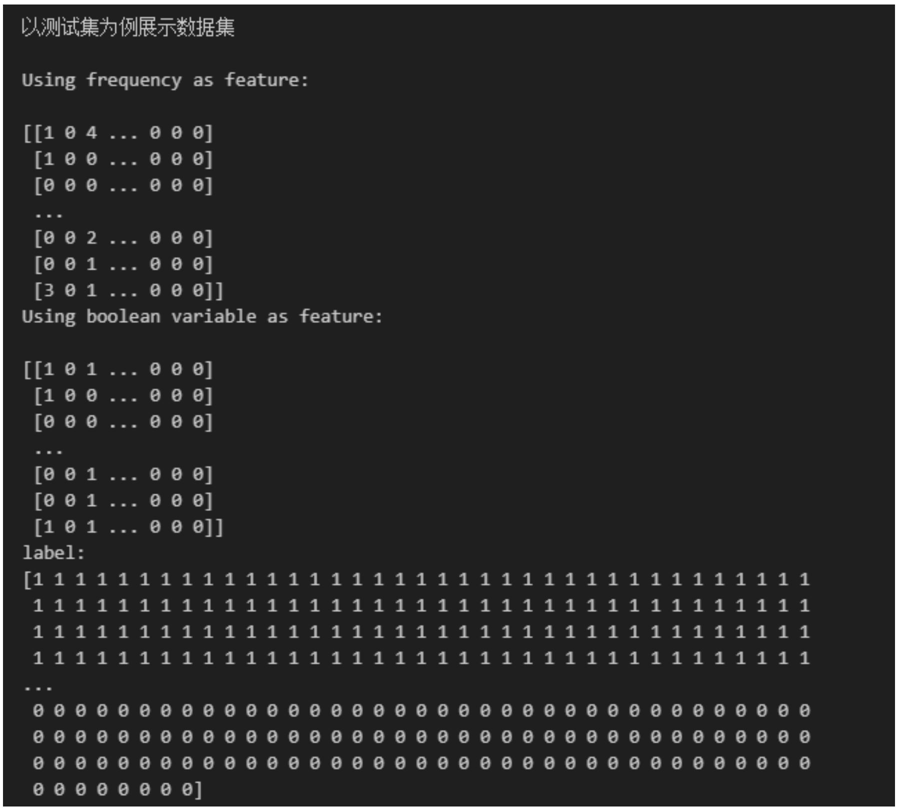

图 1 使用词袋法对数据集进行处理后的结果

朴素贝叶斯分类器

不难发现,基于贝叶斯公式来估计后验概率$P(c \mid x)$的主要困难在于:类条件概率$P(x \mid c)$是所有属性上的联合概率，基于有限训练样本直接估计联合概率，在计算上将会遭遇组合爆炸问题,在数据上将会遭遇样本稀疏问题;属性数越多，问题越严重。

为避开这个障碍,朴素贝叶斯分类器(naive Bayes classifier)采用了“属性条件独立性假设”(attribute conditional independence assumption):对已知类别,假设所有属性相互独立.换言之,假设每个属性独立地对分类结果发生影响。

基于属性条件独立性假设，$P(c \mid x)$可以重写为

$$P(c \mid \boldsymbol{x})=\frac{P(c)P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})}=\frac{P(c)}{P(\boldsymbol{x})}\prod_{i=1}^{d}P(x_i \mid c)$$

其中 d 为属性数目，x_i为在第 i 个属性上的取值。

由于对所有类别来说$P(\boldsymbol{x})$相同,因此基于最小化分类错误率的贝叶斯最优分类器$h^*(\boldsymbol{x})=\arg\max_{c\in\mathcal{Y}} P(c \mid \boldsymbol{x})$的贝叶斯判定准则有

$$h_{nb}(\boldsymbol{x})=\arg\max_{c\in\mathcal{Y}} P(c)\prod_{i=1}^{d}P(x_i \mid c)$$

伯努利朴素贝叶斯

在多元伯努利事件模型中，特征是描述输入的独立布尔值（二元变量）。 与多项式模型一样，该模型在文档分类任务中很受欢迎，其中使用二进制术语出现特征而不是术语频率。

如果x_i 是一个布尔值，表示词汇表中第 i 个词项出现或不出现，然后是文档给定类别的可能性C_k 由下式给出：

$$p(\boldsymbol{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1-p_{ki})^{(1-x_i)}$$

其中 p_{ki} 是类别 C_k生成项 x_i的概率。

多项式朴素贝叶斯

对于多项式事件模型，样本（特征向量）表示多项式 $(p_1,\cdots,p_n)$ 生成某些事件的频率，其中 p_i是事件 i 发生的概率（或多类情况下的 K 个此类多项式）。 特征向量 $X=(x_1,\cdots,x_n)$ 是一个直方图，其中 x_i计算事件 i 在特定实例中被观察到的次数。 这是通常用于文档分类的事件模型，事件表示单个文档中单词的出现（参见词袋假设）。 观察直方图 x 的可能性由下式给出：

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_{i=1}^{n} x_i)!}{\prod_{i=1}^{n} x_i\,!} \prod_{i=1}^{n} p_{ki} x_i \text{ where } p_{ki} := p(x_i \mid C_k)$$

多项式朴素贝叶斯分类器在对数空间中表示时变为线性分类器

$$\begin{aligned} \log p(C_k \mid \mathbf{x}) &\propto \log\left(p(C_k)\prod_{i=1}^{n} p_{ki}^{x_i}\right) \\ &= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki} \\ &= b + \mathbf{w}_k^{\top}\mathbf{x} \end{aligned}$$

如果给定的类别和特征值从未在训练数据中同时出现，则基于频率的概率估计将为零，因为概率估计与特征值的出现次数成正比。 这是有问题的，因为它会在其他概率相乘时消除所有信息。 因此，通常需要在所有概率估计中加入一个称为伪计数的小样本校正，这样就不会将概率设置为恰好为零。这种正则化朴素贝叶斯的方法在伪计数为 1 时称为拉普拉斯平滑，在一般情况下称为 Lidstone 平滑。

拉普拉斯平滑

拉普拉斯平滑（Laplace smoothing），也称为加 1 平滑或加法平滑，是一种用于处理概率统计中零频率问题的技术。它是一种常用的平滑方法，特别在贝叶斯分类和语言模型中被广泛应用。

拉普拉斯平滑主要解决的问题是当我们计算某个事件的概率时，如果该事件在训练数据中没有出现过，根据频率计数的方法会得到概率为零的情况。这种情况下，传统的频率计数方法就会失效，因为概率为零会影响后续的计算结果。

拉普拉斯平滑通过在所有可能事件的计数上加上一个正数（通常是 1），来解决零频率问题。它的基本思想是在计算概率时，将每个事件的计数都加上一个平滑参数，这样可以避免出现零频率问题，并且在一定程度上减小了估计的偏差。

具体而言，对于一个具有 K 个可能取值的离散随机变量，使用拉普拉斯平滑的概率估计计算公式如下：

$$\hat{P}(x_i \mid c) = \frac{\mid D_{c,x_i} \mid +1}{\mid D_c \mid +N_i}$$

其中 $|D_c|$是训练集 D 中第 c 类样本集合的个数，对于离散特征，令$|D_{c,x_i}|$ 表示在D_c中

第 i 个特征上取值x_i的样本集合的个数，另外，N_i表示第 i 个特征可能取值的个数。

实验结果

伯努利朴素贝叶斯 BNBC

如图 2，在数据集上，BNBC 最终取得了 67.00%的准确率。

```
thetaPosTrue = [0.44017094 0.0014245  0.5042735  0.42877493 0.06552707 0.37179487
 0.0014245  0.26495726 0.04558405 0.03988604 0.0014245  0.0014245
 0.0014245 ]
thetaNegTrue = [0.35754986 0.0014245  0.36324786 0.26923077 0.01851852 0.33475783
 0.0014245  0.5042735  0.19230769 0.18233618 0.0014245  0.0014245
 0.0014245 ]
--------------------
BNBC classification accuracy = 0.67
```

图 2 MNBC 的分类结果

多项式朴素贝叶斯 MNBC

如图 3，在数据集上，BNBC 最终取得了 66.83%的准确率。同时，从手动实现的 MNBC 和 sklearn 的 MNBC 的对比当中可以发现，自己手动实现的 MNBC 正确。

```
thetaPos = [0.23350254 0.00039047 0.23428348 0.22100742 0.02030457 0.15657946
 0.00039047 0.10542757 0.01327606 0.01366654 0.00039047 0.00039047
 0.00039047]
thetaNeg = [0.16808424 0.00040502 0.14661806 0.10692588 0.00526529 0.1381126
 0.00040502 0.29040097 0.07290401 0.06966383 0.00040502 0.00040502
 0.00040502]
--------------------
MNBC classification accuracy = 0.6683333333333333
--------------------
Sklearn MultinomialNB accuracy = 0.6683333333333333
```

图 3 手动实现的 MNBC 和 sklearn 的 MNBC 的分类结果