自然语言处理NLP作业②

文本分类Text classification

姓名：罗福杰

学号：3120305208

班级：S0078

目录

[1 作业要求 3](#_Toc53321206)

[2 文本分类的基本概念 3](#_Toc53321207)

[2.1 文本分类的定义 3](#_Toc53321208)

[2.2 进行文本分类的步骤[1] 3](#_Toc53321209)

[2.2.1 文本预处理 4](#_Toc53321210)

[2.2.2 词向量与文本表示 4](#_Toc53321211)

[2.2.3 特征降维 5](#_Toc53321212)

[2.2.4 常见的文本分类算法[2] 6](#_Toc53321213)

[3 用20\_newsgroups数据集进行文本分类 6](#_Toc53321214)

[3.1 朴素贝叶斯算法及其实验结果 6](#_Toc53321215)

[3.1.1 朴素贝叶斯算法的原理 6](#_Toc53321216)

[3.1.2 朴素贝叶斯算法的实验过程和结果和分析 7](#_Toc53321217)

[3.2 决策树算法及其实验结果 8](#_Toc53321218)

[3.2.1 决策树算法的思想 8](#_Toc53321219)

[3.2.2 决策树算法的结果和分析 8](#_Toc53321220)

[3.3 K近邻算法及其实验结果 9](#_Toc53321221)

[3.3.1 K近邻算法原理 9](#_Toc53321222)

[3.3.2 K近邻算法的结果和分析 10](#_Toc53321223)

[3.4 三种分类方法的结论 10](#_Toc53321224)

[参考资料 11](#_Toc53321225)

# 1 作业要求

Text classification:

* This data set contains 1000 text articles posted to each of 20 online newgroups, for a total of 20,000 articles. For documentation and download, see http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html.
* The "label" of each article is which of the 20 newsgroups it belongs to. The newsgroups (labels) are hierarchically organized (e.g., "sports", "hockey").
* You should provide model evaluation results and discuss the reasons of the results.

# 2 文本分类的基本概念

## 2.1 文本分类的定义

文本分类是指按照一定的分类体系或标准使用机器对文本集进行自动分类标记的过程。从宏观上看，整个分类流程可以近似地看作数学上做映射的过程。因此，我们可用映射关系诠释文本分类的概念。

文本分类的数学定义如下:

假设给定文档集合，类别集合。其中，，分别表示集合中第i篇文档和第j个类别; m，n为集合D的文档总数和集合C的类别数。我们可以发现文档集合和类别集合之间存在一定的映射关系。当，时，表示文档 属于类; 反之，当，时，文档不属于类S，为分类器。

## 2.2 进行文本分类的步骤[1]

文本分类从流程上可分为文本预处理、文本表示、特征降维、选择分类器并进行训练等过程，如图2.1所示是文本分类的一般流程。

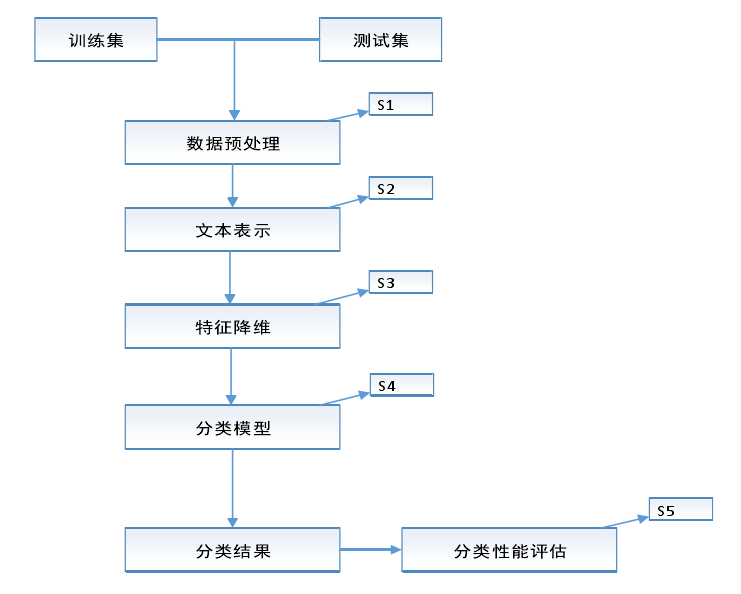


图2.1 文本分类的一般流程

### 2.2.1 文本预处理

在处理文本数据时，首先要对原始信息进行预处理。由于中文数据词语之间没有明确的分隔符且存在一定的噪音信息，所以在预处理阶段通常要经过分词、去除停用词、低频词过滤等过程。现有的分词算法可分为三大类: 分别为基于字符串匹配( 词典) 的分词方法、基于理解的分词方法和基于统计的分词方法。由于英文的语言学特征，在对英文文本进行预处理时，通常还包括词形还原、词干提取等步骤，数据预处理的质量直接影响后续的相关工作。

### 2.2.2 词向量与文本表示

文本是由词和短语构成的符号序列。要将自然语言处理问题转化成机器可学习的数学模型，首先要对词和文本进行向量化建模。

One-hot表达方式是常用的词向量表示方法。假设 N 为整个词表空间，则每个词的词向量可表示为:x = { 0，0，…，1，0…}，该词在词表中对应位置编号的维度为 1，其它的维度为0。

文本表示模型一般包括：Boolean 模型，VSM 模型和统计语言模型。近年来，随着机器学习技术不断向深层网络发展，词嵌入方法也开始得到了一些应用。下面是这几种方法的基本思想。

* **Boolean 模型**实际上是向量空间模型中的一种特殊形式，该模型的基本思想是通过查询词项在文本中是否出现作为依据再进行词项的权重系数判断的，如果当这个文本中出现了这个词项，那么就将这个词项的权重记为 1，否则记 0。
* **VSM 模型**（Vector Space Model 向量空间模型）其实是指能够代表文本对象的一类数学代数模型，它也是一种传统的文本数值化表示方法，它的基本思想是通过将文本中带有文本主题特征的那些词项表示成特征向量，那么每一个具有独立属性的词项所对应的一个特征向量就构成了这篇文本特征向量其中的一个维度，最后就将该文本转化成了特征空间中的一个高维特征向量。
* **词嵌入**的概念是指将单词或者文档进行数字化的一种表示形式，可以简单理解为单词或文档在算法中的储存形式。从一般情况讲，该表示方法会将一个单词通过映射的方式转化到一个高维的向量中（词向量）来代表，每个单词都会被映射成一个向量。常用的方法有Word2Vec，GloVe等算法。

### 2.2.3 特征降维

特征选取是对文本数据进行的一种常用的特征降维方式。特征选取的一般处理流程是：根据文本数据集特点，通过选定流程选取适合的特征计算函数，对数据集中每一条文本中的每个词项分别进行特征计算得到量化结果，将结果按照由大到小进行顺序排列，根据提前设定的阈值情况，从中选出一定数量的特征项作为原始文本数据的代表，期间不涉及特征空间的转化问题。

特征选取算法的好坏会直接影响到文本分类的效果，其核心是选取和采用适合数据集特征的计算函数进行计算。根据计算函数中公式的不同，可分为不同的特征选取算法。常用的特征选取方法有文档频率统计、信息增益、互信息、卡方统计等算法。

通过以上的文本特征选取算法可以得到一部分特征关键词，但为了实现更好的分类结果，还需要结合特征权值计算方法进行特征处理。特征关键词的权重值一般是指对各个特征关键词所具有的文本表示能力的一种衡量结果，它的大小也能够反映出不同的文本。目前最常用的特征加权计算方法是 TF-IDF 、TFC、LTC 等算法。以上特征加权算法可以根据类别特征关键词在数据集文本中分布情况的不同，计算得出不同的特征权重值，并实现更好的文本分类效果。

### 2.2.4 常见的文本分类算法[2]

目前的分类算法比较多，常见的用于文本分类上的算法主要有朴素贝叶斯算法、决策树算法、K近邻算法、支持向量机和神经网络等算法。其中，朴素贝叶斯算法、决策树算法、K近邻算法会在下面结合20newsgroups文本分类数据集中给出原理和实现的具体过程以及实验结果。

# 3 用20\_newsgroups数据集进行文本分类

## 3.1 朴素贝叶斯算法及其实验结果

### 3.1.1 朴素贝叶斯算法的原理

**朴素贝叶斯算法**是经典的文本分类算法，贝叶斯分类是指对一类文本分类算法的统称，这类分类算法借助贝叶斯定理，通过对联合概率建模以获取目标的概率值，它的核心是以贝叶斯理论为基础。

贝叶斯定理是这样规定的：假设某样本空间中存在可划分的文本集合分别为：，其中，表示样本空间中第i个文本集合，且，现在假设有一任意的随机事件B，则可根据下式计算概率。

这个概率值实际是指随机事件 B 发生的条件下样本空间中第 i 个文本集合发生的概率。这是一个先验概率，贝叶斯分类的基本思想就是可通过利用先验概率以及其他概率值进行计算获取后验概率值。简单总结，贝叶斯定理就是通过观察两件随机事件中其中一个随机时间的概率，来计算另一个随机事件概率的定律。可用下式进行解释说明：

而在实际应用中，最常见的、最简单的一种贝叶斯分类是朴素贝叶斯算法即NB 算法，它的区别在于朴素贝叶斯算法中假定了众多特征之间存在互相独立的关系，所以可分别进行独立计算。其计算公式一般如下：

在上式中，结果的值可用其它三个概率值进行计算得到。由于朴素贝叶斯分类算法存在特征互相独立的条件，所以用于文本分类时，可利用特征之间互相独立，在计算中对计算公式进行计算简化。

朴素贝叶斯分类算法对条件概率分布做了条件独立性假设，简化了计算量，属于非常经典的算法。该算法比较简单且易被理解，在文本分类问题以及处理文本的多分类情况时都具备相当稳定的分类效率和较好的分类效果。但它的缺陷是：由于做了条件独立性假设，需要计算先验概率，在处理文本特征间存在关联关系时，不适用。

### 3.1.2 朴素贝叶斯算法的实验过程和结果和分析

1. **导入数据**：首先从官网上下载此次文本分类所需的数据20\_newsgroups，共20个类别，将数据进行导入和读取。赋值给news的变量。
2. **划分训练集和测试集**：使用sklearn中的train\_test\_split函数，将数据集按照一定的比例切分为训练集和测试集，本实验设置训练集占总样本数的0.8，剩下的0.2的比例为测试集数据。
3. **提取特征向量**：**将文本转为TF-IDF向量**。通过feature\_extraction.text中CountVectorizer类和TfidfVectorizer类实现文本中特征向量的提取。CountVectorizer：只考虑词汇在文本中出现的频率，而TfidfVectorizer除了考量某词汇在文本出现的频率，还关注包含这个词汇的所有文本的数量，这能够削减高频没有意义的词汇出现带来的影响, 挖掘更有意义的特征。
4. **分为四组实验并进行对比**：使用CountVectorizer还是使用TfidfVectorizer提取文本信息，是否去除停用词。由于文本中会出现很多停用词，且这些词往往携带的信息较少，所以这次实验里对是否去除停用词做了对比。
5. **用朴素贝叶斯分类算法实现训练和预测**。这部分通过两个函数主要分为训练学习和预测两个阶段。
6. **实验结果**：

**表3.1 使用朴素贝叶斯方法的实验结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 提取特征 | 是否去除停用词 | Precision | Recall |
| CountVectorizer | 否 | 0.85 | 0.87 |
| 是 | 0.87 | 0.89 |
| TfidVectorizer | 否 | 0.84 | 0.87 |
| 是 | 0.87 | 0.89 |

1. **结果分析：**由以上的实验结果可知，去除提用词之后的准确率、召回率以及F1-score都普遍提高了一些。对于使用的哪种特征提取器CountVectorizer还是TfidVectorizer的差别并不是很大。

## 3.2 决策树算法及其实验结果

### 3.2.1 决策树算法的思想

**决策树**全称是分类和回归树，也可以处理分类问题，是一种典型分类算法。决策树类似于树形图，是利用一系列的分类规则对样本点进行一步判断和实现分类的过程。它也是数据挖掘中最常用的使用算法。决策树算法具有方法简单，分类过程中计算量小，并且易生成和易理解的优点。但它的缺陷是当需要分类的类别数量太多时，就不容易处理导致分类结果不稳定。

### 3.2.2 决策树算法的结果和分析

对于原始数据的处理过程与用朴素贝叶斯的方法一样，在设计好实验之后，采用决策树的分类算法进行分类计算，通过使用sklearn中的决策树方法来实现。

得到的实验结果如下：

**表3.2 使用决策树分类算法得到的实验结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 提取特征 | 是否去除停用词 | Precision | Recall |
| CountVectorizer | 否 | 0.64 | 0.64 |
| 是 | 0.67 | 0.67 |
| TfidVectorizer | 否 | 0.62 | 0.62 |
| 是 | 0.62 | 0.65 |

结果分析：由以上的实验结果可以看出，

* 使用CountVectorizer特征提取器的准确率和召回率要高于使用TfidVectorizer特征提取器的结果；
* 去除提用词后的准确率和召回率也会提高；
* 使用决策树分类算法的准确率和召回率要低于使用朴素贝叶斯的算法。**主要的原因是当使用决策树算法时，当需要分类的类别数量太多时，就不容易处理导致分类结果不稳定，从而导致准确率的下降**。

## 3.3 K近邻算法及其实验结果

### 3.3.1 K近邻算法原理

**K近邻算法**是比较容易理解的算法。在处理文本分类问题时，该算法的核心思想可以描述为：对于待分类文本，利用公式计算得到距离该文本最近邻的那 K 个已分类的文本，根据这 K 个已分类文本的类别以及类别比例来替该文本做出类别选择。最近邻标准，在计算中一般采用欧氏距离和 Minkowski距离进行计算。

假设存在两点分别为点和，可采用欧式距离公式进行计算如下式：

利用上式可计算得到两点间的欧氏距离，该公式同样使用于文本分类计算。K 近邻算法思想简单容易理解且易实现，在文本分类中效果明显。它的缺陷是：由于需对每个待分类文本都要进行欧式距离公式计算，在处理大规模数据库进行分类问题时面临计算量大且复杂的问题；同时，当样本分类不均衡时，容易出现分类错误的可能。

### 3.3.2 K近邻算法的结果和分析

对于原始数据的处理过程与用朴素贝叶斯的方法一样，在设计好实验之后，采用决策树的分类算法进行分类计算，通过使用sklearn中的K近邻方法来实现。

得到的实验结果如下：

**表3.2 使用K近邻分类算法得到的实验结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 提取特征 | 是否去除停用词 | Precision | Recall |
| CountVectorizer | 否 | 0.52 | 0.58 |
| 是 | 0.53 | 0.67 |
| TfidVectorizer | 否 | 0.79 | 0.80 |
| 是 | 0.80 | 0.80 |

结果分析：由以上的实验结果可以看出，

* 使用TfidVectorizer特征提取器的准确率和召回率要明显高于使用CountVectorizer特征提取器的结果；
* 去除提用词后的准确率和召回率会略微提高，不是特别明显；
* 使用K近邻分类算法的准确率和召回率要低于使用朴素贝叶斯的算法。且该方法的准确率与特征提取的方式有明显的关系。

## 3.4 三种分类方法的结论

综合以上三种分类器的结果可以看出，不同分类器在该数据集上的表现有着较大的差别，且在参数设置上，不同的参数也会带来较大的差别。此外，还可以尝试其他的分类算法，在实际应用中，应根据不同的数据集适当采用合适的分类方法才能达到比较满意的效果。

# 参考资料

1. 黄勇，罗文辉，张瑞舒.改进朴素贝叶斯算法在文本分类中的应用[J].科技创新与应用，2019（5）：24，27.
2. 王行甫，付欢欢，王琳.基于余弦相似度和实例加权改进的贝叶斯算法[J].计算机系统应用，2016（8）：166-170.
3. 王爽. 基于机器学习的自动文本分类方法研究[D].电子科技大学，2020.6.
4. <https://blog.csdn.net/Tong_T/article/details/89736810?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase>
5. <https://blog.csdn.net/sinat_28576553/article/details/80258619>
6. <https://www.jianshu.com/p/244180c064cf>
7. <https://blog.csdn.net/a_step_further/article/details/81259506>