DataMining

VSM and KNN

姓名：董潇
学号：201834861
班级：2018 级计算机专硕班
指导老师：尹建华
时间：2018 年 11 月 5 日

一、 实验要求

预处理文本数据集，并且得到每个文本的 VSM 表示。
实现 KNN 分类器，测试其在 20Newsgroups 上的效果。
20%作为测试数据集，保证测试数据中各个类的文档分布均匀。

二、 实验内容

（1） VSM

构建向量空间一个传统的方法就是:

首先利用分词工具对文本进行分词，然后通过去 stopwords、标点、转化小写等操作创建词典。

最后再通过计算每个单词的 tf 以及 idf 得到每个单词的权重，从而得到每个文件的 VSM.

（2） KNN

利用已划分好的数据集:

20news-bydate-test

20news-bydate-train

通过计算两个向量之间的距离，得到训练集与测试集每个文件的相似度，并按由大到小排列，然后设置 K 值，得到前 K 个文件中哪个类型最多，就把测试文件分到哪一类。

K 值不同，正确率也不同:

三、 实验结果

```
In [9]: runfile('D:/DataMining/HomeWork1/KNN_News/main.py', wdir='D:/DataMining/HomeWork1/KNN_News')
加载训练数据 ...
向量化数据集 ...
计算KNN ...
k = 30, 正确率: 0.702337.
k = 31, 正确率: 0.702735.
k = 32, 正确率: 0.704727.
k = 33, 正确率: 0.704594.
k = 34, 正确率: 0.706054.
k = 35, 正确率: 0.705523.
k = 36, 正确率: 0.707780.
k = 37, 正确率: 0.706585.
k = 38, 正确率: 0.709241.
k = 39, 正确率: 0.707249.
k = 40, 正确率: 0.707913.
k = 41, 正确率: 0.705921.
k = 42, 正确率: 0.706320.
k = 43, 正确率: 0.707913.
k = 44, 正确率: 0.708975.
k = 45, 正确率: 0.709506.
k = 46, 正确率: 0.710967.
k = 47, 正确率: 0.711498.
k = 48, 正确率: 0.710170.
k = 49, 正确率: 0.709904.
k = 50, 正确率: 0.707647.
```

实验结果截图

四、 实验总结

（1）这次实验中遇到了很多的问题，首先是语言问题，以前接触过一点 python，对它掌握的还不是很熟练，所以先自学了 python。

（2）其次就是 KNN 编程问题，因为开始并不是很了解他的原理以及细节，通过询问同学以及查找资料，一步一步细化，最终得以理解，可见编程是建立在理解的基础之上的。

DataMining

朴素贝叶斯分类器

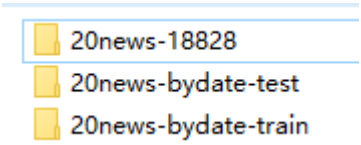
姓名：董潇
学号：201834861
班级：2018 级计算机专硕班
指导老师：尹建华
时间：2018 年 11 月 18 日

一、 实验要求

实现朴素贝叶斯分类器，测试其在 20 Newsgroups 数据集上的效果。

二、 实验内容

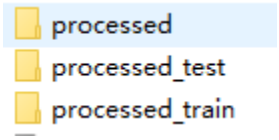
1. 准备好实验所用数据集：



数据集目录

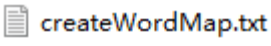
上图中，20news-18828 为总的数据集，20news-bydate-test 与 20news-bydate-train 为以 2:8 分好的测试集以及训练集。

2. 借助 NLTK 的 Python 库对文件进行预处理，包括通过对文本数据分句、分词、去 stopwords、去非字母字符、转化小写等。
在本实验中，借助 processing()函数对上述三个数据集分别进行了预处理，分别生成测试、训练、以及为建立词典做准备的预处理文件即 processed_test、processed_train、processed

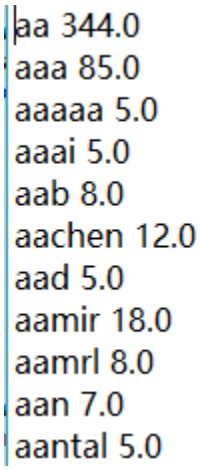


预处理文件目录

3. 创建词典
借助 createWordMap()函数，对预处理生成的为建立词典做准备 processed 文件进行过滤词频处理，生成词典 createWordMap.txt

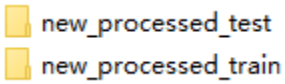


词典文件



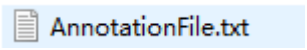
词典小部分内容截图

4. 借助 tokenWords()函数,对训练和测试预处理文件按照词典选取 token，分别得到 new_processed_test、new_processed_train



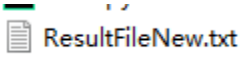
Token 文件截图

5. 创建测试文件的分类标注文件 AnnotationFile.txt，为计算准确率做准备。
   标注：序号 所属类



标注文件

6. 借助 Bayes(traindir,testdir,ResultFileNew)函数,用贝叶斯算法对测试文档分类得到分类结果文件 ResultFileNew.txt



分类结果文件

其中，分别计算条件概率、先验概率：

条件概率 =（类中单词 i 的数目+0.0001）/（类中单词总数+训练样本中所有类单词总数）

先验概率 =（类中单词总数）/（训练样本中所有类单词总数）

最后返回该测试样本在该类别的概率。

7. 将测试集的真实类别与算法册数的类别相比较，计算准确率。

三、 实验结果

```
In [31]: runfile('D:/DataMining/HomeWork2/main.py', wdir='D:/DataMining/HomeWork2')
trainTotalNum:
2065381
accuracy: 0.762917
```

实验结果截图

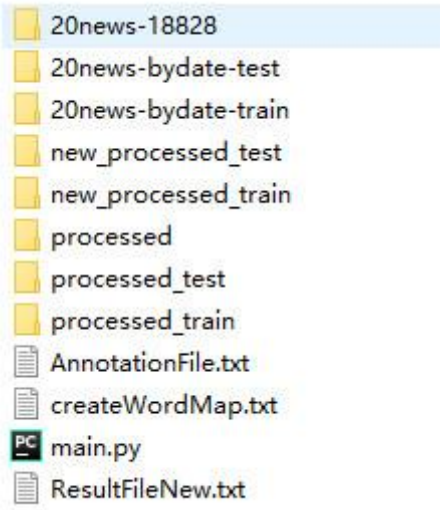
四、 实验总结

因为有了实验一的基础，此次实验的进行顺利了许多，实验一直接借助 sklearn 库函数进行的预处理，本次实验的预处理尝试手写，在此过程中也收获颇多。

此次实验中需要注意的就是数据规模大，在计算概率时，数据相乘容易溢出的问题。通过询问同学，发现可以借助 log 来避免。

五、 附录

总的文件目录截图：



文件目录截图

DataMining

聚类方法对比

姓名：董潇
学号：201834861
班级：2018 级计算机专硕班
指导老师：尹建华
时间：2018 年 12 月 26 日

五、 实验要求

用 scikit-learn 中的各种聚类方法对 Tweets 数据集进行聚类。

六、 实验内容

1. 准备好实验所用数据集：Tweets.txt
2. 对数据集进行向量化处理，方便作为输入传入要调用的函数中。
3. 调用所需的聚类函数，需要注意的是参数的设置。
4. 输出各个聚类函数的正确率进行比较。

七、 实验结果

```
In [8]: runfile('D:/DataMining/HomeWork3/cluster.py', wdir='D:/DataMining/HomeWork3')
K-means的准确率: 0.7910405358188654
AffinityPropagation算法的准确率: 0.7842866176251021
meanshift算法的准确率: 0.7455412796815585
D:\Anaconda3\lib\site-packages\sklearn\manifold\spectral_embedding_.py:234: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
  warnings.warn("Graph is not fully connected, spectral embedding"
SpectralClustering算法的准确率: 0.6277834214648518
DBSCAN算法的准确率: 0.7073939018290382
AgglomerativeClustering算法的准确率: 0.783967749879778
GaussianMixture算法的准确率: 0.779042358948001
```

实验结果截图

八、 实验总结

通过本次实验掌握了 sklearn 库里面这 7 种聚类方法的调用与使用，收获很多。在实验过程中遇到的主要问题就是，传入参数的问题，不过最终通过调查资料得以解决，收获颇多。