

# kNN实验报告-许晓康

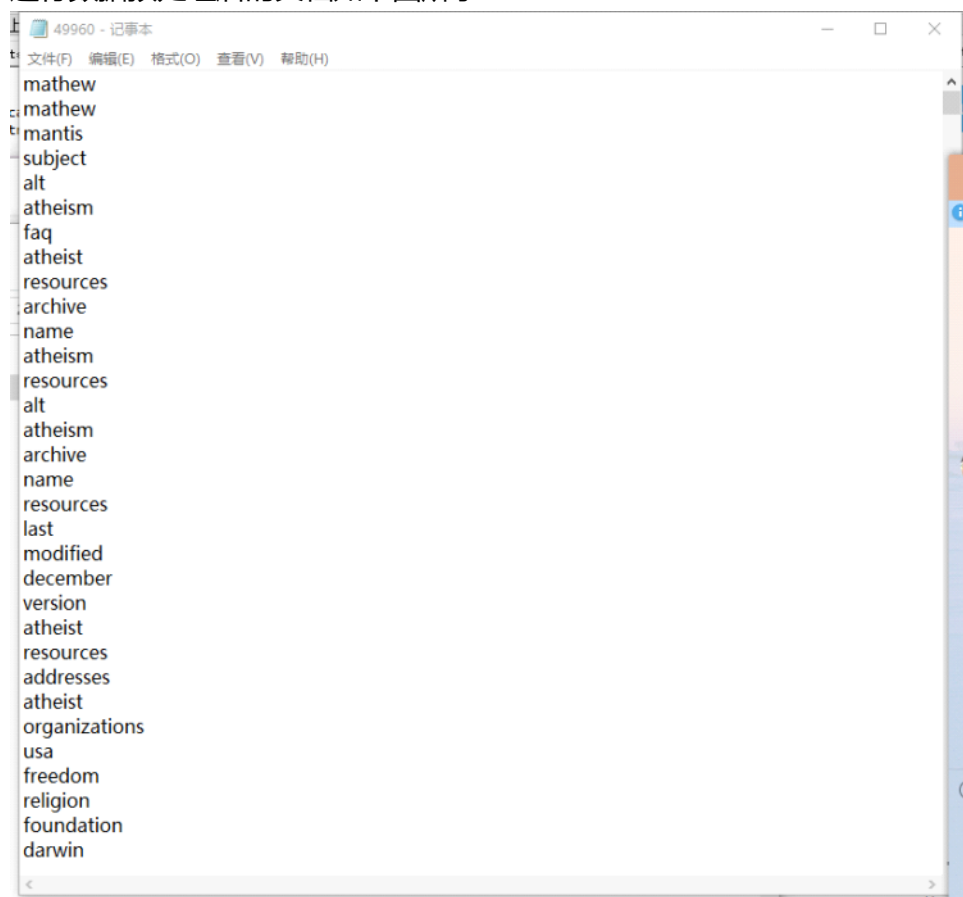
2018年10月15日 星期一 下午6:32

第一步，进行数据预处理 dataPreprocessing.py

预处理工具使用的是nltk，全称natural language toolkit 是一个基于python的自然语言处理工具集。

- 1) 使用tokenize(content)方法对句子进行分词
- 2) 去除停用词（停用词使用nltk中的英文停用词）
- 3) 去除单个字母与数字
- 4) 去掉长度小于2的单词

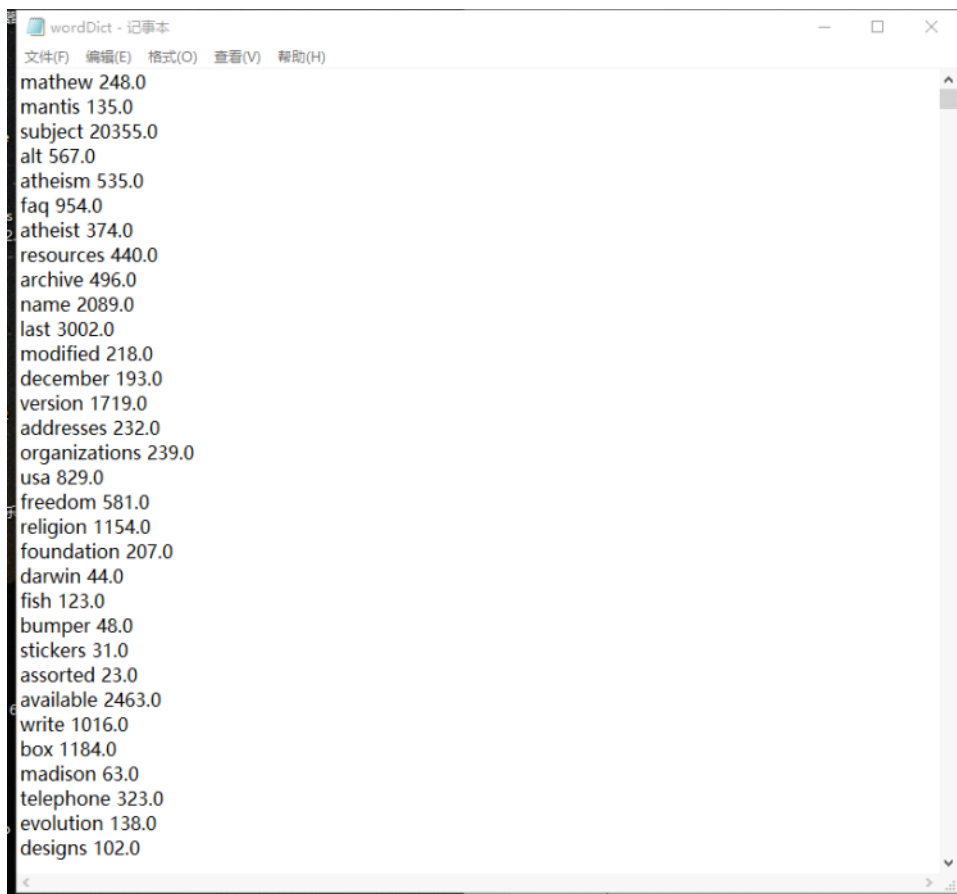
进行数据预处理后的文档如下图所示



第二步，建立字典，过滤文件，划分数据集 partitionData.py

- 1) 方法build\_word\_map():

首先对所有文件进行遍历，建立字典，遍历同时统计每个单词出现的频率，设定阈值，将小于13的单词排除，最初设定阈值为5时，字典size大概接近4万，后来调整到13，字典size为1.8万（18725），最后将字典wordDict保存到wordDict.txt文件中，如下图所示



2) 方法filter\_file():

对所有文件进行过滤，如果在字典中出现的单词则保留，否则丢弃

3) 方法partition\_data():

划分数据集，按照80%训练集 20%测试集的标准，按照文件列表的顺序划分为训练集和测试集

第三步，计算idf, tf\*idf computeTFIDF.py

1) 方法computeIDF():

读取所有的文件，在读取过程中记录文档总数目，并且为每个单词建立一个集合set()，统计出现该单词的文档。然后计算每个单词的idf值 $idf = \log(\text{docNum}/(\text{wordDocNum}+1))/\log(10)$ 并保存到文件中

2) 方法computeTFIDF():

读取训练集和测试集的文档，在每个文档中，计算每个单词的tf值，与idf值相乘作为单词的权重保存到文件中。所有的训练集保存到一个文本文档中，所有的测试集保存到一个文本文档中，文档格式都相同，其中每一行代表一个文档的数据，格式如下[类名 文档名 单词1 权重 单词2 权重...]，文档之间以换行符区分。

训练集生成的TFIDF文档

trainTfidf - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

alt.atheism 49960 mathew 0.006768 mantis 0.006872 subject 0.000084 alt 0.005101 atheism 0.020949 f  
ailing 0.001836 figmo 0.003676 netcom 0.001350 com 0.000399 net 0.002282 directly 0.001525 price 0.0  
lack 0.002909 secular 0.007130 uncovering 0.003320 history 0.008366 freethought 0.007111 quarterly C  
ionsloesn 0.003676 hrsg 0.003676 vertrieb 0.003676 ibdk 0.003676 ucherdienst 0.003676 hannover 0.0  
02010 atoms 0.002557 philip 0.004877 dick 0.006356 wrote 0.003266 many 0.002360 philosophical 0.00  
.003462 schizophrenic 0.003462 hero 0.002609 searches 0.002577 hidden 0.002136 mysteries 0.002708  
0.002158 live 0.001300 christian 0.005297 theocracy 0.003555 property 0.001917 revoked 0.003060 bank  
130 etc 0.000966 german 0.001953 translation 0.001987 gottes 0.003676 erste 0.003462 diener 0.00367  
ly 0.002266 demonstrates 0.002502 unsupportable 0.003676 incoherent 0.006429 turner 0.002346 with  
68 opinions 0.001186 popular 0.001780 observations 0.002140 traces 0.002644 expressed 0.001728 twis  
01722 hume 0.003462 moral 0.001686 newman 0.002802 kant 0.003320 sidgwick 0.003676 recent 0.00  
450 small 0.001249 mail 0.002740 server 0.003273 carries 0.002270 archives 0.002228 moderated 0.002  
alt.atheism 51288 mayne 0.084001 pipe 0.023200 fsu 0.021872 edu 0.006975 william 0.016501 subject (C  
.017198 survived 0.023541 spite 0.024073 leadership 0.021383 former 0.016152 given 0.011340 seeing (C  
alt.atheism 51290 bil 0.003915 okcforum 0.003935 osrhe 0.003925 edu 0.000402 bill 0.004347 conner 0  
19 crap 0.020428 type 0.002201 phobe 0.006135 comes 0.004248 phobia 0.010894 irrational 0.011776 f  
77 population 0.005529 growth 0.007628 evolution 0.015091 occurs 0.003350 members 0.005164 gene  
kind 0.005163 part 0.001723 aberration 0.012270 contrary 0.006487 nature 0.005340 well 0.003531 arg  
7 soon 0.002420 able 0.001841 choose 0.005580 lifestyle 0.004336 decision 0.002943 akin 0.004648 unr  
t 0.003182 realize 0.002744 course 0.001726 dangerously 0.005447 close 0.002419 establishing 0.00435  
sons 0.002557 restrict 0.003849 teaching 0.003530 logic 0.002872 deviates 0.005777 adult 0.003896 role  
alt.atheism 51291 subject 0.000917 fluids 0.029931 liquids 0.036807 mikec 0.177459 sail 0.126920 labs (C  
alt.atheism 51292 subject 0.000366 gospel 0.009368 dating 0.011881 kmr4 0.020265 cwru 0.015082 edu  
usly 0.006719 levels 0.008293 pointed 0.007886 history 0.006052 important 0.005716 fussr 0.016697 ma  
05810 consequence 0.010897 self 0.006185 contradiction 0.020632 affiars 0.016697 ever 0.009688 able  
alt.atheism 51293 subject 0.000898 morality 0.019349 constant 0.022042 biblical 0.020348 rape 0.02274  
017972 elimination 0.029314 freewill 0.101283 shown 0.018533 even 0.007649 attempted 0.023149 hov  
alt.atheism 51294 subject 0.000665 gospel 0.017019 dating 0.021585 i3150101 0.019287 dbstu1 0.01928  
0.040973 match 0.015766 case 0.008142 gospels 0.020845 zillions 0.026692 spook 0.022701 stories 0.01  
alt.atheism 51295 subject 0.001178 contradictions 0.018210 kmr4 0.016294 cwru 0.012127 edu 0.00336  
ly 0.019212 benevolent 0.020402 laws 0.010848 attempt 0.011302 explain 0.010561 away 0.008138 desc  
alt.atheism 51296 i3150101 0.018177 dbstu1 0.018177 benedikt 0.035050 rosenau 0.017910 subject 0.0

## 测试集生成的TFIDF文档

testTfidf - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

alt.atheism 51060 mathew 0.004397 mantis 0.002679 subject 0.000066 alt 0.002651  
ently 0.000763 answered 0.000796 please 0.000631 note 0.001455 arguably 0.0010  
rs 0.003105 limit 0.000726 specific 0.000596 rather 0.004231 making 0.001006 flat (C  
01653 mean 0.002177 word 0.001493 referring 0.000778 also 0.002232 shades 0.00  
valid 0.002852 finding 0.000723 single 0.001049 hand 0.001002 question 0.002517  
6 every 0.001204 kind 0.000452 really 0.001301 applicable 0.000937 detectable 0.00  
0.000621 inconsistent 0.001026 tend 0.003487 play 0.000520 digression 0.001294 a  
874 almost 0.000497 everything 0.001013 said 0.001447 loses 0.001026 refer 0.002  
made 0.001912 little 0.000777 impact 0.000787 outside 0.001182 eastern 0.000815  
0.001766 campaigns 0.001122 long 0.000759 campaign 0.001692 discriminate 0.00  
04299 upbringing 0.002386 rebelling 0.001319 parents 0.000745 attempt 0.000631  
rate 0.000863 enough 0.000802 reason 0.000876 discourage 0.001035 immoral 0.00  
01500 organization 0.000625 behavior 0.000696 deteriorated 0.001386 born 0.001  
0.000922 consequence 0.000979 personality 0.000924 makes 0.002298 honestly 0.00  
e 0.001319 reform 0.000889 hope 0.000473 leaving 0.000758 mark 0.000490 histor  
en 0.000889 mankind 0.000912 money 0.000507 buildings 0.000899 effort 0.00132  
isinterpret 0.001433 perversion 0.001206 intended 0.000666 unambiguous 0.00119  
t 0.000695 quick 0.000654 denounce 0.001253 odd 0.000783 famous 0.002466 phil  
0632 messages 0.001435 fundamental 0.000801 following 0.000901 surprised 0.00  
alt.atheism 51119 i3150101 0.007565 dbstu1 0.007565 benedikt 0.014587 rosenau  
06460 attributions 0.010095 might 0.002796 elder 0.009681 figure 0.004799 either  
even 0.008885 association 0.006175 paul 0.004231 peter 0.009468 generally 0.0150  
967 edited 0.007185 oldest 0.025001 antiquated 0.010996 estimates 0.007565 com  
alt.atheism 51120 mathew 0.056379 mantis 0.019083 subject 0.000703 university 0  
careful 0.016116 mention 0.013253 jesus 0.012117 bible 0.012584 heard 0.009471 s  
alt.atheism 51121 strom 0.181949 watson 0.061082 ibm 0.059431 com 0.011288 ro  
alt.atheism 51122 i3150101 0.006580 dbstu1 0.006580 benedikt 0.012687 rosenau  
ian 0.003562 fact 0.002725 contradiction 0.006394 unresolvable 0.010348 reasons (C  
choices 0.011612 independently 0.006754 therefore 0.007747 attribute 0.006778 kr  
ne 0.003458 appreciate 0.004342 object 0.004751 deeds 0.006708 feel 0.003448 ab

#### 第四步, knn算法预测分类 knn.py

##### 1) 方法compute\_similarity(trainData,testData):

计算测试文档和训练文档之间相似性, 首先取两个文档共有的单词构成向量, 然后计算两个向量之间的cosine值, 将cosine值作为两文档之间的相似度返回。

##### 2) 方法classify(k,test,trainMap):

根据设定的k值, 计算当前测试文档与训练文档集中每一个文档的相似性, 然后进行排序取相似度最高的前k个类别用以计算当前测试文档的预测分类, 计算方法采用加权计算, 权重为相似度, 选择加权之后最高的分类作为测试文档的预测分类返回结果。

##### 3) 方法test():

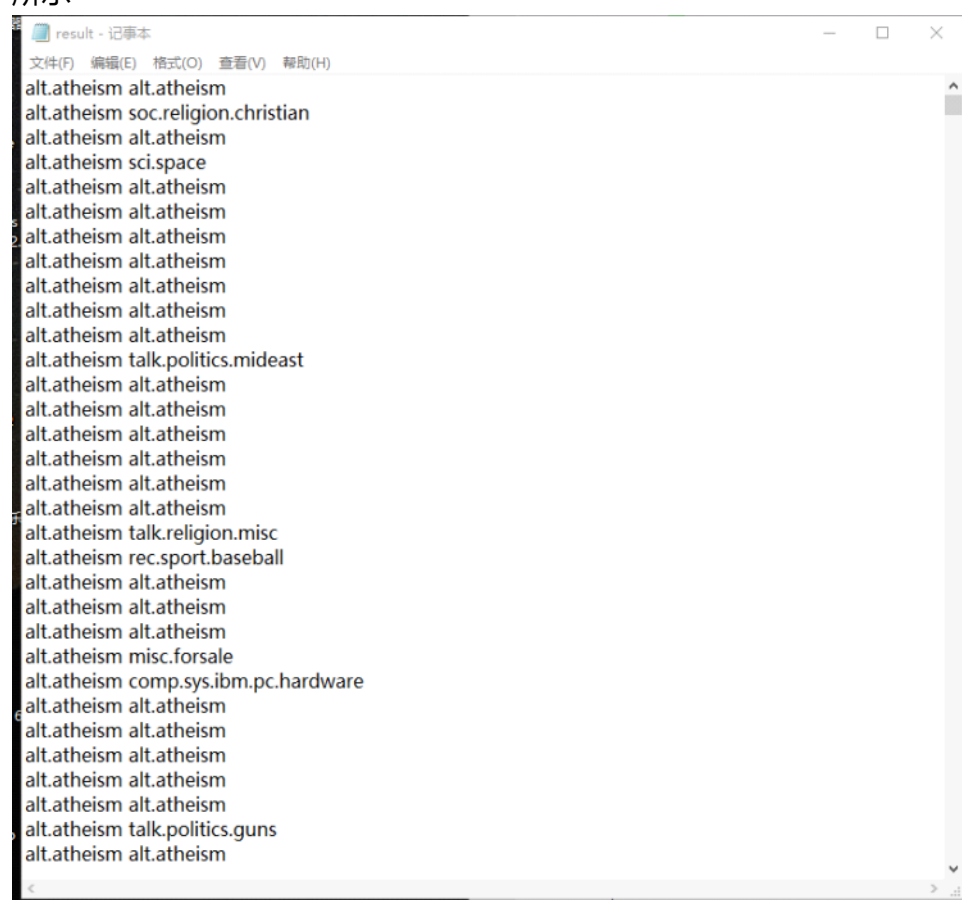
测试knn算法, 读取测试集文档和训练集文档, 对测试集中的每一条文档调用classify方法计算文档的预测分类, 然后与文档的原始分类相比, 统计正确分类的结果, 计算knn算法的准确率, 并将文档的原始分类 预测分类写入文件中。

##### 4) 方法analyse():

对test()方法分类的结果进行进一步分析, 统计不同类别的分类准确率, 并写入文件保存。

#### 实验结果及分析:

预测结果记录在result.txt文件中, 第一列为文档原始分类, 第二列为文档预测分类, 如下图所示



k值进行了多次调整, 选取了7/15/20/25/30进行实验, 统计不同k取值时的预测准确率, 统计结果如下

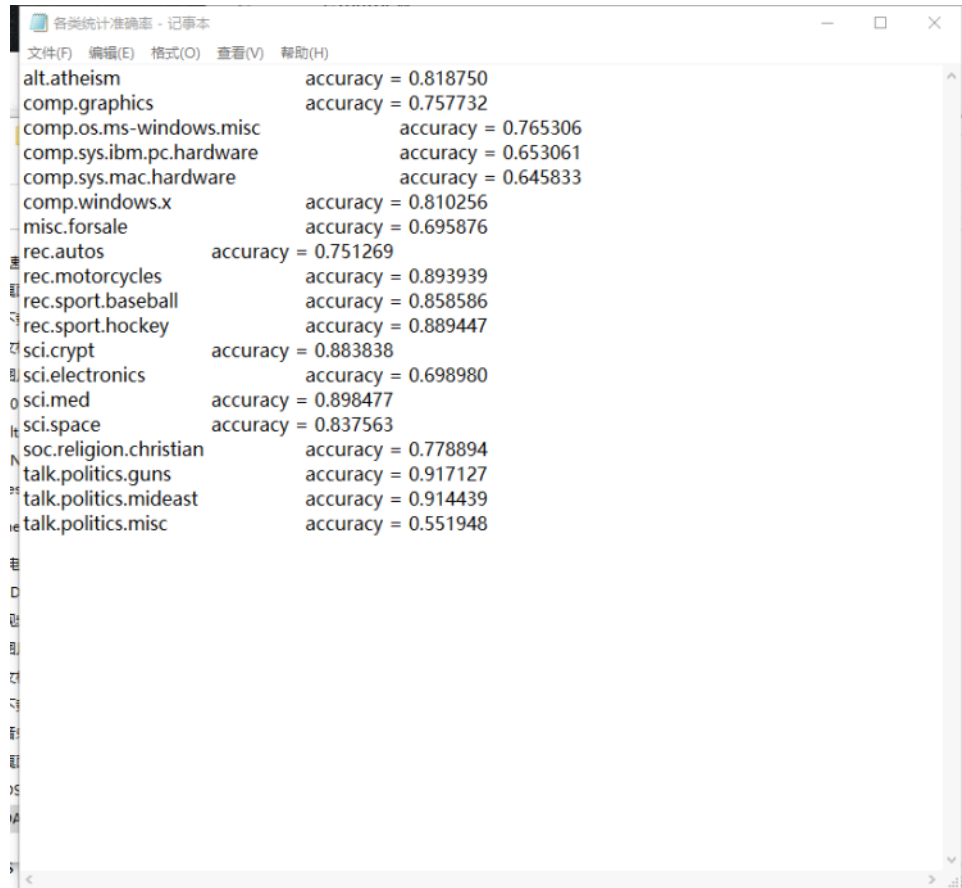
#### k取值与准确率

k	acc
7	0.745



15	0.769
20	0.782
25	0.791
30	0.788

从实验取的k值来看在k=25时预测准确率最高，但是也不是很高，所以我对预测结果进行了进一步分析，对每个类的预测结果单独分析，用analyse ()方法基于之前预测的结果计算每个类的预测准确率，保存在 各类统计准确率.txt中，结果如下图所示



可以观察到大多数类中预测准确率比较高，但是其中 misc.forsale sci.electronic talk.politics.misc的准确率偏低，从而拉低了整体的预测准确率。

总结：

通过在20news-18828数据集上实现knn算法，从最初的数据预处理到构建SVM模型，最后实现knn算法，然后改进算法。但仍然存在不足之处，算法的预测准确率不是很高，而且效率低下，预测准确率只有0.791我想进一步要做的是改进数据预处理过程和文档构建向量的过程，如何更好地用一个向量表示当前文档，如何获取到文档的主题。每次执行knn算法耗时比较长，算法效率比较低，可能要改进一下向量表示，提高运算速度。