实验报告汇总

姓名:周延　学号:201814853

实验报告 1——VSM+KNN

本实验使用 python 版本为 python3.6 操作系统为 ubuntu16.04

实验使用数据文件夹绝对路径为 pwd/../20news-18828

本实验操作分别在三个文件:readfile.py vsm.py knn.py

1.readfile.py 用于将所有数据绝对路径存放于列表 listfile 中

2.vsm.py 用于计算文档的 vsm,用字典 file_word_dict 表示,最后存储在 vsm.json 文件中

3.knn.py 用于通过 KNN 分类器计算测试集属于哪个类,计算正确率

VSM:

1. 通过函数 get_word_nubmber 用来获取每个文件中单词数目存入字典 file_word_dict,该函数会将单词小写化,使用 nltk 库进行词性还原,去掉不属于单词的符号,file_word_dict 结构为 {file-{word-numbers}},结果写入 file_word_dict.json 中;

2. 通过 file_word_dict 计算出所有文件内单词和单词出现个数存入字典 word_numbers,再对 file_word_dict 进行一些处理去掉不再 10-1000 之间的单词;

3.通过函数 com_VSM 计算出每个文件内每个单词打 TF-IDF,同样存入 vsm-word 中,结构为{file-{word-tf*idf}},结果写入 vsm.json 文件中。

0.7285145712449033, "atheist": 3.8820950628977595, "resource":
1.8686841678952038, "december": 0.9589888728156154, "version":
1.3728862691556472, "address": 1.138030192584587,
"organization": 1.0044581010149467, "usa": 1.6211008528363156,
"freedom": 1.0995315351758075, "religion": 2.19392220374364,
"foundation": 1.3645203888246091, "darwin": 3.1982551765274403,
"fish": 3.448801988011216, "bumper": 1.1427472111460681,
"sticker": 1.062974980489618, "assorted": 2.572509598511201,
"write": 2.0826713119920086, "box": 1.1329558273363958,
"madison": 1.205852639800365, "wi": 1.0954569621369095,
"telephone": 2.0263097030550057, "evolution":
1.9688853754246372, "design": 1.3507730278270795, "sell":
0.9307068720825785, "symbol": 1.055001782660802, "stick":
0.7833298389235881, "foot": 0.7518293105034667, "written":
1.0044581010149467, "inside": 0.7154370502426797, "deluxe":
1.2177315076268185, "plastic": 0.9621556127878057, "laurel":
1.4145889746473 06, "canyon": 1.2663588487128865, "north":
0.7643733197120476, "hollywood": 1.3107456704373726, "bay":
0.8977702521198128, "area": 0.8269825463666388, "gold":
0.982351597931234, "mailing": 0.8361297114798554, "net":
0.8932664644869521, "directly": 0.6897939182862077, "price":
0.5625259569894053, "american": 2.2419503380774444, "press":
2.935355754742116, "publish": 2.60058500382752, "various":
1.3182006885175102, "critique": 1.8438010820598743, "bible":
2.6187657633603965, "biblical": 0.8683191603518844,
"contradiction": 1.4780506190948952, "handbook":

KNN:

1.将所有数据分为 20%测试数据和 80%训练数据

2.如计算 vsm 时所做,分别计算出 test_data 和 train_data 的 vsm,使用 tf-idf 表示

3.对于每一个测试数据遍历训练集计算距离 cos 值

4.排序后取出 K 个最大的训练集文档,K 个文档中类最多的即为测试文档的类

5.验证分类是否正确,计算完所有测试数据后再计算分类成功率

6.以上步骤重复 5 次

7.调参,改变参数 K,找出成功率最大的 K

结果:

K 统计 3-8 之间打参数,平均成功率分别为:K=3 成功率 70.34%

K=4 成功率 74.91%

K=5 成功率 77.23%

K=6 成功率 78.42%

K=7 成功率 78.65%

K=8 成功率 78.72%

K=6 后趋于稳定

K=6 比较好

实验报告 2——navies bayes

本次实验环境为 ubuntu16.04 python 版本为 python3.6

实验要求：

本次实验采用 navie bayes 对 20newsgroups 进行分类,模型构造直接使用

sklearn 库

实验步骤：

1、获取训练数据和测试数据

```
train_d = fetch_20newsgroups(subset = 'train',categories = categories);
test_d = fetch_20newsgroups(subset = 'test',categories = categories);
```

2、获取数据集特征

```
vectorizer = HashingVectorizer(stop_words = 'english',non_negative = True,n_features =10000)
fea_train = vectorizer.fit_transform(train_d.data)
fea_test = vectorizer.fit_transform(test_d.data);
```

3、构造 naivebayes 多项式模型

```
clf = MultinomialNB(alpha = a)
```

alpha 为平滑参数 默认 1.0

4、训练集合上进行训练, 估计参数

```
clf.fit(fea_train,train_d.target);
```

5、对测试集合进行预测 保存预测结果

```
pred = clf.predict(fea_test);
```

实验结果：

alpha= 0.01 precision= 0.8005366715683742
alpha= 0.05 precision= 0.808135994679238
alpha= 0.1 precision= 0.8093082169779041
alpha= 0.15 precision= 0.809488280285531
alpha= 0.2 precision= 0.8094627780947201

alpha= 0.15 时 准确率最高

构造模型选用 alpha= 0.15

实验报告 3——聚类

本实验使用 python 版本为 python3.6 操作系统为 ubuntu16.04

实验要求：

测试 sklearn 中以下聚类算法（Kmeans、Affinity propagation、Mean-shift、Spectral clustering、Ward hierarchical clustering、Agglomerative clustering、DBSCAN、Gaussian mixtures）在 tweets 数据集上的聚类效果

评价指标为 NMI（Normalized Mutual Information）

实验步骤为：

1、处理原始数据：按行读取 Tweets.txt 并转换为字典置于列表中，对该列表进行处理，将 text 标签内容置于 test_data 列表，将 cluster 标签内容置于 labels_true 列表中，使用 TfidfVectorizer 将 test_data 列表中字符串向量化置于 X

2、利用 sklearn 中自带的 Kmeans、Affinity propagation、Mean-shift、Spectral clustering、Ward hierarchical clustering、Agglomerative clustering、DBSCAN、Gaussian mixtures 算法函数对 X 聚类，得到 labels

3、使用 sklearn 自带的计算 NMI 的函数，计算 labels_true 和 labels 的 NMI，进行对比

实验结果如下：

聚类算法	NMI
Kmeans	0.6493775338318443
Affinity propagation	0.4839682646264274
Mean-shift	0.48568072835637743
Spectral clustering	0.539574609793204
Ward hierarchical clustering	0.6644202079888861
Agglomerative clustering	0.6644202079888861
DBSCAN	0.4248045834640204
Gaussian mixtures	0.7220048834646094