

Data Mining 实验报告

Homework VSM and KNN

201834889

张玉卉

一实验要求:

- 1、使用 github 管理项目
- 2、建立向量空间模型来表示文本
- 3、KNN 实现文本分类

二实验步骤

VSM

1. 读取所有文档，遍历文件夹
2. 对获取的文档进行处理 tokenization、normalization、stopwords、Stemming、punctuation_remove
3. 预处理遍历文件的同时计算词频（一个词在所有文档出现的次数），并保存处理好的文档和单词词典
4. 过滤掉单词词典中词频太低和过高的单词
5. 计算词典中单词的 DF、IDF
6. 生成文档向量 vectors，为向量打标签表明其所属类别

KNN

第一种方法（未使用类库）

1. 使用 VSM 生成的向量
2. 将向量划分为测试和训练，随机选取 20%作为测试集，剩下 80%作为训练集。
4. 进行 knn 分类：采用余弦相似度进行计算文档相似度。计算测试文档与训练文档的余弦，选出 K 个距离最近的训练文档，统计排序 K 个文档中哪个类别最高，则该测试文档就为哪个类别。

5. 计算 KNN 分类的准确率，将计算出来的结果与真实值进行比较，统计归类正确的文档数目除以总的测试集文档数目，统计并输出正确率
6. 调整不同的 K 值，查看正确率变化


第二种方法（使用类库）

使用 `sklearn neighbors KNeighborsClassifier` 的分类器和 `numpy`

1. 读取数据，分别以矩阵的形式读取训练集和测试集合
2. 读取训练集和测试集的标签
3. 获取类库中的 `KNeighborsClassifier` 分类器，输入测试集和测试集标签进行训练
4. 预测测试集的标签
5. 比较预测值和真实值，计算准确率

三实验结果

- 1 VSM 生成了 5875 的词典和文档向量，向量保存为 CSV 格式，500 多 MB 无法上传 github（100MBlimit），所以随机上传了部分 vector（低于 100MB）

 vectorsall.csv	2018/11/5 0:52	XLS 工作表	545,704 KB
--	----------------	---------	------------

- 2.

第一种方法 KNN 的正确率

k	accuracy
1	0.62853
3	0.67708
6	0.65208
10	0.66666

20	0.64583
25	0.64583
40	0.59291
100	0.51666
300	0.46041

第二种方法 KNN 只跑了少量数据，K 值在 1-30 左右时 accuracy 大约在 0.69-0.78 左右