VSM and KNN

董广顺 201834860

一：总体目标

以 VSM 描述文本，以 KNN 为文本分类。

其中 VSM 的概念是文本可以用文本里所含的单词和单词出现的频率信息来描述，可以建立一个含有训练集中所有单词的总词典，文本里出现的词可以用对应于总词典的向量来表示，因此对文本的处理就可以转化为对向量的运算。并且文本间的相似度可以用向量的空间上的距离关系来表示。

而 knn 就是取 k 个与测试目标距离最近的元素，在这 k 个元素中，属于哪一个分类的元素最多，就认为测试目标更可能属于那个分类。

二：实现步骤

2.1：文本预处理

这一步的目的是为 VSM 做铺垫，VSM 要用文本里的单词表示文本信息，就需要先把单词提取出来。文本中含有大量的符号、数字和无意义的介词等，这对表示文本是没有价值的，要把它们去掉。这一步在 python 中可以调用 textblob 和 nltk 库的一些方法。

2.2：建立词典

在对文本做预处理的同时，我们就可以开始建立词典。这使我们可以只遍历一次文本，提高程序速度。因为算 KNN 需要用测试目标与每个文本计算距离，所以其实每个文档也都需要一个词典，与之相对的，我们还需要建立一个总的词典。在每个文档的词典中，我们可以存储 TF 值，而在总词典中，我们可以存储 IDF 值。TF 与 IDF 相乘，就是我们用来表示单词（term）在文档中重要程度的权重。

其中采用的数据结构是嵌套的字典，对 train 数据集而言，为每个文件夹建立一个字典，字典的 key 是文件夹里每个文档的路径，即名称，value 是每个文档的词典。文档的词典也用字典的数据结构，key 是 term，value 是 TF 值。程序中的方法其实返回的是每个文件夹的字典。因此有多少个文件夹就要调用多少次那个函数，最后再建立一个 list 存储这些字典。总的词典属于全局变量，在遍历完所有文件夹的文档后，总词典也构造完成，但此时总词典里的 value 是 DF，还不是 IDF，要遍历完所有文档知道总的文档数目后才能计算 IDF。

2.3：构建向量

在总词典和每个文档的词典都建立好后，就可以开始构建向量了。其中向量是对应每个文本而言的，向量的维度与总词典所含 term 的数量相同，每个维度的值是总词典的 IDF 乘以文本的 TF，确定属于同一维度的关键是总词典中的 term 与文本中的 term 对应起来。文本中没有的单词，在向量中这一维度就为 0。

2.4：向量同一化

对于不同大小的文本而言，其 TF 值的平均值是不一样的，因此只用 TF 的绝对大小表示 term 的重要程度是不准确的，因此需要根据文本中所含 term 的数量对 TF 值进行一些校准

2.5：knn

向量建立好后，就可以进行 knn 算法了。先为测试文本也建立一个向量，方法同上，然后用这个向量与全部 train 数据集的文本向量计算距离，取出其中 k 个距离最近的向量，计算哪个文件夹中含有的这个 k 个向量对应的文本最多，我们就认为 test 文本属于哪个文件夹。并且程序的方法中返回该文件夹的路径。

2.6：程序执行

在 main 程序中，对 test 数据集进行遍历，对每一个文本进行归类。由于我们知道 test 中的文本应该属于哪个文件夹，所以我们可以计算正确率，以此来调整 K 值，以达到更高正确率。

最开始读文件的时候，被编码格式卡住，python2.7 的转变编码格式的语句在 3.7 里不能用

为了只遍历一遍文本，需要在为每个文本建立词典的同时建立总的词典。并且降低循环层数，可以大幅提高运算速度。

为了将文本的词典与文本对应起来，采用了嵌套词典的数据结构。外层词典的 key 是文档路径，value 是文档的字典。

然后现在对建立词典这一步而言，有一个缺陷是每次运行程序都要算一次词典和向量，这使得程序可以对不同的数据集复用，但这其实没有什么意义，对不同的数据集其实总要做一些改变的，因此程序的设计应该更倾向于针对特定数据集，可以有一次计算把词典和 VSM 都算好存到硬盘上，然后之后的每次运行都可以直接去硬盘取这些数据而不用再算一遍。

选用的数据集是 20news-bydate，这个数据集的 train 和 test 数据集是分开的，与 18828 那个不同，所以没有做把数据集按 80%和 20%分开那一步。

数据集的筛选做的不够。

程序的复用做的不够好。因为想只遍历一次 train 数据集就同时建好每个文本的词典和总词典，因此在 createDict 的时候要同时往总词典里添词。而对于 test 数据集而言，也需要 createDict 然后计算 VSM，但是不能调用之前写的 createDict 函数，因为那个函数里还有对总词典的操作是不该运行的。这个其实完全是可以复用的。

VSM 的 tf 值在遍历文本建立词典的时候就做好，IDF 存在大词典的 value 里。