作业报告-1

学　号：201814841　　姓　名：徐　强

10月8日收到此次作业后，经过近一个月的学习和实践，基本完成了作业要求的内容。报告内容主要分为作业完成情况、学习收获和几点思考三个部分。

一、作业完成情况

（一）学习 python

对于一个仅在十年前学过C语言且基本没有编程基础的学生，学习 python 还是比较困难的，我利用两周时间结合老师教学课件给出的教程及《零基础入门学习 python》一书，了解了 python 的基础知识，根据老师教学课件的相关内容搭建了 python 编程环境和注册 GitHub 账号，掌握了一些基本操作。因 deadline 临近，只能先开始慢慢写程序，边写边学。

（二）程序实现

按照 TF-IDF 和 KNN 文档处理流程，将程序划分为五个步骤编写，具体实现过程如下：

1.文档读取

编写文件读取时比较顺利，基本能够利用 OS 模块及前期所学编程知识解决，以嵌套列表的结构存储了所有文档。

2.预处理

使用 String 模块替换了文档中其它字符内容，使用老师 PPT 所给的 Textblob 模块进行了 tokenization，使用 NLTK 模

块进行了 lemmatize、steming 和去掉 stopwords，使语料库里的每个文档均成为一个由一系列单词组成的列表。

3.统计词频生成词典

使用 collections 模块的 Counter 类统计了所有单词在所有文档中的词频，并根据统计出的词频，设置了高频词和低频词上下限（9 ≤ wordfrequency ≤ 1500）去除了 non-informative words and rare words，生成了一组由 23211 个单词组成的词典。

4.计算 TF、IDF 生成 VSM

使用 collections 模块的 Counter 类分别计算每个文档对于词典的词频 N1、语料库中文本的总数 N2、语料库中包含词 x 的文本数 N(x)，利用 TF 公式 a+（1-a）N1/max N1（a=0.1）和 TDF 公式 log((N2+1)/(N(x)+1)+1)（该公式主要解决在一些特殊的情况下会出现的小问题，对 IDF 做了一些平滑），最后利用 numpy 模块完成 VSM 的生成。

5.利用 KNN 完成 test 文档测试分类并统计正确率

使用 sklearn.model_selection 模块对语料库和标签进行了随机选取，按照作业要求取出 80%文档做训练，剩余 20%做测试，经过计算测试文档和训练文档向量夹角的余弦值来确定向量所表示文档的相似性，设置的 K 值为 3，取出 cos 值最大的 K 个，统计划分出测试文档的分类，对比测试文档初始的分类，经过五次计算平均的正确率为 0.709506。

二、学习收获

1.通过编写 TF-IDF 和 KNN 程序，初步掌握了一些程序编写的基本方法和 python 的列表、字典和数组等结构的处理及计算的基本方法，因编程经验不足造成了语句产生的错误较多，通过请教同学也掌握一些程序调试的方法。

2.对于 TF-IDF 及 KNN 主要通过老师所发上课课件进行理解和巩固，处理语料库及词频统计中所使用的模块主要通过《NLTK 基础教程》和 CSDN 网站学习使用方法，通过此次作业基本掌握了上述内容。

3.通过调整设置词典生成中 wordfrequency 上下限和 KNN 中的 K 值（具体记录数据见表格），选取的了正确率较高参数，进一步理解了 TF-IDF 及 KNN 原理。

三、几点思考

1.在上述的程序中，训练集和测试集是在生成 VSM 后划分的，这样测试集在词典选取时已经在训练集中，影响了词典的生成，从而影响了后面 KNN 的正确率。后面编写了在读取文档时按 80%随机选取的函数，但是影响后续处理的数据结构，故未进行测试，争取在下次作业实现。

2.在整个处理过程中基本默认列表中文档的顺序没有发生改变，这样才能和储存标签的列表一一对应，如何同步处理标签和文档，需要一个合适的数据存储结构才能实现，但未能实现，还需要继续学习。

作业报告-2

学　号：201814841　　姓　名：徐　强

经过三周的学习和实践，基本完成了作业要求的内容。报告内容主要分为作业完成情况、学习收获和几点思考三个部分。

一、作业完成情况

按照 NBC 处理流程，将程序划分为三个步骤编写，具体实现过程如下：

（一）文档读取并划分训练集和测试集

利用 sklearn.model_selection 模块实现了随机划分 80%训练集和 20%测试集，并根据后面 NBC 需要将文档用字典结构存储，键为文档分类，值为每个元素为一篇文档的列表。

（二）文档预处理并去除高低频词

1.基本和 KNN 的处理过程相同，使用 String 模块替换了文档中其它字符内容，使用老师 PPT 所给的 Textblob 模块进行了 tokenization，使用 NLTK 模块进行了 lemmatize、steming 和去掉 stopwords 和单词长度小于 3 的单词，使字典结构里的每个文档成为一个由一系列单词组成的列表。

2.使用 collections 模块的 Counter 类统计了每个类的单词在每类文档中的词频，并根据统计出的词频，设置了高频词和低频词上下限，去除了每类中的 non-informative words and rare words。

（三）进行 NBC 并计算正确率

1.根据 Bayes 公式进行分类的基本思想是：

$$P(\text{“属于某类”}|\text{“具有某特征”}) = \frac{P(\text{“具有某特征”}|\text{“属于某类”})P(\text{“属于某类”})}{P(\text{“具有某特征”})}$$

计算已知测试数据的特征的条件下要计算测试数据属于某类的最大概率（后验概率），等于计算已知训练数据属于某类的条件下所具有特征的概率（相似度）乘以属于训练数据中某类的概率的最大值。

2.按照 NBC，假设一个数据中的特征之间独立：

$$P(x|c)=P(x1,x2\ldots xn|c)=P(x1|c)*P(x2|c)\ldots P(xn|c)$$

3.按照平滑后的多项式模型分别计算测试数据文档中每个单词在一个类出现的概率：

$$\hat{P}(w_i \mid c) = \frac{count(w_i,c)+1}{\left(\sum_{w\in V} count(w,c)\right) + |V|}$$

4.将上述概率值的连乘取对数改为连加，计算测试集中每篇文档属于训练集中每个类的概率，划分至最大概率值的那一类，完成 NBC。

5.对比测试集原本属于的类和 NBC 后的类，在考虑权重的情况下，计算 5 次后的平均正确率约为 89.57%。

二、学习收获

1.重新学习了 Bayes 公式并将其用在具体问题的解决中，理解了 naive 的本质就是将数据的各个特征值近似看做相互

独立，进而将条件联合概率近似为条件独立概率的连乘。

2.完成了上次作业中随机划分训练集和测试集的任务。

三、几点思考

1.NBC 基本没有可以调整的参数，优化的方法不多。经过权重计算正确率提升效果不是很明显，且计算权重的时间复杂度有点高，程序运行时间偏长。

2.NBC 对于数据的预处理程度要求比较高，特征选取的不同也会影响 NBC 的效果。通过调整设置类词典上下限（具体记录数据见表格）观察 NBC 的正确率，感觉去除高低频词对整个 NBC 结果影响不大。

3.调整了训练集和测试集的大小（具体记录数据见表格），在训练集为 10%的情况下 NBC 还能得到约 70% 的正确率，感觉 NBC 方法还是效果不错的。

作业报告-3

学　号：201814841　　姓　名：徐　强

经过四周的学习和实践，基本完成了作业要求的内容。报告内容主要分为作业完成情况、学习收获和几点思考三个部分。

一、作业完成情况

按照聚类处理流程，将程序划分为四个步骤编写，具体实现过程如下：

（一）文档读取

Tweets 文件内容为 JSON 格式，打开文件后按行进行了读取，并将数据和聚类标签分别存放在列表结构中。

（二）计算 TF-IDF

与 KNN 中生成 VSM 方法类似，利用 Sklearn 库中的 CountVectorizer 和 TfidfTransformer 模块，计算出所有文本数据的的 TF 和 IDF，并生成 TF-IDF 权重矩阵。

（三）分别利用 7 种聚类方法进行聚类

1.K-means

1.1 算法

a.随机选择 k 个中心。

b.重复下列步骤直到达到停止条件停止条件：聚类中心不再发生变化或所有的距离最小或迭代次数达到设定值。

遍历所有样本，把样本划分到距离最近的一个中心。

计算每个簇的平均值作为新的质心。

1.2 算法评价

该算法聚类效果不错，也容易理解，速度快；但是需要自己确定 K 值，且初始中心的选取会影响最终的聚类结果。

2.Affinity Propagation

2.1 算法

AP 聚类是通过在样本对之间发送消息直到收敛来创建聚类。然后使用少量示例样本作为聚类中心来描述数据集，聚类中心是数据集中最能代表一类数据的样本。在样本对之间发送的消息表示一个样本作为另一个样本的示例样本的适合程度，适合程度值在根据通信的反馈不断更新。更新迭代直到收敛，完成聚类中心的选取，因此也给出了最终聚类。

2.2 算法评价

与 K-Means 等聚类算法不同的地方在于 AP 聚类不需要提前确定聚类的数量，即 K 值，但是运行效率较低。

3.Mean-shift 均值迁移

3.1 算法描述

Mean-shift 聚类的目的是找出最密集的区域，同样也是一个迭代过程。在聚类过程中， 首先算出初始中心点的偏移均值，将该点移动到此偏移均值，然后以此为新的起始点，继续移动，直到满足最终的条件。

3.2 算法评价

Mean-shift 也引入了核函数，用于改善聚类效果。除此之外，Mean-shift 在图像分割， 视频跟踪等领域也有较好的应用；但需要先用工具计算 bandwidth，如果设置不合适会影响聚类效果。

4.Spectral Clustering 谱聚类

4.1 算法描述

将样本看作顶点，样本间的相似度看作带权的边，从而将聚类问题转为图分割问题：找到一种图分割的方法使得连接不同组的边的权重尽可能低(这意味着组间相似度要尽可能低)，组内的边的权重尽可能高(这意味着组内相似度要尽可能高)。

4.2 算法评价

能够识别任意形状的样本空间且收敛于全局最优解。

5.Agglomerative Clustering 层次聚类

5.1 算法描述

自底向上的层次聚类。

初始时，所有点各自单独成为一类，然后采取某种度量方法将相近的类进行合并，并且度量方法有多种选择。合并的过程可以构成一个树结构，其根节点就是所有数据的集合，叶子节点就是各条单一数据。

AgglomerativeClustering 中可以通过参数 linkage 选择不

同的度量方法， 用来度量两个类之间的距离，可选参数有ward（类间距离等于两类对象之间的最小距离）、complete（类间距离等于两组对象之间的最大距离）、average（类间距离等于两组对象之间的平均距离）三个。

5.2 算法评价

可能会产生聚类结果得到的类的大小不均衡的结果。由于层次聚类涉及到循环计算，所以时间复杂度比较高，运行速度较慢。

6.DBSCN 密度聚类

6.1 算法

从某个选定的核心点出发，不断向密度可达的区域扩张，从而得到一个包含核心点和边界点的最大化区域，区域中任意两点密度相连。

6.2 算法评价

不需要指定 cluster 的数目，聚类的形状可以是任意的，能找出数据中的噪音，对噪音不敏感，聚类结果几乎不依赖于节点的遍历顺序；但需要设置合适的领域半径和最小核心点数量。

7.GaussianMixtureModel 混合高斯模型

7.1 算法描述

聚类算法大多数通过相似度来判断，而相似度又大多采用欧式距离长短作为衡量依据。而 GMM 采用了新的判断

依据：概率，即通过属于某一类的概率大小来判断最终的归属类别。

7.2 算法评价

GMM 的优点是投影后样本点不是得到一个确定的分类标记，而是得到每个类的概率，这是一个重要信息。GMM 不仅可以用在聚类上，也可以用在概率密度估计上。

（四）利用 NMI 方法进行评分

最终结果为：

```
The K-Means score is:0.769170
The Affinity Propogation score is:0.794534
The Mean-Shift score is:0.742985
The Spectral Clustering score is:0.781465
The Agglomerative Clustering-average score is:0.896244
The DBSCAN score is:0.654360
The Gaussian Mixtures score is:0.790958
```

二、学习收获

1.通过实验重新复习了老师课堂讲的 K-means 聚类、Agglomerative Clustering 层次聚类和 DBSCN 密度聚类三种方法，能够做到基本掌握，对 Mean-shift 均值迁移、Spectral Clustering 谱聚类、GMM 混合高斯模型、Affinity Propagation 和 BIRCH 五种方法有了初步的了解。

2.对 TF-IDF 的权重矩阵有了新的认识，可以用在对文本进行分类和聚类的多种任务中。

三、几点思考

1.层次聚类得分较高，密度聚类得分较低，有可能和密

度聚类的领域半径和最小核心点数量的设置有关。

2.从程序运行时间来看，谱聚类最快，GMM 时间最长，GMM 方法收敛速度慢主要原因可能为数据不足时估算协方差矩阵困难，同时算法会发散并且找具有无穷大似然函数值的解，需人为地对协方差进行正则化。