大数据技术与应用作业二实验报告

小组成员与分工：

王子旭：数据清洗 黄廷基：聚类与可视化

贺嘉瑞：问题回答 张鑫钰、宋昕洋：数据分析、实验报告

一、实验流程说明

1. 读取原始数据文件

采用pandas库可以直接读取.csv数据文件，并查看数据各方面信息。

1. 数据清洗
2. 去停用词

根据网上常用的停用词库，使用jieba库中lcut函数分割文本，逐词判断并删除文本中停用词，同时采用jieba库的posseg函数对剩余有效文本数据进行词性分析，进一步筛选文本。

1. 去除特殊符号

调用pandas库函数去掉问题中’？’、标点符号、特殊符号等。

1. 去除异常文本

使用pandas库清理文本中空行、重复数据、仅含字母和数字的文本以及问题中小于三个字的行。

1. 保存处理后数据并生成tfidf矩阵

将处理后的文本数据按照初始顺序保存为.txt文件并输出所有统计词语的字典形式；调用sklearn库的TfidfVectorizer函数生成文本数据对应的tfidf矩阵。

1. 聚类

本次实验采用两种不同聚类方式，包括层次聚类与K-均值聚类，均可实现文本聚类目标。

1. 层次聚类

（1）采用主成分分析方法PCA对tfidf矩阵进行降维；

（2）调用AgglomerativeClustering库函数实现对数据的层次聚类；

（3）给聚类结果的簇贴标签以描述每个簇对应文本的中心话题；

（4）输出聚类结果的二维图像及运行时间。

1. K-均值聚类
2. 使用PCA方法降维；

（2）调用KMeans库函数对降维后数据进行K-均值聚类；

（3）给聚类结果的簇贴标签以描述每个簇对应文本的中心话题；

（4）调用matplotlib绘制聚类结果二维图像，并确定各类中心点。

1. 所用模型与方法

本次文本聚类实验中，小组主要使用了K均值聚类与层次聚类方法，下面将对两种聚类方法分别进行介绍。

1. 层次聚类

（1）方法介绍

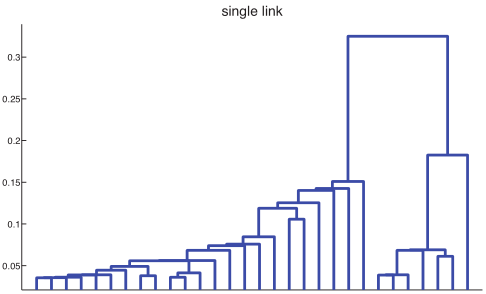
层次聚类可通过计算不同类别数据点间的相似度，在不同层次上对数据集进行划分，形成树状的聚类结构，包括自下而上合并和自上而下分裂两种方法。AgglomerativeClustering是一种常用的层次聚类算法，属于自下而上合并的方法。其原理是：最初将每个对象看成一个簇，然后将这些簇根据某种规则一步步进行合并，直到达到预设的簇类个数。

根据不同簇之间差异的不同定义，Agglomerative clustering有三种变体。[1]

① Single link

单链接聚类（single link clustering）也叫最近邻聚类（nearest neighbor clustering）。两个簇之间的距离定义为两簇中离得最近的两个元素之间的距离：

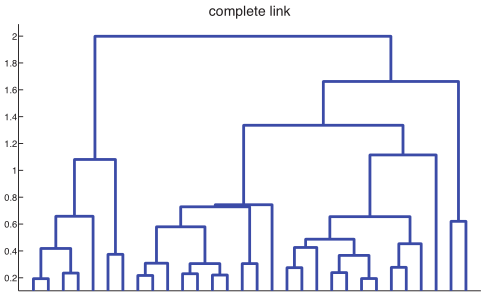
IMG_256



② Complete link

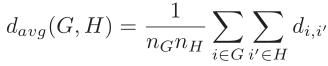
complete link clustering也叫最远邻聚类（furthest neighbor clustering）。两个簇之间的距离定义为两簇中离得最远的两个元素之间的距离：

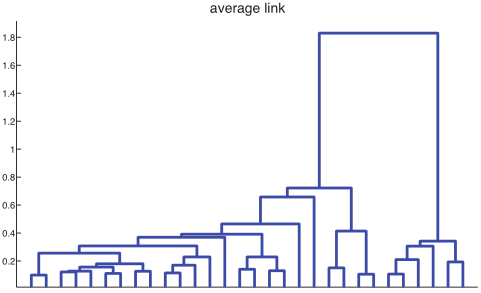
IMG_256



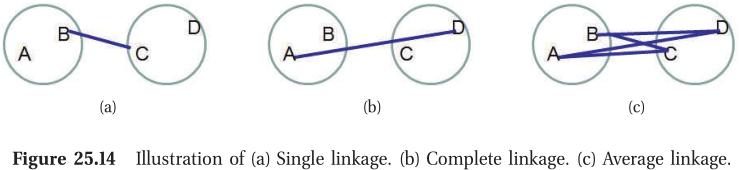
③ Average link

平均链接聚类（average link clustering）是上面两种聚类方式的折中，得到的是相对紧密的集群。平均链接聚类测量所有对之间的平均距离:





三种聚类模型的直观描述：



（2）算法流程[2]

输入：

● 数据集D；

● 聚类簇距离度量函数；

● 聚类簇数量K；

输出：

● 簇划分结果C={C1,C2,...,CK}；

算法步骤：

● 初始化：将每个样本都作为一个独立的类簇；

● 迭代过程：

计算聚类簇之间的距离，找出距离最近的两个簇，将这两个簇合并。

● 终止条件：聚类簇的数量为K。

（3）函数详解

sklearn.cluster.AgglomerativeClustering函数

参数:

n\_clusters：指定分类簇的数量。

connectivity：一个数组或者可调用对象或者None，用于指定连接矩阵。

affinity：一个字符串或者可调用对象，用于计算距离。

memory：用于缓存输出的结果，默认为不缓存。

compute\_full\_tree：如果该参数为True，当训练了n\_clusters后，训练过程不会停止而是继续训练生成一颗完整的树。

linkage：一个字符串，用于指定链接算法，包括：

‘ward’：单链接single-linkage算法

‘complete’：全链接complete-linkage算法

‘average’：均连接average-linkage算法

pooling\_func：一个可调用对象，它的输入是一组特征的值，输出是一个数。

属性：

labels：每个样本的簇标记

n\_leaves\_：分层树的叶节点数量

n\_components：连接图中连通分量的估计值

children：一个数组，给出了每个非节点数量

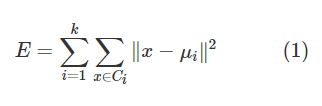
方法：

fit(X[,y])：训练样本

fit\_predict(X[,y])：训练模型并预测每个样本的簇标记

1. K-均值聚类
2. 模型介绍

KMeans算法在给定一个数k之后，能够将数据集分成k个“簇”。算法需要最小化平方误差：



其中μi是簇的均值向量，或者说是质心。|x−μi|2代表每个样本点到均值点的距离。

KMeans算法距离度量最常用的就是闵可夫斯基距离（亦即p范数），即当p=2时闵可夫斯基距离即为欧氏距离（2范数）；当p=1时闵可夫斯基距离即为曼哈顿距离（1范数）。

算法原理如下：对于给定的样本集，按照样本之间的距离大小，将样本集划分为K个簇。使簇内的点尽量紧密地连在一起，同时保证簇间的距离尽可能大。

1. 算法流程[3]

输入：样本数据集D，聚类簇数k

(1) 从样本中随机选取k个样本点作为初始的均值向量{μ1,μ2,⋯,μk}

(2) 循环以下几步直到达到停止条件：

（2.1）令Ci=**∅**(1≤i≤k)；

（2.2）对所有样本点计算他们到k个均值向量之间的距离，取其中距离最短的距离对应的均值向量的标记作为该点的簇标记，然后将该点加入相应的簇Ci；

（2.3）对每一个簇计算他们新的均值向量μi，如果相比之前的向量有变化，就更新为新的均值向量。

1. 函数详解[4]

sklearn.cluster.KMeans函数

参数：

n\_clusters：生成的聚类数。

max\_iter：执行一次k-means算法所进行的最大迭代数。

n\_init：用不同的质心初始化值运行算法的次数，最终解是在inertia意义下选出的最优结果。

init：此参数指定初始化方法，有三个可选值：’k-means++’， ‘random’，或者传递一个ndarray向量。

‘k-means++’ 采用特殊方法选定初始质心从而加速迭代过程的收敛。

‘random’ 随机从训练数据中选取初始质心。

’ndarray’ 需形如 (n\_clusters, n\_features) 并给出初始质心。

precompute\_distances：预计算距离，三个可选值‘auto’，True 或 False。

n\_jobs：指定计算所用的进程数，同时进行n\_init指定次数的计算。

random\_state：用于初始化质心的生成器。如果值为一个整数，则确定一个seed。

属性：

cluster\_centers\_：向量，表示聚类中心的坐标（n\_clusters, n\_features）。

Labels\_: 每个点的类别。

inertia\_：每个点到其簇的质心的距离之和。

方法：

fit(X[,y])：计算k-means聚类。

fit\_predictt(X[,y])：计算簇质心并给每个样本预测类别。

fit\_transform(X[,y])：计算簇并将X 转换为聚类距离空间。

get\_params([deep])：取得估计器的参数。

predict(X):predict(X)：估计每个样本最接近的簇。

score(X[,y])：计算聚类误差。

set\_params(\*\*params)：为估计器手动设定参数。

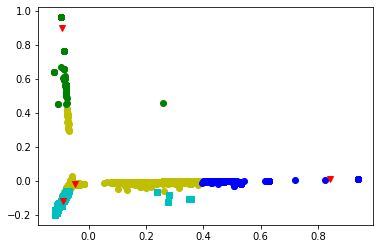
transform(X[,y]): 将X转换为聚类距离空间。

**三、实验结果分析**

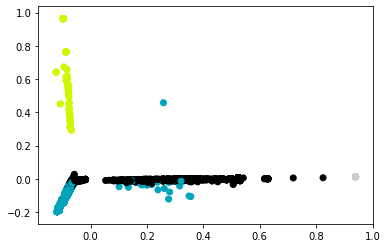
首先，依据上文中提到的聚类方法，我们选取了k-means聚类和Agglomerative clustering两种方法。

下面两张图反映了聚类后可视化结果：

·k-means结果可视化：



·Agglomerative clustering结果可视化：



从结果中不难看出，两种方法的结果是极为相近的，从一定程度上说明了聚类结果的可靠性。这里我们将k-means为例，将聚类后的结果做成词云。

以下是四类文本中使用的词语组成的词云：

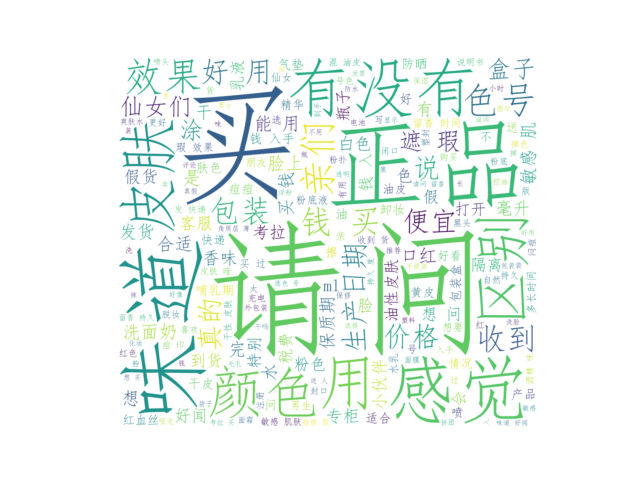
·类别一：

词云如下图所示，词频较高的词语有：“请问”、“买”、“正品”、“味道”、“感觉”、“有没有”等。

整体来看：该类别的问题主要围绕商品本身的类别、特征信息。

首先，这类问题中出现了大量美妆用品（即类别和功能信息），如：“乳液”、“气垫”、“洗面奶”、“遮瑕”、“防晒”、“隔离”、“精华”等等，这些词语很有可能就是商品名称的一部分，同时也解释说明了商品功能。

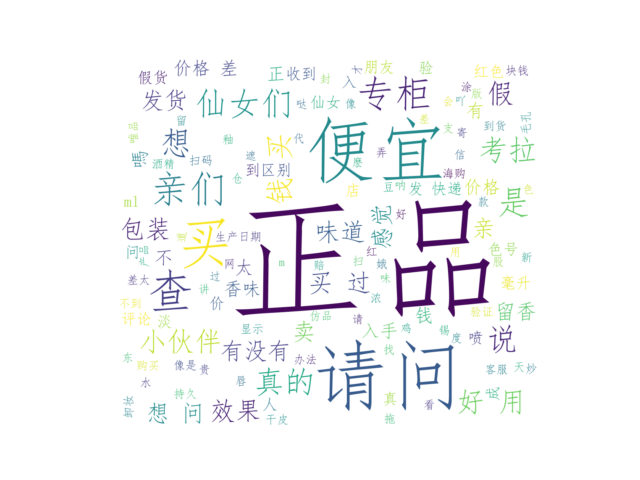
另一方面，这类词语还包含了大量的商品物理特征信息，如：“颜色”、“味道”、“感觉”等。具体而言，与颜色相关的词有：“白色”、“红色”、“哑光”等等；与味道相关的词语有“香”、“留香”、“好闻”等词语；与感觉相关的词语有“持久”、“合适”等。



·类别二：

词云如下图所示，词频较高的词语有：“正品”、“便宜”、“请问”、“买”等。

整体来看：该类别的问题主要围绕商品的质量等信息。产品质量方面，主要有：“正品”、“包装”、“查”、“假货”、“差”等等。



·类别二：

词云如下图所示，词频较高的词语有：“孕妇”、“能用”、“适合”等。

整体来看该类别的问题主要围绕特定人群的使用信息。

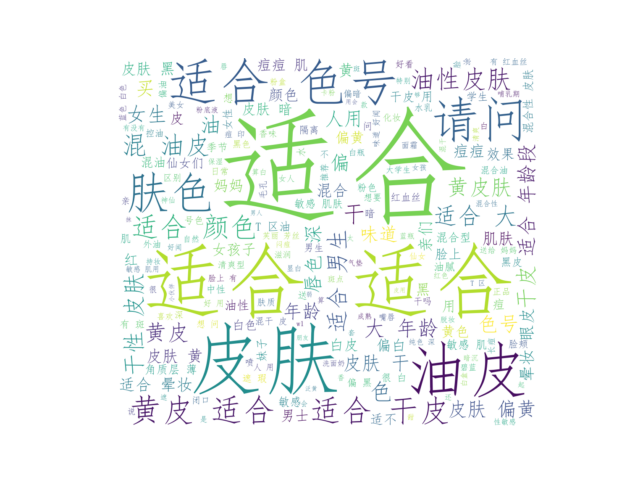


·类别四：

词云如下图所示，词频较高的词语有：“适合”、“油皮”、“色号”、“肤色”等。

整体来看：该类别的问题主要围绕商品的适用情况。

这类问题主要是询问产品可能的适用情况，如“干皮”、“油皮”、“混合型”、“黄皮”等对应“皮肤”；“男生”、“女生”对应性别等。



实验结果总结：

总体来看，这几类中都高频出现了“请问”、“有没有”等词语，主要原因是这些数据据来自提问，所以每个类中均有出现，可考虑屏蔽这类词以提高聚类的准确性。

从各类的结果来看，不同类别的问题有不同的倾向，虽然有一定的交叉，但不影响整体分析。

参考资料

[1] <https://www.tqwba.com/x_d/jishu/16683.html>

[2] <https://www.cnblogs.com/jfdwd/p/9251242.html>

[3] <https://www.cnblogs.com/cy0628/p/13916359.html>

[4] <https://blog.csdn.net/c_living/article/details/85258988>