|  |  |
| --- | --- |
| **自然语言处理及应用**  **实验报告** | |
|  | |
|  |  |
| **名称** | 文本聚类 |
| **姓名** |  |
| **班级** |  |
| **学号** |  |
| Email |  |
| **日期** |  |

# 实验目的

1. 基于机器学习聚类算法实现文本聚类。
2. 提供模型评估结果并进行讨论。

# 实验环境

1. 操作系统：windows 10 专业版。
2. CPU型号：Intel(R) Core(TM) i5-6300 CPU @ 2.30GHz
3. Python版本：python 3.7.0。
4. IDE: Pycharm Community Edition 2020.2.3 x64
5. 内存：8G

# 实验方法

文本聚类总共包括数据预处理、分词、文本向量化表示、聚类、结果评价五个部分。

1. 数据预处理

从www-2.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html处下载文本数据集并解压在项目目录下。遍历文件夹以获取所有的文本种类，并读取对应种类下的文本数据，本次实验所用数据为第二次作业实验的训练集部分。

1. 分词

预处理中得到语料库都是没有分词的原始语料，现在需要将英文句子去除换行符处理为单独的词汇，并且使用nltk提供的英文停用词表对原始数据进行处理，这样可以方便后续的文档结构化表示。数据集分词处理完保存路径为./dataset\_res。

1. 结构化表示-向量空间模型

向量空间模型（VSM）：把对文本内容的处理简化为向量空间中的向量运算，并且它以空间上的相似度表达语义的相似度，直观易懂。当文档被表示为文档空间的向量，就可以通过计算向量之间的相似性来度量文档间的相似性。该阶段可以获得文本词条矩阵，以及文本词汇表。得到的结果存储在。./train\_word\_bag/tfdifspace.dat

1. 文本聚类

本实验采用K-means聚类算法和层次聚类算法进行文本聚类。

K均值聚类（Kmeans）算法作为无监督聚类算法中的代表，该算法的主要作用是将相似的样本自动归到一个类别中。聚类试图将数据集中的样本划分为若干个通常是不相交的子集，每个子集称为一个“簇”，聚类既能作为一个单独过程，用于找寻数据内在的分布结构，也可作为分类等其他学习任务的前驱过程。

Kmeans算法十分简单易懂而且非常有效，但是合理的确定K值和K个初始类簇中心点对于聚类效果的好坏有很大的影响。

主要思想：在给定K值和K个初始类簇中心点的情况下，把每个点(亦即数据记录)分到离其最近的类簇中心点所代表的类簇中，所有点分配完毕之后，根据一个类簇内的所有点重新计算该类簇的中心点(取平均值)，然后再迭代的进行分配点和更新类簇中心点的步骤，直至类簇中心点的变化很小，或者达到指定的迭代次数。

假定给定数据样本X，包含了n个对象其中每个对象都具有m个维度的属性。Kmeans算法的目标是将n个对象依据对象间的相似性聚集到指定的k个类簇中，每个对象属于且仅属于一个其到类簇中心距离最小的类簇中。对于Kmeans，首先需要初始化k个聚类中心，1<k<n，然后通过计算每一个对象到每一个聚类中心的欧式距离，如下式所示：

上式中，表示第i个对象1≤i≤n，表示第j个聚类中心的1≤ j≤k，表示第i个对象的第t个属性1≤t≤m，表示第j个聚类中心的第t个属性。

依次比较每一个对象到每一个聚类中心的距离，将对象分配到距离最近的聚类中心的类簇中，得到k个类簇。

Kmeans算法用中心定义了类簇的原型，类簇中心就是类簇内所有对象在各个维度的均值，其计算公式如下

式中，表示第l个聚类的中心，1≤l≤k，表示第l个类簇中对象的个数，表示第l个类簇中第i个对象，1≤i≤。

算法过程：

选择k个对象作为初始的簇的质心；

repeat

计算对象与各个簇的质心的距离，将对象划分到距离其最近的簇；

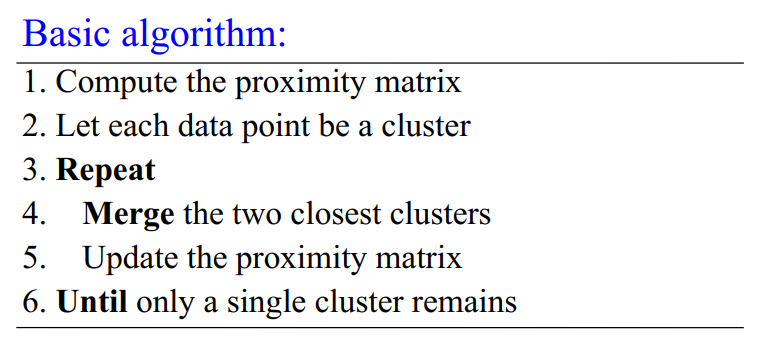
重新计算每个新簇的均值；

until 簇的质心不再变化或达到最大迭代次数。

层次聚类：先计算样本之间的距离。每次将距离最近的点合并到同一个类。然后，再计算类与类之间的距离，将距离最近的类合并为一个大类。不停的合并，直到合成了一个类。其中类与类的距离的计算方法有：最短距离法，最长距离法，中间距离法，类平均法等。比如最短距离法，将类与类的距离定义为类与类之间样本的最短距离。

层次聚类算法根据层次分解的顺序分为：自下向上和自上向下，即凝聚的层次聚类算法和分裂的层次聚类算法（agglomerative和divisive），也可以理解为自下而上法（bottom-up）和自上而下法（top-down）。本实验采用的是凝聚的层次聚类算法。

算法过程：



1. 评价指标

本实验采用轮廓系数以及可视化结果来对最后的聚类结果进行定量与定性的评价。当文本类别未知时，可以选择轮廓系数作为聚类性能的评估指标。轮廓系数取值范围为[-1,1]，取值越接近1轮廓系数越大，表示簇内实例之间紧凑，簇间距离大相反，说明聚类性能越好取值越接近-1则说明聚类性能越差。

1. **实验结果**

在本次实验中，选取的聚类数据为新闻文本数据中的前12类，每类中选取400条数据，实验细节介绍在压缩包中readme.md中。

利用Kmeans聚类算法和层级聚类算法对文本进行聚类，将簇类别个数分别设置为3、4、5、6 … 12、13、14，通过观察轮廓系数来定量来评价聚类的效果。实验的输出结果如图1，图2所示，其中第一行表示聚类的每类的个数，第二行表示聚类轮廓系数。

如图3所示，为两类聚类方法的轮廓系数图，因为数据的实际类别为12类，当聚类个数从3逐渐增长到12时，可以看出，无论是Kmeans聚类方法还是层次聚类方法，模型聚类的效果都在逐渐提升，当聚类个数等于数据的实际类别12时，效果达到最好，但是超过数据的实际类别时，又出现了下降的趋势。

比较层次聚类方法和Kmeans聚类方法可以看到，在本数据集上，Kmeans聚类模型对于数据集的聚类效果在整体上优于层次聚类方法。

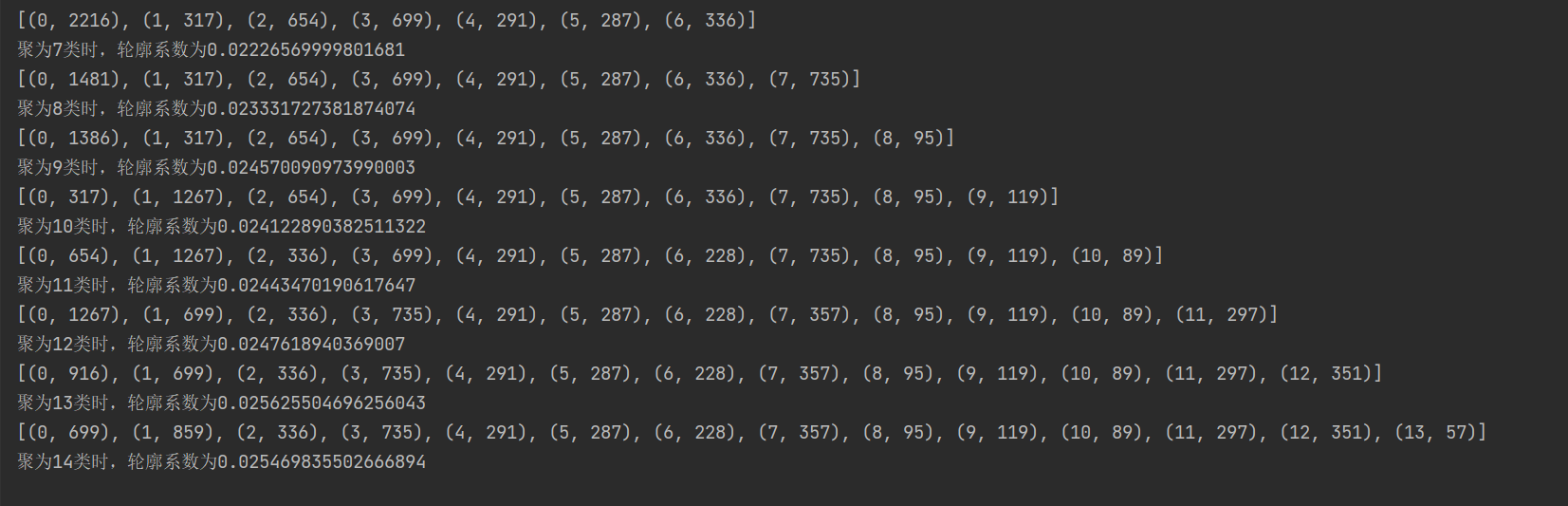


图 1 H\_Clustering运行截图

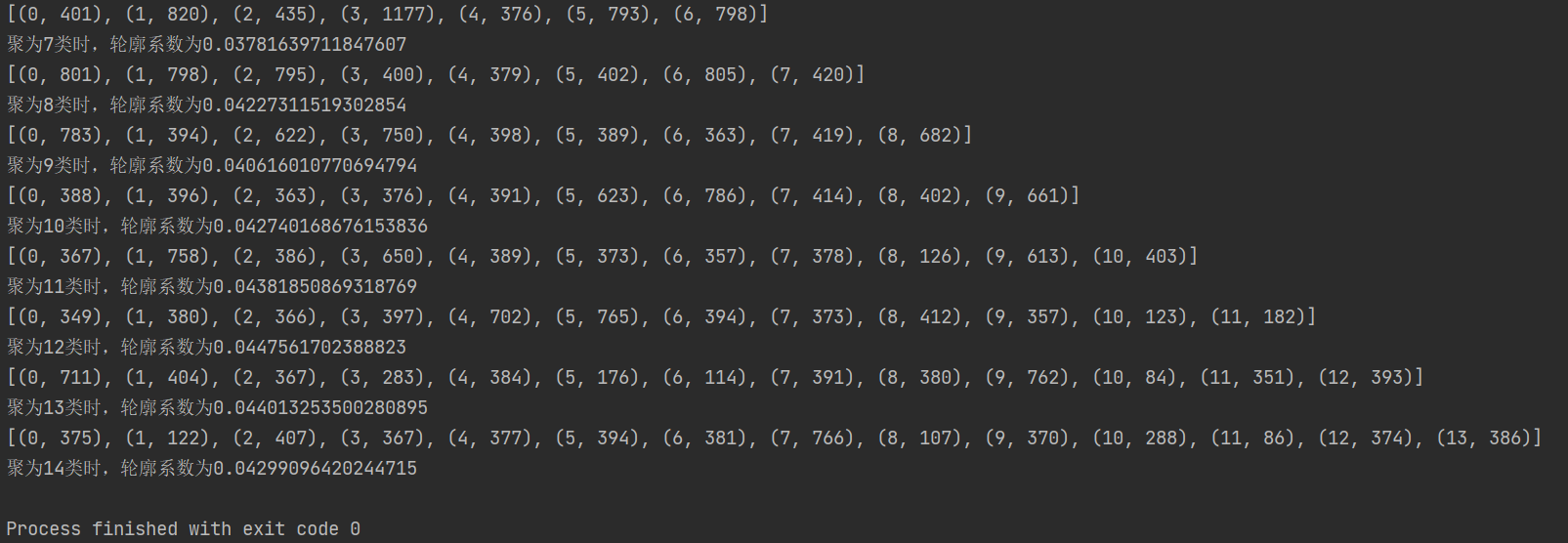


图 2 k\_mean运行截图

图 3 两类层次聚类的轮廓系数

为了更直观的查看层次聚类以及Kmeans聚类的聚类效果，本实验使用tsne降维算法将高维数据降维，并绘制出聚类的散点图，定性评价聚类效果。如图4为数据的初始分布图，图5~图7为kmeans聚类模型的可视化聚类结果，图8~图10为层次聚类模型的可视化聚类结果。从可视化结果可以看出，kmeans模型无论那一个聚簇个数下，在本实验数据下，效果都优于层次聚类模型。

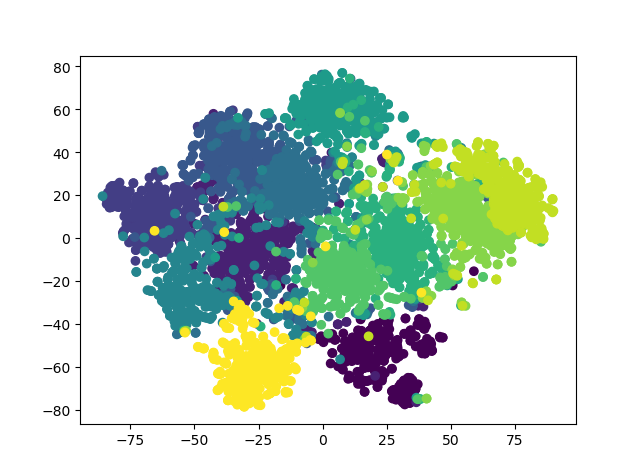


图 4 数据初始分布图

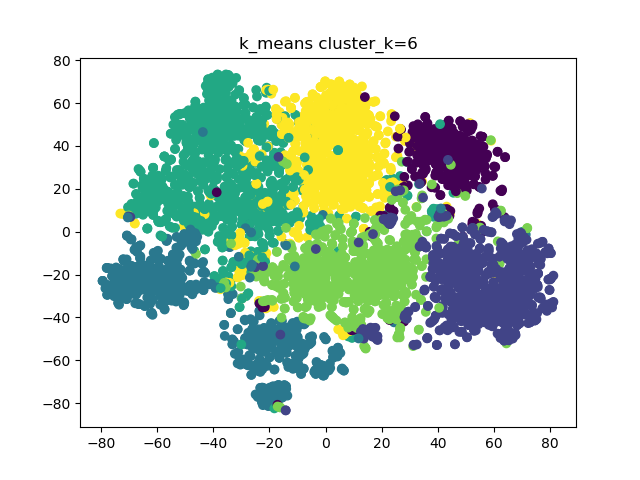


图 5 kmeans聚6类

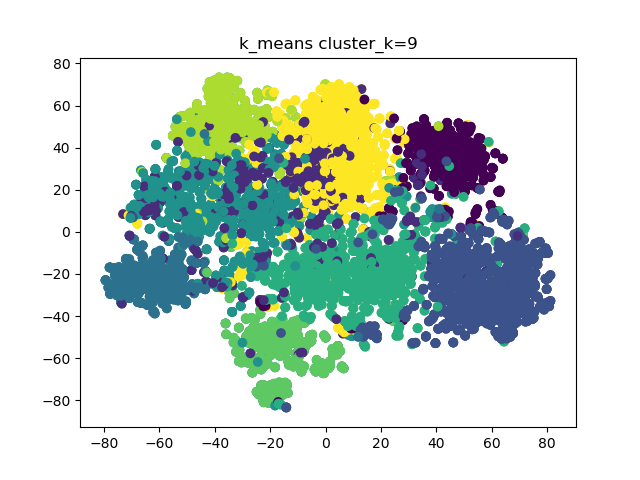


图 6 kmeans聚9类

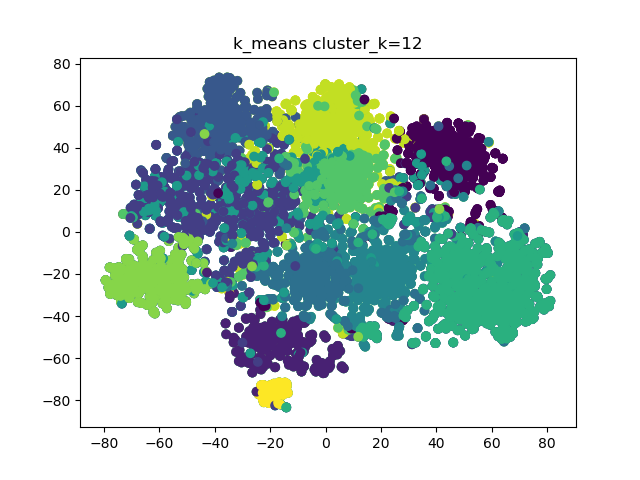


图 7 kmeans聚12类

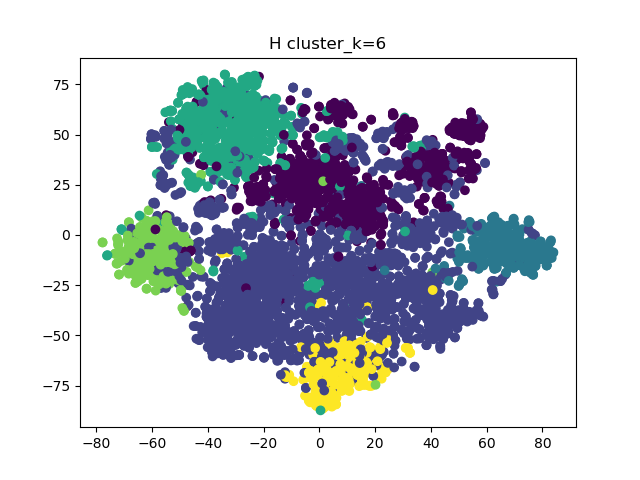


图 8 层次聚类聚6类

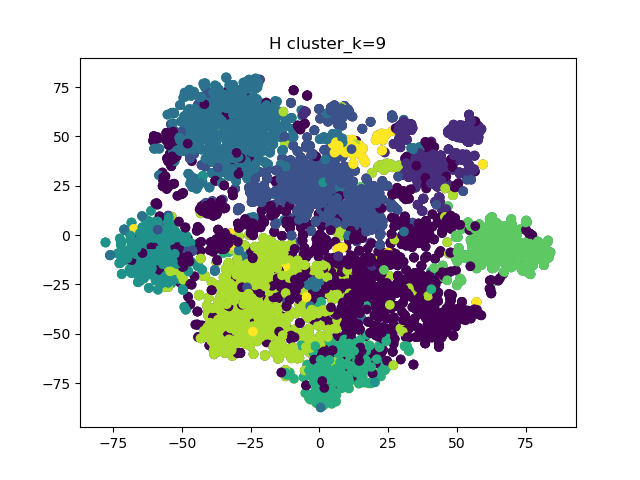


图 9 层次聚类聚9类

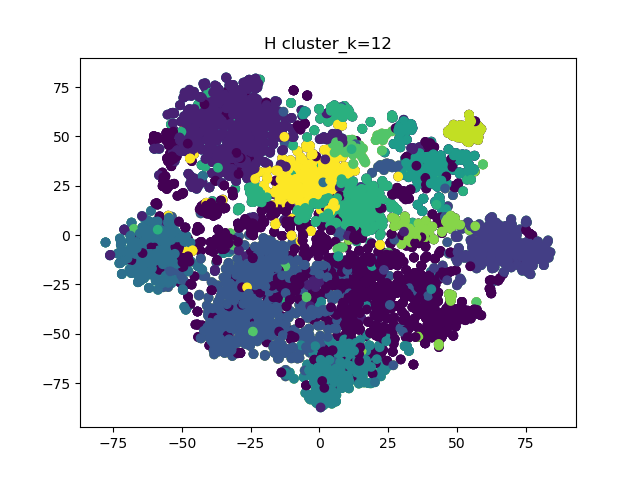


图 10 层次聚类聚12类

1. **遇到问题及解决思路**

1.利用得到向量空间模型进行聚类时，因为选取的特征太多，导致文本词条向量过大，加之，运行环境为个人PC，内存容量较小，每次运行时会出现Memoryerror的问题。解决方法，利用第二次作业的文本聚类模型来寻找较小但又不影响模型效果的特征个数来达到减小内存使用的目的。

2.数据的维数较高，难以直观地观测数据的聚类效果。解决方法，通过使用TSNE模型将数据降维后绘制数据的分布图。