山东大学 计算机科学与技术 学院

信息检索与数据挖掘 课程实验报告

学号：201600130053	姓名：王斌	班级： 16 人工智能

实验题目：预处理文本数据集，并且得到每个文本的 VSM 表示。

实验内容：

#Homework 1: VSM

#预处理文本数据集，并且得到每个文本的 VSM 表示。

The 20 Newsgroups dataset is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.

#20news-18828.tar.gz （http://qwone.com/~jason/20Newsgroups/20news-18828.tar.gz ） ?- 20 Newsgroups; duplicates removed, only "From" and "Subject" headers (18828 documents)

实验环境：

Spyder+python3.6

Win10

实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

一、 各种 sklearn 提供的聚类方法简介：

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

其中各种方法的原理实现可以查看 Sklearn 的官方文档，不再赘诉。链接如下：

https://scikit-learn.org/stable/modules/clustering.html#

二、 对 tweet 数据集的简单处理：

根据每条推特都有很整齐的格式，可以简单处理出需要的 tweet 内容文字和 true_label，具体实现函数如下：

处理后得到

ground_truth #每条推特的正确聚类标签[37,5,8,58.....]

tweets_list #处理过的推特内容列表[推特内容 1，内容 2，.....]。

```
def token(line):
    index = line.index(",")
    Text = line[10:index-1]
    cluNumber = line[index+12:-2]
    return (Text,cluNumber)

def tweets_process():
    global ground_truth,tweets_list
    print("tweets processing...")
    f = open(r"C:\Users\93568\Documents\GitHub\DataMining\work5Clustering with sklearn\data\Homework5Tweets.txt")
    lines = f.readlines()#读取全部内容
    for line in lines:
        (text,cluNumber) = token(line)
        number = int(cluNumber)
        tweets_list.append(text)
        ground_truth.append(number)
```

三、 将 tweet 表示为 tfidf 的矩阵：

利用 python 提供的特征提取的工具包：

```
from sklearn.feature_extraction.text import  TfidfVectorizer

def get_tfidf_matrix():
    global tfidf_matrix,tweets_list
    tfidf_vectorizer = TfidfVectorizer(tokenizer=token_split, lowercase=True)
    '''
    tokenizer: 指定分词函数
    lowercase: 在分词之前将所有的文本转换成小写，因为涉及到中文文本处理，
    所以最好是 False,本 tweet 数据集已经全是小写可设为 True
    '''
    #tfidf_matrix = tfidf_vectorizer.fit_transform(tweets_list)
    #上面一行代码等价于下面两行代码
    tfidf_vectorizer.fit(tweets_list)
    tfidf_matrix = tfidf_vectorizer.transform(tweets_list)
#    joblib.dump(tfidf_matrix, 'tfidf_matrix.pkl')
#    tfidf_matrix = joblib.load('tfidf_matrix.pkl')
```

四、 运行结果及一些问题：

（1） KMeans:

max_iter=200, n_init=20, init='k-means++'：

```
tweets processing...
number of class labels: 89
NMI_score_Kmeans:        0.7918603628391333
```

（2） AffinityPropagation:

```
tweets processing...
number of class labels: 89
(2472, 4640)
NMI_score_AffP:  0.7811293918446975
```

（3） MeanShift:

开始输入相同的 tfidf 矩阵时出现

TypeError: A sparse matrix was passed, but dense data is required. Use X.toarray() to convert to a dense numpy array.（使用 toarray()/todense()后结果如下：）

```
tweets processing...
number of class labels: 89
(2472, 4640)
NMI score meanshift:     -1.6132928326584306e-06
```

可能是密度质心的方法对于高维数据结果很差，输出的标签结果都是 0：

```
[0 0 0 ... 0 0 0]
```

（4） DBSCN:

没有进行调参，默认参数运行结果如下：

```
tweets processing...
number of class labels: 89
(2472, 4640)
NMI score DBSCN:         0.10893421538815534
```

（5） SpectralClustering、ward hierarchical clustering、AgglomerativeClustering、Birch:

```
tweets processing...
number of class labels: 89
(2472, 4640)
NMI_score_SpectralClustering:    0.199231478302035
NMI_score_Ward_hc:               0.7896935252719657
NMI_score_Agg:                   0.19622595222521289
NMI score Birch:                 0.711684280695025
All cluster methods have been done!Great!!!
```

（6） Gaussian mixtures：略…

结论分析与体会：

对 sklearn 的各种聚类方法有了初步的了解，基本掌握了其使用方法。