

数据挖掘实验报告

- 1, VSM and KNN
- 2, NBC
- 3, Clustering with sklearn

姓名: 祝瑶佳

学号: 201834893

VSM and KNN

项目：为文本建立 **VSM** 并使用 **KNN** 进行文本分类

一、实验方法

1. 掌握文本预处理的方法(提取词干，去除停用词等)。
2. 计算每个文档的词频得到词频矩阵。
3. 计算 TF-IDF 值，计算权重，提取关键词。
4. 得到 VSM。
5. 处理文档得到训练集和测试集。
6. 计算欧式距离。
7. 计算 K 个最近邻并合并。
8. 测试 KNN 分类效果。

二、实验任务

1. 预处理文本数据集，并且得到每个文本的 VSM 表示
2. 实现 KNN 分类器，测试其在 20Newsgroups 上的效果。

三、实验数据

20 Newsgroups

四、实验步骤

1，文本处理

(1) 分词

读取文档按空格分词，并且去掉符号。将文档划分成单词，并对单词做一些处理：大写字母变成小写字母，名词复数变单数，去掉停用词，各种时态和形式的动词变成原形，只保留英文词干部分。

(2) 划分训练集和测试集

将数据集划分成训练集和测试集，其中训练集占 80%，测试集占 20%。

(3) 创建词典

从训练集中读取所有的文档，统计所有的单词及词频，计算 TF-IDF 的值，提取关键词，并创建字典。

(4) 得到文本的向量表示 VSM。

2, 使用 KNN 进行文本分类

将数据集划分为训练集和数据集, 计算每一个测试实例到训练集实例的欧式距离, 对所有距离进行排序, 得到 K 个最近邻。对最近邻进行合并排序, 最后测试分类准确度。

五、实验结果

结果如下图: 数据集的词频矩阵:

```
d1 = [2, 0, 4, 3, 0, 1, 0, 2]
d2 = [0, 2, 4, 0, 2, 3, 0, 0]
d3 = [4, 0, 1, 3, 0, 1, 0, 1]
d4 = [0, 1, 0, 2, 0, 0, 1, 0]
d5 = [0, 0, 2, 0, 0, 4, 0, 0]
d6 = [1, 1, 0, 2, 0, 1, 1, 3]
d7 = [2, 1, 3, 4, 0, 2, 0, 2]
```

TF-IDF:

```
idf:
[0.24303804868629444, 0.24303804868629444, 0.146128035678238, 0.146128035678238, 0.8450980400142568, 0.066946789630
61322, 0.5440680443502757, 0.24303804868629444]
```

```
>>> distances # 对应的距离
array([[ 13.37908816,  13.60147051,  13.60147051,  13.60147051,
         13.60147051,  13.6381817 ]])
```

六、实验结论和感想

本实验中, k 值的选取不同, 准确率也存在差别, 可以多尝试下不同的 K 值, 找到使得准确率最高的 K 值。

不管是在 VSM 模型的建立中, 还是 KNN 分类中, 由于数据量很大, 会发生内存溢出现象。

NBC

项目：使用朴素贝叶斯分类器分类文档

一、实验方法

1. 将文档分成训练集和测试集
2. 将训练集进行分词处理(去除重复词，去掉符号)得到词典
3. 将每个类的所有文档分词得到词典
4. 将训练集得到的词典与每个类的词典进行比较，如果训练词典中的词在类词典中出现，则记录出现次数，没出现则记为 0，得到训练集对应每个类词频向量
5. 统计每个词在每个类出现的次数和此类中所有词出现的次数，计算得到 $p(\text{每个词}|\text{每个类})$
6. 将测试集文档进行分词处理并得到词典。
7. 比较测试集词典和训练集词典，如果有相同的词，则对应词的 $p(\text{每个词}|\text{每个类})$ 可以从训练集中沿用，可以放入测试集概率向量列表中。
8. 将测试集中词典的所有 $p(\text{每个词}|\text{每个类})$ 相乘再乘以类概率 p_{class} 得到该文档属于此类的概率，依照此方法计算出每个测试文档属于各个类的概率
9. 比较每个训练文档属于各个类的概率，找到最大的概率，并将文档归于相应的类。

二、实验任务

- 1, 实现朴素贝叶斯分类器，测试其在 20 Newsgroups 数据集上的效果

三、实验数据

20 Newsgroups

四、实验步骤

1, 文本处理

- (1) 分词并创建词典

读取文档按空格分词，并且去掉符号和重复。将文档划分成单词，

- (2) 划分训练集和测试集

将数据集划分成训练集和测试集，从每个类抽取 80%为训练集，20%为测试集，组成最终的训练集和测试集。

- (3) 得到训练集对应每个类词频向量

2, 训练贝叶斯分类器

- (1) 得到训练集的每个词属于每个类的概率向量

3，测试贝叶斯分类器

- (1) 根据训练效果得到测试集每个词属于每个类的概率向量
- (2) 得到每个文档属于每个类的概率并选择概率最大的，将文档归于此类。

五、实验结果

训练集的概率向量：

```
C:\Users\Embedded\Desktop\homework2\venv\Scripts\python.exe C:/Users/Embedded/Desktop/homework2/venv/NB.py
1. 04551186327e-07

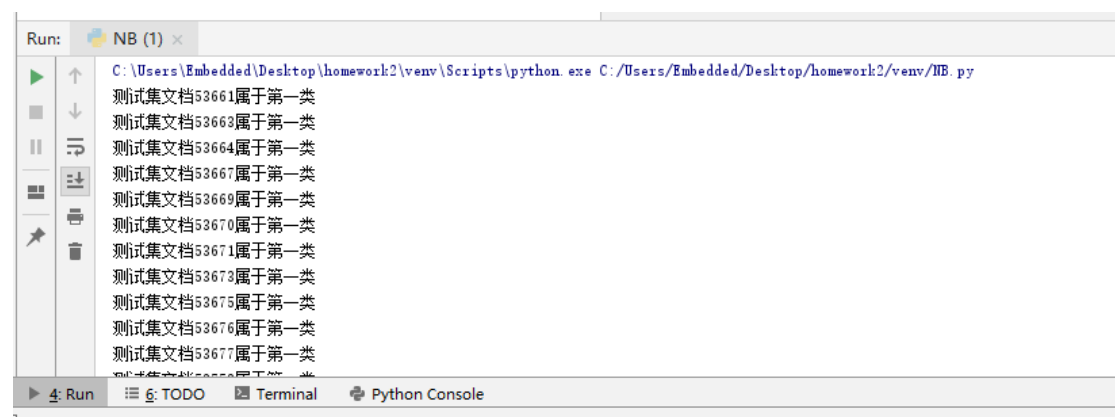
2. 09102372653e-07

8. 36409490613e-07

1. 25461423592e-06

1. 35916542225e-06
```

分类结果：



六、实验结论和感想

词频向量是统计每个词在每个类的所有文档中出现的次数。

概率向量是统计每个词在每个类的所有文档中出现的概率。

贝叶斯分类器的特点就是简单实用，分类效果也不错。

Clustering with sklearn

项目：测试 sklearn 中聚类算法在 tweets 数据集上的聚类效果。并使用 NMI(Normalized Mutual Information)作为评价指标。

一,实验方法

10. 将文档分词，并记录真实类号和类总数(89 个类)。
11. 计算所得分词的 TF-IDF 的值获得词权重
12. 调用 KMens 方法，返回预测的聚类结果
13. 使用 NMI 评价标准，将真实聚类结果与预测值作比较，得到 NMI 值。
- 5, 调用 Affinity propagation 方法返回聚类结果并计算 NMI 值
- 6, 调用 Mean-shift 方法返回聚类结果并计算 NMI 值
- 7, 调用 Spectral clustering 方法返回聚类结果并计算 NMI 值
- 8, 调用 Ward hierarchical clustering 方法返回聚类结果并计算 NMI 值
- 9, 调用 Agglomerative clustering 方法返回聚类结果并计算 NMI 值
- 10, 调用 DBSCAN 方法返回聚类结果并计算 NMI 值
- 11, 调用 Gaussian mixtures 方法返回聚类结果并计算 NMI 值

二,实验任务

- 1, 测试 sklearn 中聚类算法在 tweets 数据集上的聚类效果。并使用 NMI(Normalized Mutual Information)作为评价指标。

三,实验数据

Tweets.txt

四,实验步骤

1, 文本处理

- (1) 将文档分词(可根据标点符号分)，并记录真实类号和类总数(89 个类),
- (2) 计算所得分词的 TF-IDF 的值获得词权重

2, 分别调用 8 种聚类方法(89 个类，与真实类数相符合)

- (1)得到预测聚类结果

3, 使用 NMI 评价指标分别评价 8 次不同聚类方法所得到的聚类效果

- (1)输入预测聚类结果和真实类结果，得到 NMI 的值，NMI 的值越高，说明聚类效果越好。

五,实验结果

分词的 TF-IDF 矩阵:

| Run: clustering x clustering x clustering x | | | | | | | |
|---|-----------|----------|----|----------------|------------|----|------------|
| ▶ | ↑ | 0. | 0. | 0. | 0. | 0. | 0. |
| ■ | ↓ | 0. | 0. | 0. | 0. | 0. | 0. |
| | ↺ | 0. | 0. | 0. | 0. | 0. | 0. |
| | ↻ | 0. | 0. | 0. | 0. | 0. | 0. |
| ☐ | ↕ | 0. | 0. | 0. | 0. | 0. | 0. |
| ☐ | ↔ | 0. | 0. | 0. | 0.49192244 | 0. | 0. |
| ✦ | 🗑 | 0. | 0. | 0. | 0. | 0. | 0. |
| | | 0. | 0. | 0. | 0. | 0. | 0. |
| | | 0. | 0. | 0. | 0. | 0. | 0.56834702 |
| | | 0. | 0. | 0. | 0. | 0. | 0. |
| ▶ 4: Run | ☰ 6: TODO | Terminal | | Python Console | | | |

调用 KMeans 后的聚类效果:

| | | | | | | | | | | | | | | | for i in range(str_num) | | | | | | | | | | | | | | | | | | |
|--|---|--------------|---------|--|----------|--|----------------|--|--|--|--|--|--|--|-------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Run: | | clustering x | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div><div>▶</div><div>■</div><div> </div><div>☐</div><div>✦</div></div> <div><div>↑</div><div>↓</div><div>↺</div><div>↻</div><div>↕</div><div>↔</div><div>🗑</div><div>🗑</div></div> | <div>58 10 4 3 37 32 52 37 63 30 29 76 0 33 37 3 63 0 13 82 30 17 33 60</div> <div>74 38 41 88 10 50 24 22 0 64 4 22 88 76 4 3 66 3 29 1 64 16 64 0</div> <div>58 39 7 64 9 7 54 78 73 43 13 5 41 83 7 57 9 14 53 65 31 13 78 76</div> <div>58 65 79 55 27 8 75 23 80 11 10 23 53 13 18 76 76 19 66 55 4 45 18 76</div> <div>45 3 22 10 9 3 59 43 78 3 75 3 51 4 58 16 12 13 70 58 7 53 72 3</div> <div>4 72 14 4 3 37 4 12 14 28 13 4 4 77 4 34 22 60 77 24 72 13 17 58</div> <div>85 73 71 64 49 49 3 9 66 22 4 4 19 48 49 75 3 71 52 71 27 53 4 12</div> <div>3 23 3 3 3 28 30 43 38 4 3 58 43 27 87 51 88 4 3 62 65 76 77 37</div> <div>58 66 60 60 29 34 22 37 51 73 76 3 17 24 39 24 79 76 28 88 85 3 3 26</div> <div>25 77 34 48 32 23 17 53 5 51 5 4 4 10 73 28 56 3 77 66 4 79 41 47</div> <div>48 7 79 25 4 68 23 3 24 77 2 17 4 77 15 32 6 47 13 21 4 58 82 81</div> <div>46 65 31 28 30 47 74 17 76 42 30 3 87 67 39 1 86 13 22 24 61 64 52 88]</div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 4: Run | | 6: TODO | | Terminal | | Python Console | | | | | | | | | | | | | | | | | | | | | | | | | | |

NMI 评价标准的值:

```
Run: clustering x
3 23 3 3 3 28 30 43 38 4 3 58 43 27 87 51 88 4 3 62 65 76 77 37
58 66 60 60 29 34 22 37 51 73 76 3 17 24 39 24 79 76 28 88 85 3 3 26
25 77 34 48 32 23 17 53 5 51 5 4 4 10 73 28 56 3 77 66 4 79 41 47
48 7 79 25 4 68 23 3 24 77 2 17 4 77 15 32 6 47 13 21 4 58 82 81
46 65 31 28 30 47 74 17 76 42 30 3 87 67 39 1 86 13 22 24 61 64 52 88]
C:\Users\Embedded\Desktop\homework3\venv\lib\site-packages\sklearn\metrics\clu:
FutureWarning)
0.6870340799488106

Process finished with exit code 0
```

使用 Affinity propagation 的聚类结果

```
Run: clustering (1) x
165 143 34 224 301 143 37 315 301 323 35 30 299 184 224 302 193 234
284 144 303 257 196 303 69 165 133 268 212 60 190 83 303 91 42 187
304 175 233 261 62 303 284 186 305 24 320 160 306 177 88 231 307 229
304 261 308 303 265 96 35 243 253 241 309 194 310 24 294 65 119 12
95 268 159 291 239 197 323 302 39 311 292 313 158 39 303 304 312 264
213 312 235 17 291 84 166 292 235 326 313 213 314 315 87 237 143 108
194 167 312 313 24 36 268 133 93 23 71 71 58 119 196 289 316 316
98 242 317 239 64 5 318 319 320 304 282 292 335 288 136 321 322 254
151 48 51 87 335 4 48 10 156 323 298 17 298 15 186 209 298 324
4 35 108 108 299 63 143 84 188 185 203 242 102 167 144 167 305 211
54 115 195 103 95 81 208 330 22 72 221 43 49 277 123 54 60 107
173 325 185 326 8 280 298 327 294 305 328 182 234 303 86 208 61 84
```

NMI 评价标准的值:

```
clustering (1) x
C:\Users\Embedded\Desktop\homework3\venv\Scripts\python.exe C:/Users/Embedded/Desktop/homework3/clustering.py
C:\Users\Embedded\Desktop\homework3\venv\lib\site-packages\sklearn\metrics\cluster\supervised.py:732: FutureWarning:
FutureWarning)
0.4812428226957844

Process finished with exit code 0
```

使用 Agglomerative clustering 的聚类结果及 NMI 值


```

clustering (1) ×
2 2 35 6 39 2 2 12 1 5 26 2 32 3 22 2 72 29 8 22 78 0 45 4
3 45 85 3 5 4 3 72 85 65 29 3 2 68 3 34 19 9 77 1 45 29 16 22
63 2 2 62 42 42 40 39 0 19 3 3 2 2 42 2 2 2 44 2 64 0 3 72
2 13 7 2 2 18 17 12 28 3 2 22 63 2 53 32 7 3 14 17 57 7 29 4
22 39 82 82 59 48 19 4 52 73 7 2 2 1 70 1 36 23 65 5 63 29 20 79
56 28 2 3 30 13 16 0 16 52 25 19 3 6 73 65 37 2 28 26 3 2 10 3
43 68 2 56 3 4 80 2 1 78 31 16 2 28 2 30 61 35 29 4 3 22 55 11
53 57 67 65 17 3 3 16 23 2 17 2 53 24 70 13 7 29 19 1 48 62 44 8]
C:\Users\Embedded\Desktop\homework3\venv\lib\site-packages\sklearn\metrics\cluste
FutureWarning)
0.6948668794233029

```

使用 Spectral clustering 的聚类结果及 NMI 值

```

un: clustering (1) ×
60 81 20 0 5 11 0 48 28 28 87 13 34 8 13 31 40 27 88 10 12 43 33 84
54 49 46 36 7 26 42 20 0 64 54 20 36 69 54 58 70 51 87 9 64 1 50 45
17 4 29 50 70 29 39 42 0 23 33 62 46 61 4 48 84 82 24 56 67 86 42 69
17 28 25 72 35 0 70 9 54 32 7 9 24 86 41 69 69 88 70 62 54 51 41 18
3 81 15 7 70 0 14 23 42 62 59 0 58 54 17 0 59 86 36 17 29 24 55 52
54 55 31 54 62 52 54 59 31 45 86 54 59 47 33 80 20 65 18 42 55 40 72 17
75 78 0 64 37 63 0 70 70 20 54 20 88 82 37 15 0 0 0 0 35 24 54 59
0 9 20 27 66 45 12 23 49 33 0 17 23 68 34 58 36 54 6 12 28 69 86 52
17 70 65 65 87 80 20 52 71 63 69 82 84 42 4 42 25 1 45 36 75 40 14 13
77 49 80 54 76 9 43 6 43 71 62 20 54 7 63 45 79 58 49 70 54 25 46 19
59 29 25 77 54 52 57 73 42 29 53 43 54 49 74 52 27 19 86 30 54 17 10 16
34 28 67 45 12 19 54 43 47 3 12 0 34 83 4 9 69 86 20 42 35 64 0 36]

clustering (1) ×
C:\Users\Embedded\Desktop\homework3\venv\Scripts\python.exe C:/Users/Embedded/Desktop/homework3/clustering.py
C:\Users\Embedded\Desktop\homework3\venv\lib\site-packages\sklearn\metrics\cluster\supervised.py:732: FutureWarn
FutureWarning)
0.6682208437214124

Process finished with exit code 0

```

使用 Ward hierarchical clustering 的聚类结果及 NMI 值

| clustering (1) × | | | |
|------------------|----|----|----|
| 22 | 9 | 2 | 2 |
| 3 | 7 | 10 | 65 |
| 22 | 7 | 78 | 58 |
| 22 | 22 | 36 | 16 |
| 2 | 2 | 35 | 6 |
| 3 | 45 | 85 | 3 |
| 63 | 2 | 2 | 62 |
| 2 | 13 | 7 | 2 |
| 22 | 39 | 82 | 82 |
| 56 | 28 | 2 | 3 |
| 43 | 68 | 2 | 56 |
| 53 | 57 | 67 | 65 |

```

clustering (1) ×
C:\Users\Embedded\Desktop\homework3\venv\Scripts\python.exe C:/Users/Embedded/Desktop/hor
C:\Users\Embedded\Desktop\homework3\venv\lib\site-packages\sklearn\metrics\cluster\super
FutureWarning)
0.6948668794233029

Process finished with exit code 0

```

使用 Gaussian mixtures 的聚类结果及 NMI 值

```

7 4 4 4 4 7 2 2 4 4 0 4 4 0 4 4 0 4 4 4 4 0 0 1 0 4 4 4 1 4 4 4 5 7 4 4 4
4 4 4 4 2 2 4 0 0 4 4 4 4 4 4 4 4 4 2 0 4 4 1 4 4 4 4 4 2 0 4 7 4 4 4 4
1 0 4 4 7 2 1 4 7 2 4 4 4 4 4 4 4 6 4 2 4 5 4 2 4 4 2 4 4 4 4 4 4 4 1 4 0 3
1 5 2 5 6 5 0 0 3 4 4 4 4 1 2 0 4 4 0 4 2 4 4 0 4 1 4 4 1 2 5 0 1 4 4 4 0
4 4 0 7 4 4 4 7 6 4 4 0 0 5 2 4 4 4 4 4 2 1 2 4 4 4 4 4 4 4]
C:\Users\Embedded\Desktop\homework3\venv\lib\site-packages\sklearn\metrics\cluster\super
FutureWarning)
0.29211190255507585

```

六,实验结论和感想

Sklearn 是一个很好的工具，对于一些聚类分类算法，直接调用就可以节省很多时间，但是对于同一个文本，不同的聚类方法效果也不同，有的比较好，有的比较差，而且一些聚类方法比如高斯方法，如果聚类个数增多，NMI 值会提升，但是如果太多就会报错无法聚类。