报告

学号:3019213043　　　姓名:刘京宗　　　班级:软工 5 班

1. 目标

练习如何进行聚类、聚类分析和聚类可视化。熟悉开发系统、其图形用户界面和输入数据格式。

2. 数据

数据集是 GSE7390_transbig2006affy_demo.csv。其中包含了 TRANSBIG 验证研究中 198 名未治疗患者的信息。请参阅介绍数据集的 README.txt 文件。

3. 实验

任务 1

1) 预处理

a) 认识你的数据。数据中有多少个实例、名义属性、数字属性？也就是要熟悉数据集的不同属性，它们的分布情况。用表格列出数值属性的范围。是否有一些数值属性没有用？如果有，就删除这个属性。

  i. 数据中有 198 个实例，共有 28 个属性，其中 samplename,id,geo,filename,hosipital,Surgery_type,Histtype,Angioinv,Lymp_infil,node,grade,er,e.rfs,e.os,e.dmfs,e.tdm,risksg,risknpi,risk_AOL,veridex_risk 20 个属性是名义属性 age,size,t.rfs,t.os,t.dmfs,t.tdm,NPI,AOL_os_10y 8 个属性是数值属性。

  ii. 8 个数值属性均是有用的，不删除。

b) 有些名义属性有太多不同的值。它们不包含有用的信息。将它们从考虑中删除，以创建一个新的.arff 数据集文件。列出被删除

属性的名称。

i. samplename,id,geo,filename,hosipital 包含的属性与实验无关，选择删去，node 属性对于所有数据只有 1 个值，也选择删去。

c) 规范化数据，创建一个新的.arff 数据集文件。在你的.arff 文件中列出数据集的前 10 个数据实例。

@data

i. 0.916667,0.545455,0,0.111111,0.333333,0.5,1,0,0.070081,1,0.088262,1,0.06657,1,0.06657,1,Poor,0.859155,Poor,0.076503,Poor,Poor

ii. 0.916667,0.545455,0,0.222222,0.333333,1,1,1,0.007218,1,0.719148,0,0.719804,0,0.719804,0,Poor,0.859155,Poor,0.248634,Poor,Poor

iii. 0.666667,0.431818,1,0.111111,0,0.5,1,0,0.046915,1,0.086588,1,0.044417,1,0.044417,1,Poor,0.823944,Poor,0.172131,Poor,Poor

iv. 0.5,0.272727,0,0.111111,0.333333,1,1,1,0.241094,1,0.681656,1,0.6824,1,0.6824,0,Poor,0.774648,Poor,0.68306,Poor,Poor

v. 0.611111,0.545455,0,0.111111,0.333333,0.5,0.5,1,0.43085,1,0.444878,1,0.411555,1,0.411555,1,Poor,0.507042,Poor,0.554645,Poor,Poor

vi. 0.944444,0.318182,0,0.222222,0.333333,0.5,0.5,1,0.743423,0,0.709775,0,0.710453,0,0.710453,0,Poor,0.43662,Poor,0.63388,Poor,Good

vii. 0.555556,0.318182,1,0.111111,0.333333,1,1,0,0.068452,1,0.647289,0,0.648113,0,0.648113,0,Poor,0.788732,Poor,0.565574,Poor,Poor

viii. 0.944444,0.431818,1,0.222222,0.333333,0.5,0,1,0.66298,1,0.632671,1,0.63353,1,0.63353,0,Poor,0.119718,Good,0.606557,Poor,Good

ix. 0.638889,0.545455,0,0.222222,0.666667,?,1,1,0.685215,0,0.653983,0,0.654792,0,0.654792,0,Poor,0.859155,Poor,0.363388,Poor,Poor

x. 0.388889,0.431818,0,0.222222,0.666667,1,0.5,1,0.035041,1,0.149297,1,0.123344,1,0.123344,1,Poor,0.471831,Poor,0.590164,Poor,Poor

2) K-means

a) 设定不同的 k: k = 2 to 6,

b) 使用不同的距离指标: Euclidean and Manhattan (= cityblock).

c) 使用不同的种子值: 10, 27, 43;

d) 在聚类前对每个属性进行标准化和不进行标准化.

使用一个表格来形成每个实验及其结果的摘要：使用的参数，每个聚类中的实例数，中心点，以及每个聚类的误差值（"聚类内平方误差之和"）。你对每个聚类有什么有趣的发现或观察。

下表中 E 表示选择 Euclidean 距离，M 表示选择 Manhattan 距离。

T 表示数据进行了标准化，F 表示没有进行标准化。

k	距离	种子	标准化	实例数	聚类中心点	误差
2	E	10	T	0 55 (28%) 1 143 (72%)	Cluster 0: 0.527778,0.431818,0,0.111111,0.333333,0,0,1,0.15227,1,0.605334,0,0.606256,0,0.606256,0,Poor,0.119718,Good,0.786885,Good,Good Cluster 1: 0.75,0.272727,0,0.111111,0.333333,0.5,0.5,1,0.552037,0,0.526333,0,0.527441,0,0.527441,0,Poor,0.422535,Poor,0.745902,Poor,Poor	437.1647026
3	E	10	T	0 40 (20%) 1 101 (51%) 2 57 (29%)	Cluster 0: 0.527778,0.431818,0,0.111111,0.333333,0,0,1,0.15227,1,0.605334,0,0.606256,0,0.606256,0,Poor,0.119718,Good,0.786885,Good,Good Cluster 1: 0.75,0.272727,0,0.111111,0.333333,0.5,0.5,1,0.552037,0,0.526333,0,0.527441,0,0.527441,0,Poor,0.422535,Poor,0.745902,Poor,Poor Cluster 2: 0.388889,0.431818,0,0.222222,0.666667,1,0.5,1,0.035041,1,0.149297,1,0.123344,1,0.123344,1,Poor,0.471831,Poor,0.590164,Poor,Poor	316.7792652

4	E	10	T	0 39 (20%) 1 67 (34%) 2 57 (29%) 3 35 (18%)	Cluster 0: 0.527778,0.431818,0,0.111111,0.333333,0,0,1,0.15227,1,0.605334,0,0.606256,0,0.606256,0,Poor,0.119718,Good,0.786885,Good,Good Cluster 1: 0.75,0.272727,0,0.111111,0.333333,0.5,0.5,1,0.552037,0,0.526333,0,0.527441,0,0.527441,0,Poor,0.422535,Poor,0.745902,Poor,Poor Cluster 2: 0.388889,0.431818,0,0.222222,0.666667,1,0.5,1,0.035041,1,0.149297,1,0.123344,1,0.123344,1,Poor,0.471831,Poor,0.590164,Poor,Poor Cluster 3: 0.638889,0.431818,1,0.111111,0,0.444444,1,0,0.508265,0,0.484378,0,0.485584,0,0.485584,0,Poor,0.823944,Poor,0.177596,Poor,Poor	288.7048550560438
5	E	10	T	0 39 (20%) 1 46 (23%) 2 56 (28%) 3 33 (17%) 4 24 (12%)	Cluster 0: 0.527778,0.431818,0,0.111111,0.333333,0,0,1,0.15227,1,0.605334,0,0.606256,0,0.606256,0,Poor,0.119718,Good,0.786885,Good,Good Cluster 1: 0.75,0.272727,0,0.111111,0.333333,0.5,0.5,1,0.552037,0,0.526333,0,0.527441,0,0.527441,0,Poor,0.422535,Poor,0.745902,Poor,Poor Cluster 2: 0.388889,0.431818,0,0.222222,0.666667,1,0.5,1,0.035041,1,0.149297,1,0.123344,1,0.123344,1,Poor,0.471831,Poor,0.590164,Poor,Poor Cluster 3: 0.638889,0.431818,1,0.111111,0,0.444444,1,0,0.508265,0,0.484378,0,0.485584,0,0.485584,0,Poor,0.823944,Poor,0.177596,Poor,Poor Cluster 4: 0.583333,0.431818,0,0.111111,0.305785,0.444444,0.5,1,0.196973,1,0.57688,0,0.577869,0,0.577869,0,Poor,0.471831,Poor,0.560109,Poor,Poor	271.5590475

6	E	10	T	0 39 (20%) 1 46 (23%) 2 36 (18%) 3 33 (17%) 4 24 (12%) 5 20 (10%)	Cluster 0: 0.527778,0.431818,0,0.111111,0.333333,0,0,1,0.15227,1,0.605334,0,0.606256,0,0.606256,0,Poor,0.119718,Good,0.786885,Good,Good Cluster 1: 0.75,0.272727,0,0.111111,0.333333,0.5,0.5,1,0.552037,0,0.526333,0,0.527441,0,0.527441,0,Poor,0.422535,Poor,0.745902,Poor,Poor Cluster 2: 0.388889,0.431818,0,0.222222,0.666667,1,0.5,1,0.035041,1,0.149297,1,0.123344,1,0.123344,1,Poor,0.471831,Poor,0.590164,Poor,Poor Cluster 3: 0.638889,0.431818,1,0.111111,0,0.444444,1,0,0.508265,0,0.484378,0,0.485584,0,0.485584,0,Poor,0.823944,Poor,0.177596,Poor,Poor Cluster 4: 0.583333,0.431818,0,0.111111,0.305785,0.444444,0.5,1,0.196973,1,0.57688,0,0.577869,0,0.577869,0,Poor,0.471831,Poor,0.560109,Poor,Poor Cluster 5: 0.861111,0.431818,1,0.111111,1,0.5,0.5,0,0.049709,1,0.221268,1,0.180118,1,0.180118,1,Poor,0.471831,Poor,0.153005,Poor,Poor	257.6163398
4	M	10	T	0 40 (20%) 1 64 (32%) 2 54 (27%) 3 40 (20%)	Cluster 0: 0.527778,0.431818,0,0.111111,0.333333,0,0,1,0.15227,1,0.605334,0,0.606256,0,0.606256,0,Poor,0.119718,Good,0.786885,Good,Good Cluster 1: 0.75,0.272727,0,0.111111,0.333333,0.5,0.5,1,0.552037,0,0.526333,0,0.527441,0,0.527441,0,Poor,0.422535,Poor,0.745902,Poor,Poor Cluster 2: 0.388889,0.431818,0,0.222222,0.666667,1,0.5,1,0.035041,1,0.149297,1,0.123344,1,0.123344,1,Poor,0.471831,Poor,0.590164,Poor,Poor Cluster 3: 0.638889,0.431818,1,0.111111,0,0.444444,1,0,0.508265,0,0.484378,0,0.485584,0,0.485584,0,Poor,0.823944,Poor,0.177596,Poor,Poor	554.6217777

3	M	10	T	0 41 (21%) 1 100 (51%) 2 57 (29%)	Cluster 0: 0.527778,0.431818,0,0.111111,0.333333,0,0,1,0.15227,1,0.605334,0,0.606256,0,0.606256,0,Poor,0.119718,Good,0.786885,Good,Good Cluster 1: 0.75,0.272727,0,0.111111,0.333333,0.5,0.5,1,0.552037,0,0.526333,0,0.527441,0,0.527441,0,Poor,0.422535,Poor,0.745902,Poor,Poor Cluster 2: 0.388889,0.431818,0,0.222222,0.666667,1,0.5,1,0.035041,1,0.149297,1,0.123344,1,0.123344,1,Poor,0.471831,Poor,0.590164,Poor,Poor	608.5459698
3	E	27	T	0 58 (29%) 1 90 (45%) 2 50 (25%)	Cluster 0: 0.666667,0.590909,0,0.111111,0,0.444444,1,0,0.321071,0,0.304954,0,0.306579,0,0.306579,0,Poor,0.873239,Poor,0.065574,Poor,Good Cluster 1: 0.25,0.5,1,0.166667,0.305785,0.444444,1,0,1,0,0.955702,0,0.955805,0,0.955805,0,Poor,0.84507,Poor,0.224044,Poor,Good Cluster 2: 0.5,0.454545,0,0.222222,0,0.444444,0,1,0.492666,1,0.591274,1,0.470667,1,0.470667,1,Poor,0.126761,Good,0.79235,Good,Good	314.6373435
3	E	43	T	0 101 (51%) 1 40 (20%) 2 57 (29%)	Cluster 0: 0.5,0.318182,0,0.333333,0.305785,0.444444,0.5,1,0.668102,0,0.637581,0,0.638428,0,0.638428,0,Poor,0.43662,Poor,0.81694,Good,Poor Cluster 1: 0.722222,0.25,0,0.111111,0.333333,0.5,0,1,0.559255,0,0.533252,0,0.534343,0,0.534343,0,Good,0.06338,Good,0.887978,Good,Good Cluster 2: 0.305556,0.590909,1,0.888889,0.333333,0,0.5,0,0.677998,0,0.647065,0,0.64789,0,0.64789,0,Poor,0.521127,Poor,0.169399,Poor,Good	316.7792652
3	E	10	F	0 40 (20%) 1 101 (51%) 2	Cluster 0: 43,2.5,0,1,1,1,1,1,1429,1,5571,0,5571,0,5571,0,Poor,2.5,Good,88.7,Good,Good Cluster 1: 51,1.8,0,1,1,2,2,1,4863,0,4863,0,4863,0,4863,0,Poor,3.36,Poor,87.2,Poor,Poor Cluster 2: 38,2.5,0,2,2,3,2,1,422,1,1484,1,1233,1,1233,1,Poor,3.5,Poor,81.5,P	316.7792654

				57 (29%)	oor,Poor	
4	E	10	F	0 39 (20%) 1 67 (34%) 2 57 (29%) 3 35 (18%)	Cluster 0: 43,2.5,0,1,1,1,1,1,1429,1,5571,0,5571,0,5571,0,Poor,2.5,Good,88.7,Good,Good Cluster 1: 51,1.8,0,1,1,2,2,1,4863,0,4863,0,4863,0,4863,0,Poor,3.36,Poor,87.2,Poor,Poor Cluster 2: 38,2.5,0,2,2,3,2,1,422,1,1484,1,1233,1,1233,1,Poor,3.5,Poor,81.5,Poor,Poor Cluster 3: 47,2.5,1,1,0,1.888889,3,0,4487,0,4487,0,4487,0,4487,0,Poor,4.5,Poor,66.4,Poor,Poor	288.7048555

根据以上表格，我们不难得出如下结论：

a)聚类数即 k 值越大，每个聚类的误差值（"聚类内平方误差之和"）越小。但聚类数的一味增长是没有意义的，在极端情况下，每个实例归为一类,误差值为 0，但此时已经失去了聚类的意义。

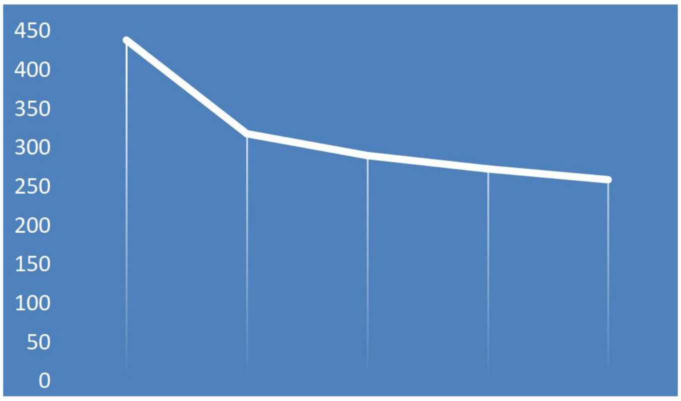
b)在本实验的数据集下，选择 Euclidean 作为距离指标的效果好于选择 Manhattan (= cityblock)的效果。

c)种子值对实验结果几乎没有影响。

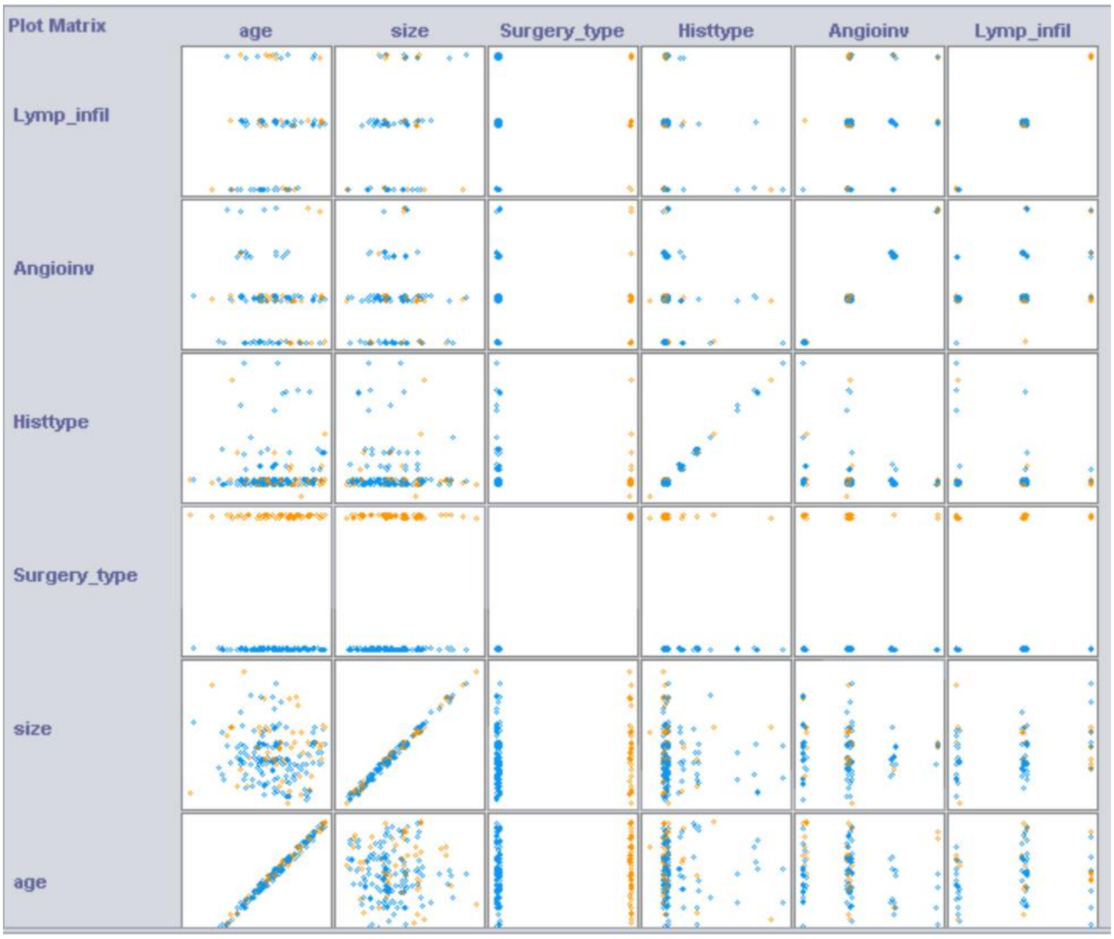
d)数据是否标准化对实验结果几乎没有影响。

3) 聚类分析

选择最佳聚类（即误差值最低）。制作结果的散点图可视化。



虽然在上述参数中 k=6 是最优的参数，但从上图中我们发现 k=3 时是误差值改变的一个拐点，在此后误差值随 k 值的变化较为平缓。正如在上述分析中所提到的聚类数的一味增长是没有意义的，在极端情况下，每个实例归为一类,误差值为 0，但此时已经失去了聚类的意义。因此我们选择 k=3 时的结果进行展示。



任务 2

Practice to code for k-means algorithm with the following dataset. The Number of clusters is 3. The distance function is Euclidean distance. List the initial data, the center of the iteration 1, and the final clusters.

ID	Feature 1	Feature 2
1	2	10
2	2	5
3	8	4
4	5	8
5	7	5
6	6	4
7	1	2
8	4	9

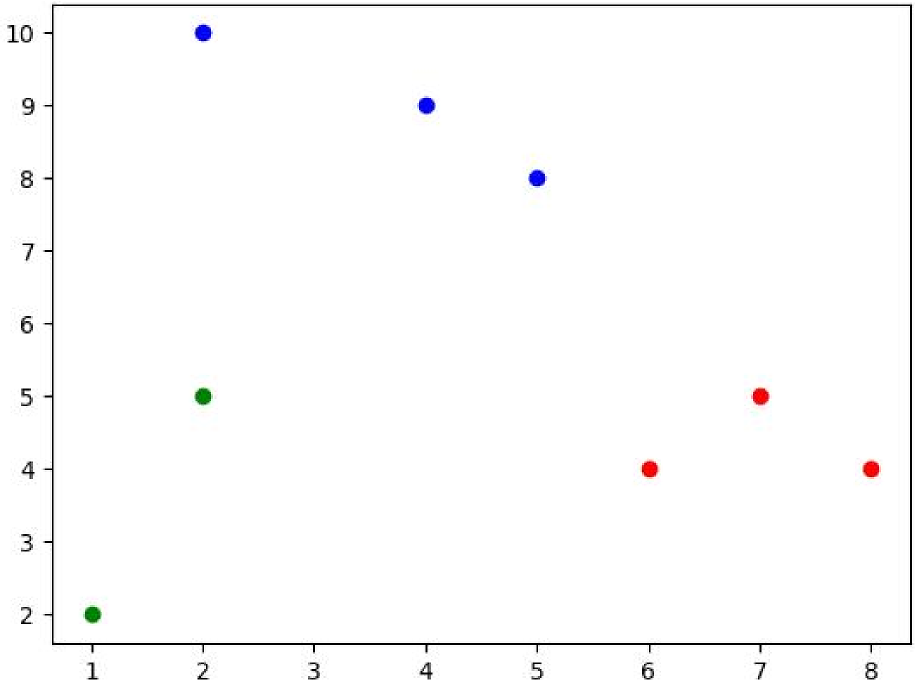
编写 python 程序如下：

```
import random
import numpy as np
import matplotlib.pyplot as plt

k = 3
rnd = 0
ROUND_LIMIT = 10
THRESHOLD = 1e-10
melons = []
clusters = []

f = open('test4.2.txt', 'r')
for line in f:
    melons.append(np.array(line.split(' ')).astype(np.int32))
```

```python
mean_vectors = random.sample(melons, k)
for v in mean_vectors:
    print(v)
while True:
    rnd += 1
    change = 0
    clusters = []
    for i in range(k):
        clusters.append([])
    for melon in melons:
        c = np.argmin(
            list(map(lambda vec: np.linalg.norm(melon - vec, ord=2), mean_vectors))
        )

        clusters[c].append(melon)

    for i in range(k):

        new_vector = np.zeros((1, 2))
        for melon in clusters[i]:
            new_vector += melon
        new_vector /= len(clusters[i])

        change += np.linalg.norm(mean_vectors[i] - new_vector, ord=2)
        mean_vectors[i] = new_vector
    if rnd == 1:
        for v in mean_vectors:
            print(v)
    if rnd > ROUND_LIMIT or change < THRESHOLD:
        break

print('最终迭代%d 轮' % rnd)

colors = ['red', 'green', 'blue']

for i, col in zip(range(k), colors):
    for melon in clusters[i]:
        plt.scatter(melon[0], melon[1], color=col)

plt.show()
```

初始中心点（随机选择）：[1 2]　,[5 8],[4 9]

第一轮迭代后的聚类中心：[[1.5 3.5]],[[6.5　5.25]],[[3.　9.5]]最终聚类结果如下图：


10
9
8
7
6
5
4
3
2
1
2
3
4
5
6
7
8