人工智能第二次实验实验报告

第二部分：非监督学习算法

李瀚民
PB16001680

[任务描述]:
用给出的葡萄酒属性数据集，以及葡萄酒原本的种类，进行 PCA 和原始数据 K-means 聚类的对比，通过对比数据的 Sihouette 以及 Rand 系数，来比较实际上的效率高低。

[算法思想]:
首先简要介绍一下 PCA，主元分析试图将给出的数据从 d 维降低到 m 维，使得数据在 m 维的投影最分散，或者说，数据到 m 维的平方距离最小，通过构建协方差矩阵，选出 m 个特征值构造投影矩阵来实现投影，m 维互不相关，因此取最大的 m 个特侦向量即可。在本次实验中直接由函数 PCA 进行处理。
接着是 K-means 聚类，其思想主要是，（1）先（随机-第一次）选 k 个中心，（2）然后将所有点按中心分类，接着（3）选 k 个新的中心，重复（2），（3），由（2），（3）中点距离中心的距离缩小可知，k-means 在一些情况下可以得到良好的分类结果。在本次实验中直接用函数 KMeans 进行处理
然后说一下轮廓系数和兰德系数，轮廓系数取值范围[-1， 1]，取值越高越好，兰德系数取值范围[0， 1]当取 1 时完美拟合。

[实验环境]:
Windows 10 pro + pycharm

[实验结果分析]:
（1） 不同 threshold 的降维结果：（以下测试时 k=4，循环迭代次数：4）
当 threshold = 0.3 时

```
    dr_wine_data = PCA(wine_data, 0.3)
main()
main
C:\Users\lihanming\miniconda3\envs\AI\python.exe C:/AI/main.py
After PCA dimensons: 1
Original:     Sihouette: -0.03579902528408782   Rand: 0.630948961026525
PCA:    :     Sihouette: -0.0830408899110619    Rand: 0.605988062178812
```

仅生成了一个特征，显而易见的不行，并且没有原来的直接聚类比较好
当 threshold = 0.5 时

```
    dr_wine_data = PCA(wine_data, 0.5)
main()
main
C:\Users\lihanming\miniconda3\envs\AI\python.exe C:/AI/main.py
After PCA dimensons: 2
Original:     Sihouette: -0.012865580632141318   Rand: 0.8296785725685467
PCA:    :     Sihouette: -0.07067260376804722    Rand: 0.7139391617989722
```

此时生成了两个特征，并且可以看出来没有原来的聚类效果好
当 threshold = 0.7 时

```
    # dimenson reduction
    dr_wine_data = PCA(wine_data, 0.7)

main()
```

```
main
C:\Users\lihanming\miniconda3\envs\AI\python.exe C:/AI/main.py
After PCA dimensons: 4
Original:     Sihouette: -0.0193352344490734   Rand: 0.6908295828146446
PCA:    :     Sihouette: -0.017763713881703735   Rand: 0.7092151042165405
```

生成了 4 个特征，并且聚类效果和原来的差不多

当 threshold = 0.85 时

```
    # dimenson reduction
    dr_wine_data = PCA(wine_data, 0.85)

main()
```

```
main
C:\Users\lihanming\miniconda3\envs\AI\python.exe C:/AI/main.py
After PCA dimensons: 6
Original:     Sihouette: -0.021776970012564914   Rand: 0.7133007756391841
PCA:    :     Sihouette: -0.038828388730862355   Rand: 0.6945960611573941
```

生成了 6 个特征，并且聚类效果和原来差不多

当 threshold = 0.93 时

```
    # dimenson reduction
    dr_wine_data = PCA(wine_data, 0.93)

main()
```

```
main
C:\Users\lihanming\miniconda3\envs\AI\python.exe C:/AI/main.py
After PCA dimensons: 10
Original:     Sihouette: -0.025205345568280985   Rand: 0.6676561652143381
PCA:    :     Sihouette: -0.010305799134064195   Rand: 0.6624852373200549
```

保留了 10 个特征，并且聚类效果比原先的略要好一些。

结合固定变量的原则，综合以上的发现，在 k=4 时（固定变量），其实有效特征在 4 个左右，PCA 多选出来的特征并不会对聚类结果产生本质的影响。对于其他的 k 值

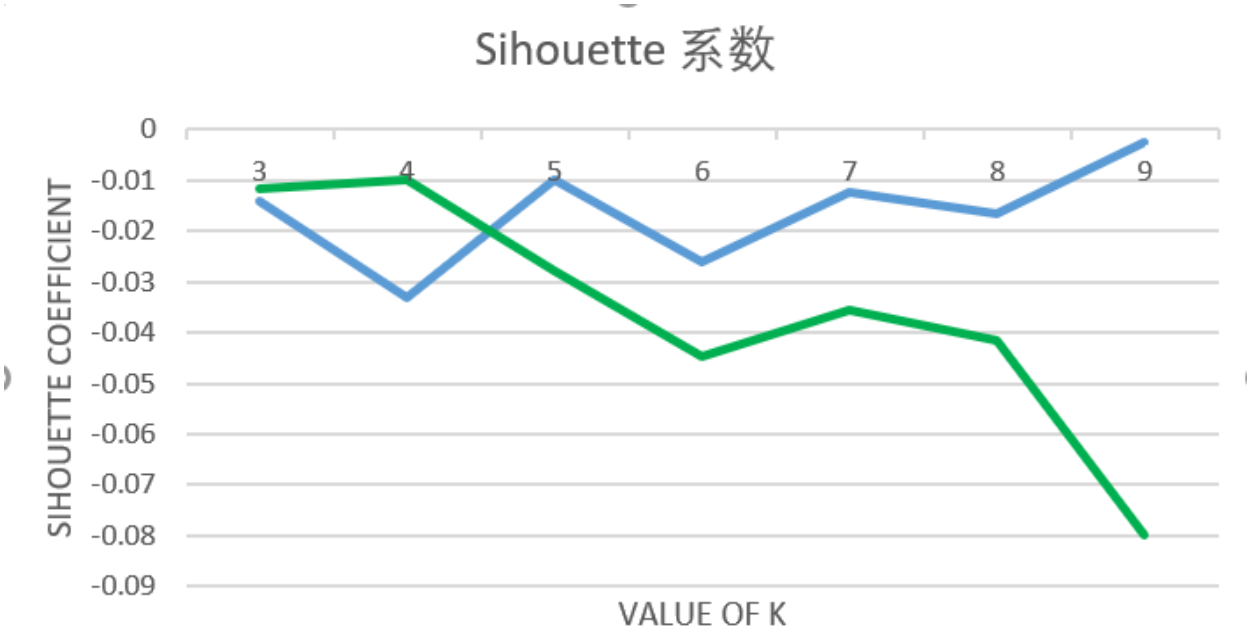
或者迭代次数，除非出现经过迭代所有的数据最终成为一类的情况，其余情况基本都符合上述规律，但是由于一开始选择的 k 个点具有随机性，所以其实会产生一定的偏差。

(2) 不同聚类数系数与结果对比：

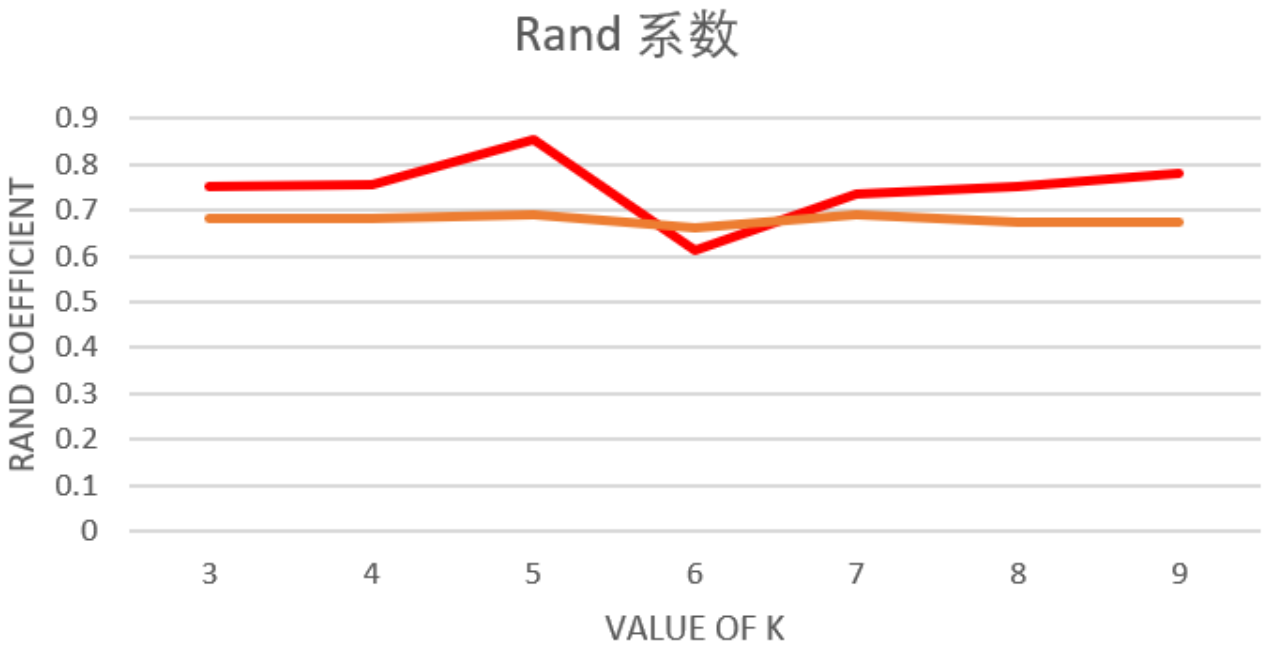
K 值	Sihouette（means）	Rand（means）	评价（曲线走势）
3/org	-0.014079916974713624	0.7513485907625522	持平
3/pca	-0.011456391676469878	0.6825944013533787	持平
4/org	-0.03291510589928555	0.7570940662006448	上升
4/pca	-0.009776779560080317	0.6809345973379297	上升
5/org	-0.00996115748118406	0.8541287624884293	下降
5/pca	-0.02788870010733245	0.6915956462063902	下降
6/org	-0.026085052146658808	0.613521018864311	上升
6/pca	-0.04482558401999697	0.6605062402247119	下降
7/org	-0.012449970600040518	0.7338568099843595	上升
7/pca	-0.03556845906318318	0.6901911966548565	上升
8/org	-0.016439482398744154	0.7516039452264675	上升
8/pca	-0.0416388613720723	0.6739123495802611	下降
9/org	-0.002650143812794202	0.7818634492004214	持平

9/pca	-0.08002792687711768	0.6735293178843883	

作出以下曲线图来表示：



其中蓝色线代表着原始未经过 PCA 处理的数据聚类轮廓系数，绿色则表示经过 PCA 的聚类轮廓系数，可以看出来，当 K=3，4 时两者表现都很好，说明原本的数据分类其实已经比较完好，当 K 继续变大是，原本数据并没有明显变化，但是 PCA 会出现一定的变化，这里选定的 threshold 值是 0.93，为之前选择的最佳值。

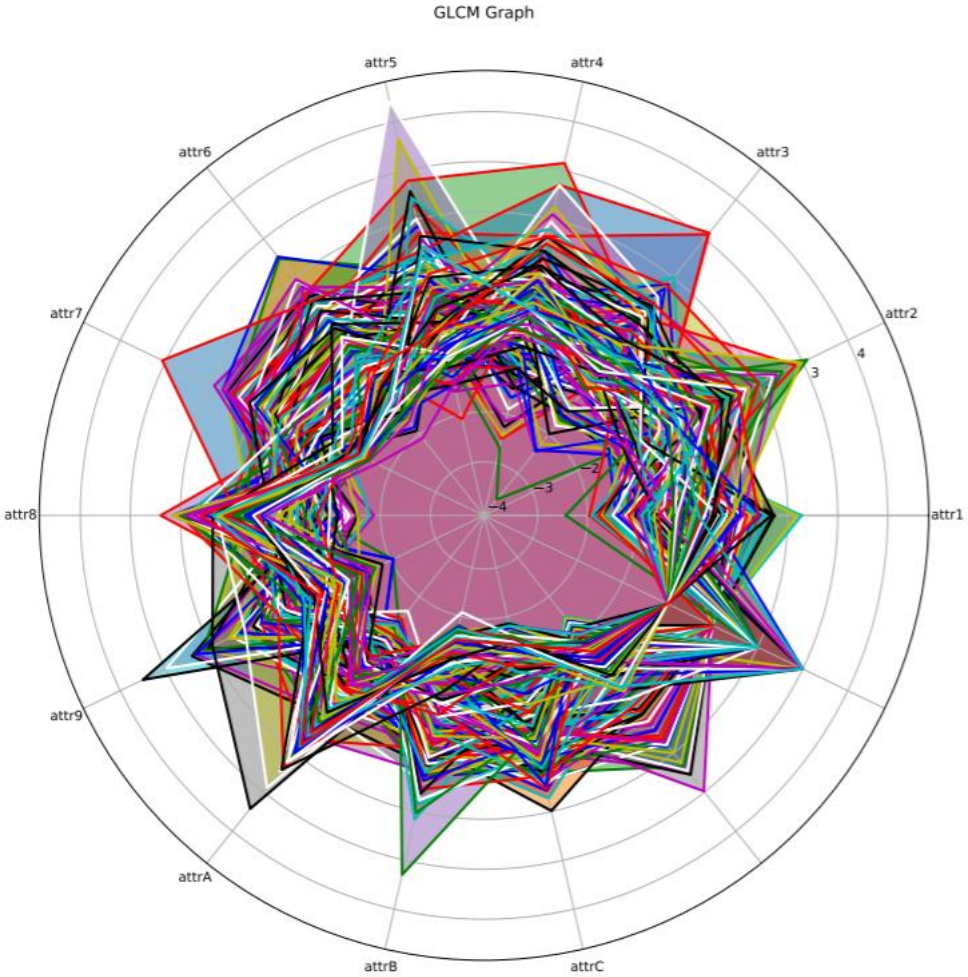


可以看出 Rand 系数这里的走势轮廓系数其实基本一致，由于 K-Means 一开始引入了随机性，所以有一些小的波动基本也算正常，这里与之前比较类似，所以也算符合预期。

[可视化处理]:

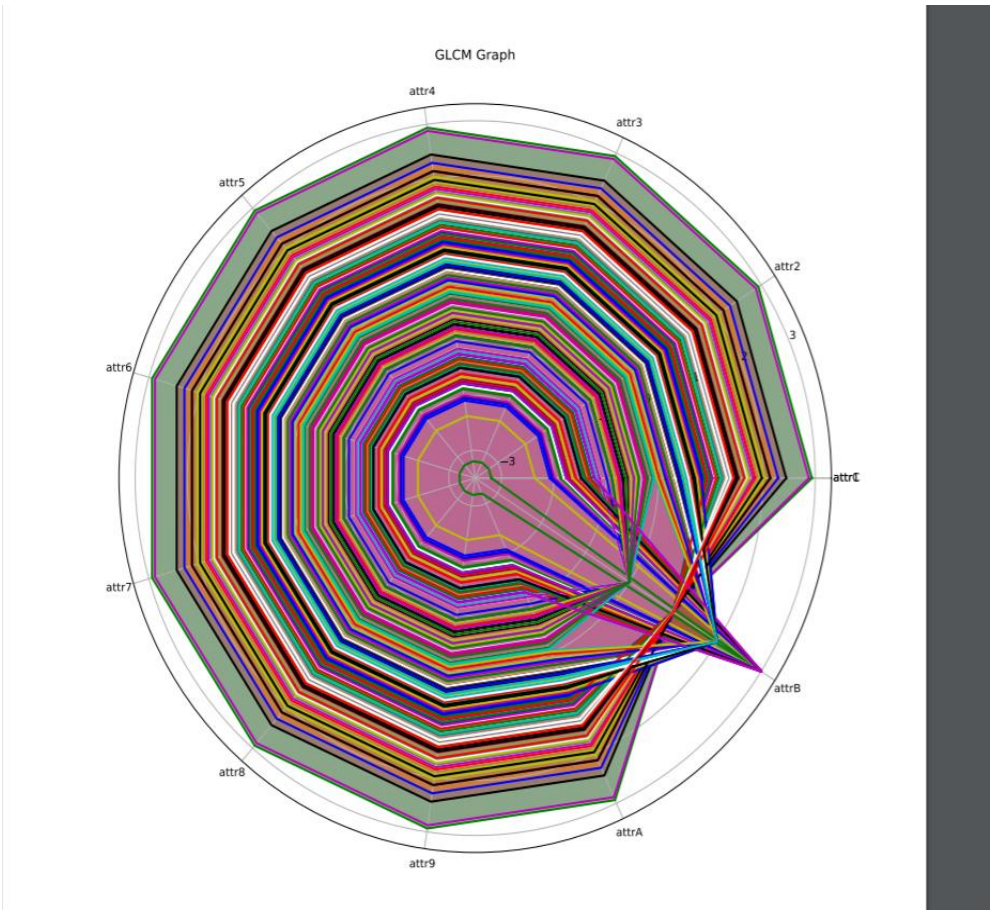
通过灰度共生雷达图来处理（radar-GLCM）处理。

K = 4，threshold = 0.93 未经 PCA 处理：



样本有点太多了，效果不太好。

同样参数经过 PCA 处理：



可以看出特侦之间的相关性明显要变弱了，所以总体来说 PCA 有一定效果，虽然这个图看起来有点费劲，但是还是可以反映一些实际情况。