第二次实验课

一. 实验的内容

1. 数据的生成

（1） 利用 sklearn 库提供的函数生成一组高维数据。

（2） 导入任意一个高维 UCI 标准数据集。

2. 降维（PCA）

我们把上一步得到的高维数据集进行降维操作，为了可视化的缘故，我们选择将其降到二维。

3. 聚类（k-means）

将降维后的数据集进行 k-means 聚类，选择 NMI（归一化互信息）作为衡量聚类结果的指标。

（1） 选择不同的初始向量，对比聚类结果。

（2） 选择不同的k值，观察聚类结果。

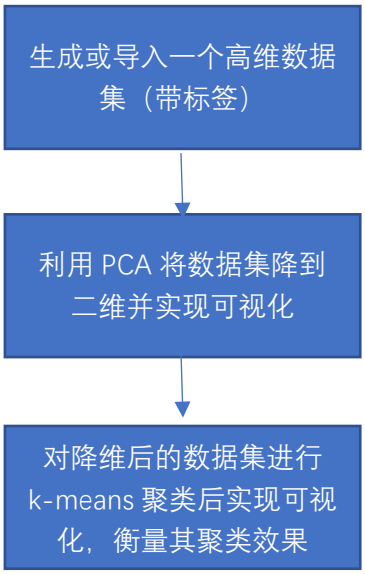

图 1 实验内容流程图

二. 实验的代码

```
# 数据生成
sklearn.datasets.make_blobs(n_samples=100,                    # 样本个数
                            n_features=2,                     # 样本特征数
                            centers=None,                     # 簇中心
                            cluster_std=1.0,                  # 标准差
                            center_box=(- 10.0, 10.0),        # 范围
                            shuffle=True,                     # 打乱顺序
                            random_state=None,                # 是否复现这
                            return_centers=False)             # 输出中心
# 降维
sklearn.decomposition.PCA(n_components=3,                     # 降维后的维数
                          random_state=None)                  # 是否复现
# 聚类
sklearn.cluster.KMeans(n_clusters=8,                          # 聚类簇数k
                       init='k-means++',                      # 初始向量的选取
                       n_init=10,                             # 运行10次取最好的结果
                       max_iter=300,                          # 循环次数
                       tol=0.0001,                            # 收敛阈值
                       random_state=None)                     # 是否复现
# NMI
sklearn.metrics.normalized_mutual_info_score(labels_true,     # 真实标签
                                             labels_pred)     # 预测标签
```

图 2 关键伪代码

三. 实验的结果

1. 降维后的散点图

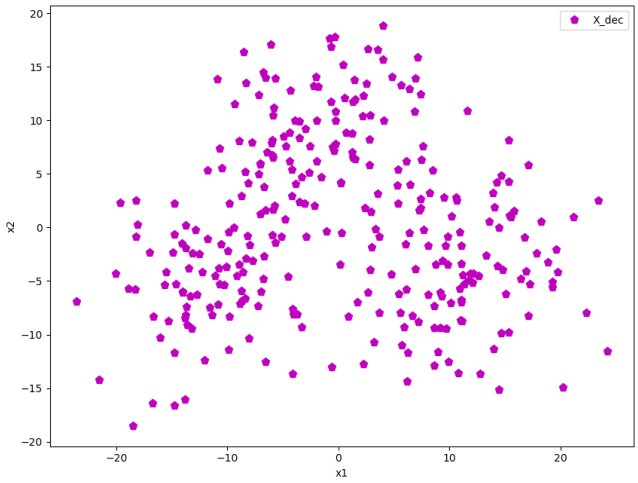

图 3 降维后的散点图

2. 不同k值聚类后的散点图

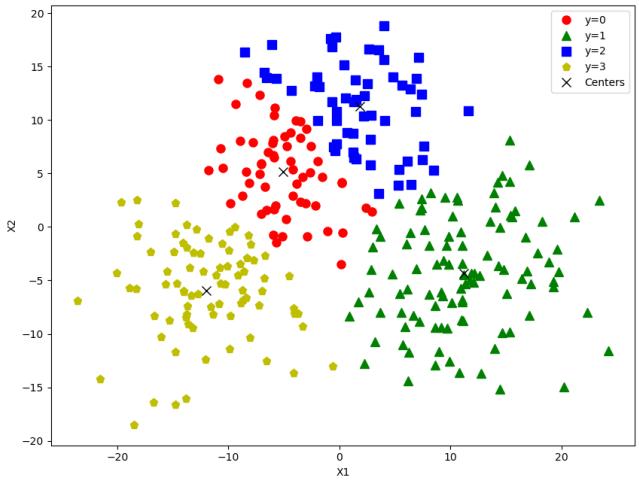

图 4 k=4 的情况

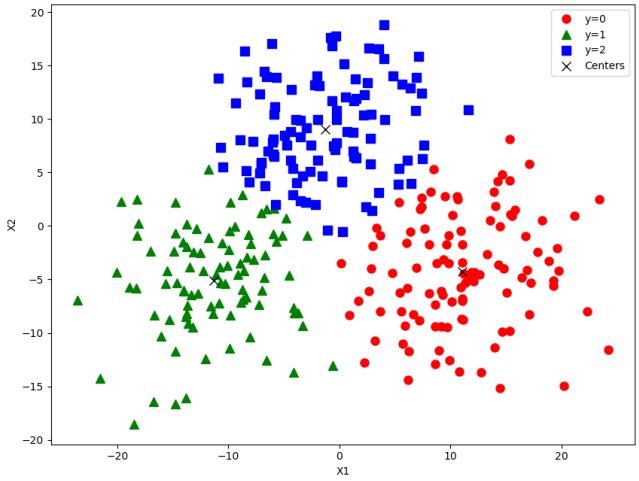

图 4 k=3 的情况

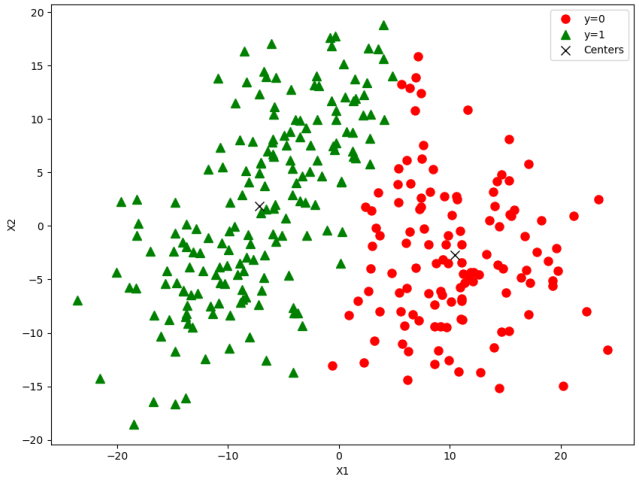

图 5 k=2 的情况

3. 聚类评价指标

我们在在固定模型的其它参数情况下，随机选择初始向量，运行 20 次得到的结果如下所示。实验证明，NMI 的值与初始向量的选取有关。

```
for i in range(20):
    model = KMeans(n_clusters=3,
                   init='random',
                   n_init=1).fit(X_dec)  # 调用k-means模型
    y_pred = model.labels_  # 输出预测标签
    clu_centers = model.cluster_centers_  # 输出聚类中心
    print('NMI‘s value :', metrics.normalized_mutual_info_score(y, y_pred))  # NMI值
```

图 6 k-means 模型

NMI‘s value : 0.7787825510531236
NMI‘s value : 0.7787825510531237
NMI‘s value : 0.8001329799720475
NMI‘s value : 0.8001329799720475
NMI‘s value : 0.8001329799720475
NMI‘s value : 0.7787825510531236
NMI‘s value : 0.7856128301970217
NMI‘s value : 0.7787825510531234
NMI‘s value : 0.7787825510531236
NMI‘s value : 0.8128112279157803
NMI‘s value : 0.8001329799720475

NMI‘s value : 0.7856128301970219
NMI‘s value : 0.8001329799720475
NMI‘s value : 0.8001329799720475
NMI‘s value : 0.8128112279157803
NMI‘s value : 0.7787825510531237
NMI‘s value : 0.8128112279157806
NMI‘s value : 0.7856128301970217
NMI‘s value : 0.7856128301970219
NMI‘s value : 0.8128112279157806