实验八：聚类

一、说明

●实验采用 jupyter notebook, 请填写完代码后提交完整的 ipynb 文件

●文件命名 规则：班级_姓名_ML2019_HW8.ipynb,如:计科 1701_张三_ML2019_HW8.ipynb

●提交方式：采用在线提交至：

http://pan.csu.edu.cn:80/invitation/8d5a43a1-9292-4f3c-a9d5-6e18f464beba

●实验提交截至日期：2020.1.5 23:59

二、实验内容

在“无监督学习”（unsupervised learning）中，训练样本的标记信息是未知的，目标是通过对无标记训练样本的学习来揭示数据的内在性质及规律，为进一步的数据分析提供基础。在本次实验中，我们将使用 K-means 算法和 GMM 算法来对我们的数据集进行聚类；

三、实验目标

掌握 K-means 聚类算法；

掌握 GMM 聚类算法；

四、实验操作步骤

1. 启动 jupyter notebook

参考实验一，打开文件 Clustering.ipynb

2. 完成实验任务

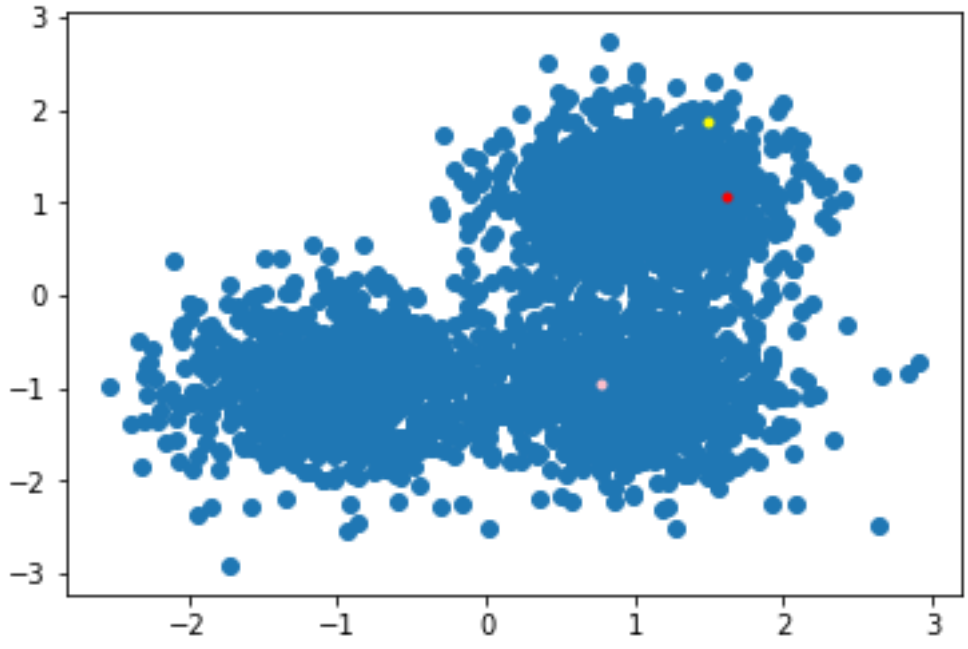
任务 1 随机初始化 centroids

具体内容见 Clustering.ipynb 文件，参考答案：

```
#START CODE HERE#
i = np.random.randint(0, X.shape[0], K)
centroids = X[i, :]
#END CODE HERE#
```

检测条件：运行任务 1 后，输出以下结果

```
初始 centroids 为:  [[ 1.62   1.08]
 [ 0.78 -0.94]
 [ 1.49   1.87]]
```



任务 2 寻找每个样本离之最近的 centroid

具体内容见 Clustering.ipynb 文件，参考答案：

```
#START CODE HERE#
for i in range(m):
    distance = np.sum(np.power(centroids - X[i, :], 2), axis=1)
    index[i] = np.argmin(distance, axis=0)
#END CODE HERE#
```

检测条件：运行任务 2 后，输出以下结果

```
前 10 个索引为： [1. 0. 0. 2. 1. 1. 1. 0. 0. 0.]
```

任务 3 更新 centroids

具体内容见 Clustering.ipynb 文件，参考答案：

```
### START CODE HERE ###
for i in range(K):
    idx = np.where(index == i)
    centroids[i, :] = np.mean(X[idx, :], axis=1)
### END CODE HERE ###
```

检测条件：运行任务 3 后，输出以下结果

```
更新后的 centroids 为： [[ 1.07924138  0.84577931]
 [ 0.01198136 -0.96947033]
 [ 0.95470339  1.6675    ]]
```

任务 4 整合前面的内容，完成对数据集的聚类

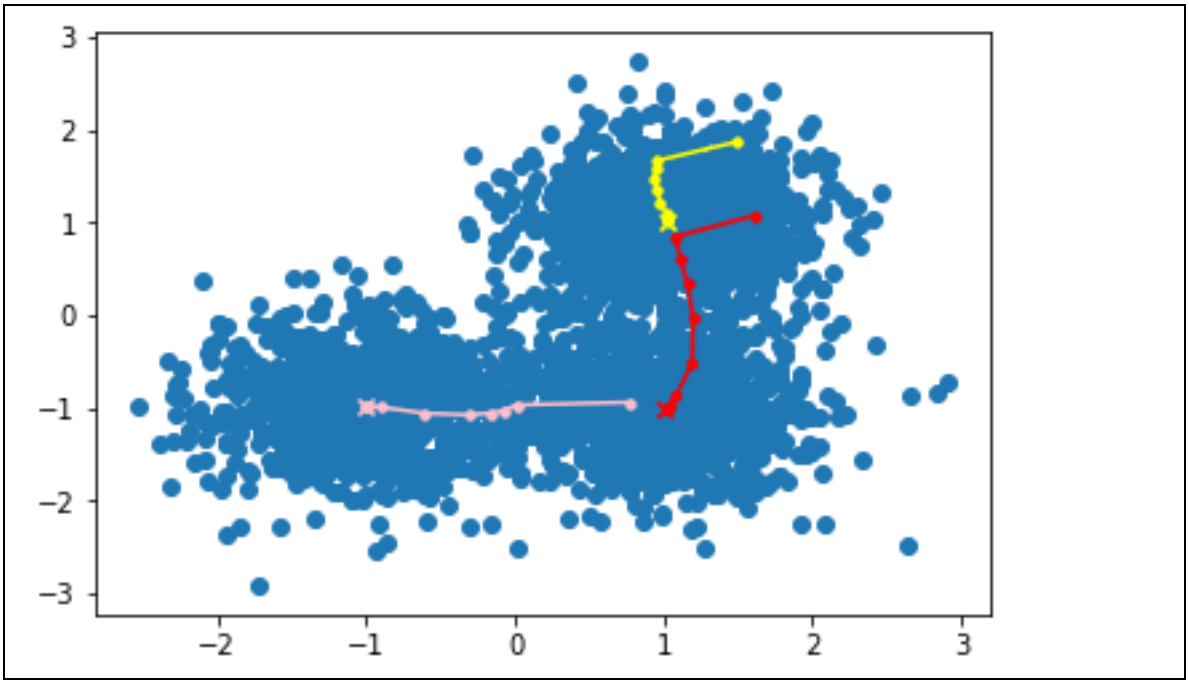
具体内容见 Clustering.ipynb 文件，参考答案：

```
#初始化聚类簇中心 centroids
#START CODE HERE#
centroids = kMeansInitCentroids(X, K)
#END CODE HERE#

#计算索引集 index
#START CODE HERE#
index = findClosestCentroids(X, centroids)
#END CODE HERE#

#更新 centroids
#START CODE HERE#
centroids = computeCentroids(X, index, K)
#END CODE HERE#
```

检测条件：运行任务 4 后，输出以下结果



任务 5 完成 EM 算法的 E 步骤

具体内容见 Clustering.ipynb 文件，参考答案：

```
#START CODE HERE#
#计算各模型中所有样本出现的概率，行对应样本，列对应模型
prob = np.zeros((m, K))
for k in range(K):
    prob[:, k] = alpha[k] * mul_normal(data, mu[k], sigma[k])
gamma = prob / np.sum(prob, axis=1, keepdims=True)
#END CODE HERE#
```

检测条件：运行任务 5 后，输出以下结果

```
[[0.27926041 0.41488758 0.30585201]
 [0.33855553 0.31438905 0.34705542]
 [0.32877483 0.34119351 0.33003167]
 [0.36472108 0.33078144 0.30449749]
 [0.26201086 0.43229054 0.3056986 ]
 [0.27737269 0.41066866 0.31195865]
 [0.2963848  0.36449846 0.33911674]
 [0.34046085 0.36464825 0.29489089]
 [0.34252555 0.34306642 0.31440803]
 [0.34066638 0.31310663 0.34622698]]
```

任务 6 完成 EM 算法的 M 步骤

具体内容见 Clustering.ipynb 文件，参考答案：

```
#START CODE HERE#
#初始化高斯混合分布的模型参数值，因为要更新它们
mu = np.zeros((K, n))
sigma = []
mk = np.sum(gamma, axis=0)
```

```
#更新每个高斯混合成分的模型参数
for k in range(K):
    #更新 mu
    mu[k, :] = gamma[:, k].reshape(1, m) * data / mk[k]
    #更新 sigma
    sigma_k = (data - mu[k]).T * np.multiply((data - mu[k]), gamma[:,
k].reshape(m, 1)) / mk[k]
    sigma.append(sigma_k)
#更新 alpha
alpha = mk / m
sigma = np.array(sigma)     #为了保持一致，还需将 sigma 转回 array
#END CODE HERE#
```

检测条件：运行任务 6 后，输出以下结果

```
The mu_test is: [[ 0.40729641 -0.21531751]
 [ 0.39499207 -0.38839127]
 [ 0.23195875 -0.35311593]]
The sigma_test is: [[[1.09405764 0.45442549]
  [0.45442549 1.1883895 ]]

 [[1.11085862 0.38770788]
  [0.38770788 1.10681648]]

 [[1.18905033 0.46874545]
  [0.46874545 1.10981631]]]
The alpha_test is: [0.30744145 0.35697314 0.33558541]
```

任务 7 整合高斯混合聚类算法，对我们的数据进行聚类

具体内容见 Clustering.ipynb 文件，参考答案：

```
#START CODE HERE#
gamma = Expectation(data, mu, sigma, alpha, K)
mu, sigma, alpha = Maximization(data, gamma, K)
```

```
#END CODE HERE#
```

检测条件：运行任务 7 后，输出以下结果

```
The first 10 elements in gamma is: [[7.19238109e-06 9.99590
239e-01 4.02568641e-04]
 [9.98954907e-01 8.43714591e-04 2.01378679e-04]
 [9.61879109e-01 3.78303210e-02 2.90570157e-04]
 [9.99996223e-01 3.77714219e-06 4.21924890e-11]
 [6.05856138e-08 9.98657039e-01 1.34289993e-03]
 [5.66085810e-06 9.98127708e-01 1.86663117e-03]
 [3.77445635e-03 7.81517115e-01 2.14708429e-01]
 [9.93991137e-01 6.00884977e-03 1.36614537e-08]
 [9.98357801e-01 1.64194208e-03 2.56967788e-07]
 [9.99425237e-01 4.84772029e-04 8.99910125e-05]]
```