Assignment 3

Pattern Recognition and Machine Learning

FUDAN UNIVERSITY @ 2020 SPRING

1 问题描述

在本次作业中，需要构造一个没有任何标签的集群数据集，并设计一个高斯混合模型来完成无标签的聚类任务。

2 数据生成

根据公式1和表4生成三个均值为 μ_i ，协方差矩阵为 Σ_i , 在整个数据集中占比为 π_i 的二维高斯分布（i=0,1,2）。总数据集大小为 300。

$$f_i(x) = \frac{1}{\sqrt{(2\pi)^k \left|\Sigma_i\right|}} exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)) \tag{1}$$

i	μ	Σ	π
0	[5, 0]	[[1,0],[0,1]]	$\frac{1}{4}$
1	[1, 1]	[[1,0],[0,3]]	$\frac{1}{2}$
2	[0, 5]	[[1,-1],[-1,3]	$\frac{1}{4}$

表 1: 三个二维高斯分布的参数

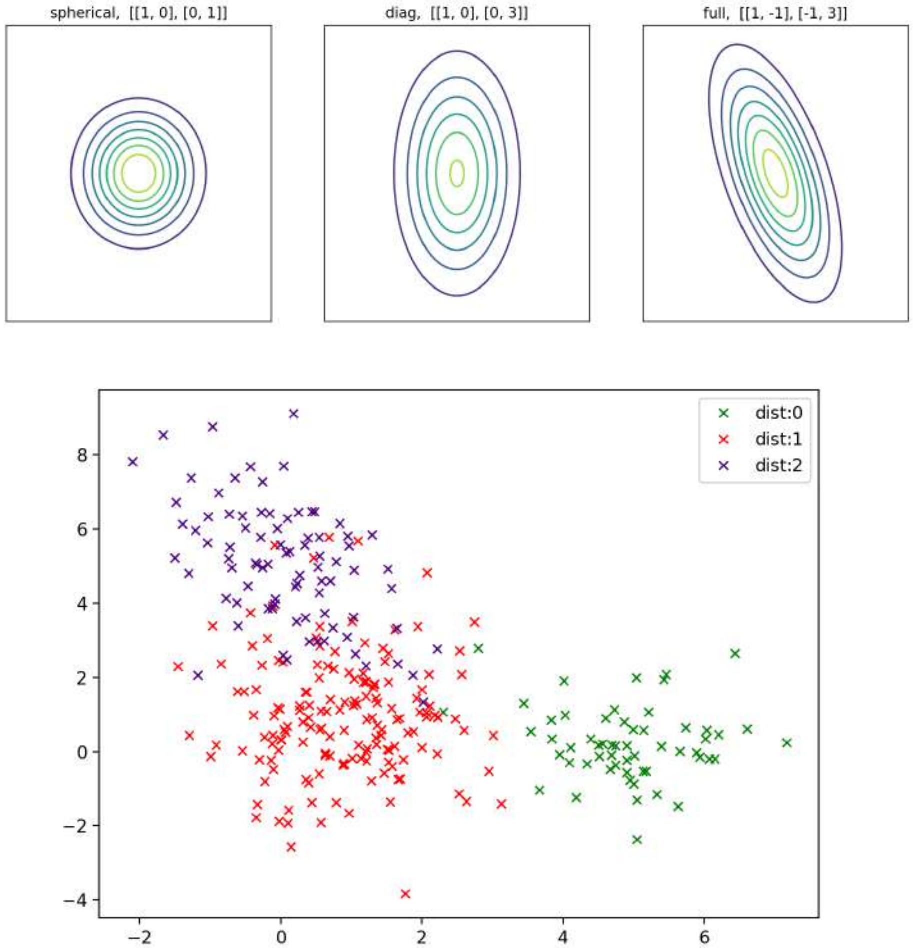

图 1: 数据分布

3 高斯混合模型

高斯混合模型 (Gaussian Mixture Model，GMM) 是由多个高斯分布组成的模型，其总体密度函数为多个高斯密度函数的加权组合。

$$P(y|\theta) = \sum_{k=1}^{K} \pi_k \Phi(y|\theta_k) \tag{2}$$

其中，π_k 是系数，$\pi_k \geqslant 0$ ，$\sum\limits_{k=1}^{K} \pi_k = 1$ ；$\Phi(y|\theta_k)$ 是高斯分布函数，$\theta_k = (\mu_k, \sigma_k^2)$。

4 高斯混合模型参数估计的 EM 算法

输入：观测数据 $y_1, y_2, ..., y_N$，高斯混合模型；

输出：高斯混合模型参数。

在 GMM 中，每个簇对应一个概率分布，我们要做的是学习这些分布的参数, 即高斯的均值 μ 和方差 σ^2。

4.1 初始化

取参数的初始值开始迭代；

- **均值** μ: 从观测数据中随机选择 n_components 个（不放回），被选中的数据作为初始化的均值。
- **方差** σ^2: 每个高斯分布的方差统一初始化为整个数据集上的协方差矩阵。
- **系数** π：假设每个高斯分布系数相同

```
# initialize parameters
    np.random.seed(self.seed)
    chosen = np.random.choice(n_row, self.n_components, replace=False)
    self.means = X[chosen]

    shape = self.n_components, n_col, n_col
    # for np.cov, rowvar = False,indicates that the rows represents obervation
    self.covs = np.full(shape, np.cov(X, rowvar=False))

    self.weights = np.full(self.n_components, 1 / self.n_components)
```

4.2 E 步

依据当前模型参数，计算分模型 k 对观测数据 y_j 的响应度

$$\hat{y}_{jk} = \frac{\pi_k \Phi(y_j|\theta_k)}{\sum\limits_{k=1}^{K} \pi_k \Phi(y_j|\theta_k)} \qquad j = 1, 2, ...N; k = 1, 2, ..., K \tag{3}$$

```
    def _compute_log_likelihood(self, X):
        for k in range(self.n_components):
            prior = self.weights[k]
            likelihood = multivariate_normal(self.means[k], self.covs[k]).pdf(X)
            self.resp[:, k] = prior * likelihood
        return self

    def _do_estep(self, X):
        self._compute_log_likelihood(X)
        log_likelihood = np.sum(np.log(np.sum(self.resp, axis=1)))
        # normalize over all possible cluster assignments
        self.resp = self.resp / self.resp.sum(axis=1, keepdims=1)
        return log_likelihood
```

4.3 M 步

计算新一轮迭代的模型参数

$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{y}_{jk} y_j}{\sum\limits_{j=1}^{N} \hat{y}_{jk}} \qquad k = 1, 2, ..., K \tag{4}$$

$$\sigma_k^2 = \frac{\Sigma_{j=1}^{N} \hat{y}_{jk} (y_j - \mu_k)^2}{\sum\limits_{j=1}^{N} \hat{y}_{jk}} \qquad k = 1, 2, ..., K \tag{5}$$

$$\pi_k = \frac{\sum\limits_{j=1}^{N} \hat{y}_{jk}}{N} \qquad k = 1, 2, ..., K \tag{6}$$

```
def _do_mstep(self, X):
    # total responsibility assigned to each cluster, N^{soft}
    resp_weights = self.resp.sum(axis=0)

    # weights
    self.weights = resp_weights / X.shape[0]

    # means
    weighted_sum = np.dot(self.resp.T, X)
    self.means = weighted_sum / resp_weights.reshape(-1, 1)

    # covariance
    for k in range(self.n_components):
        diff = (X - self.means[k]).T
        weighted_sum = np.dot(self.resp[:, k] * diff, diff.T)
        self.covs[k] = weighted_sum / resp_weights[k]

    return self
```

4.4 收敛

重复 E 步和 M 步，直到收敛。收敛条件为迭代次数已经达到预先设定的 n_iters 或两次训练的对数似然值之差小于预先设定的 tolerance。

```
for i in range(self.n_iters):
    log_likelihood_new = self._do_estep(X)
    self._do_mstep(X)
    self.get_labels()
    if abs(log_likelihood_new - log_likelihood) <= self.tol:
        self.converged = True
        break
    log_likelihood = log_likelihood_new
    self.log_likelihood_trace.append(log_likelihood)
```

5 模型性能

5.1 评价指标

在无监督聚类中，真实标签未知，则必须使用模型本身进行评估。

- 轮廓系数（Silhouette Coefficient, SC）

  SC 结合内聚度和分离度两种因素，可以用来在相同原始数据的基础上用来评价不同算法、或者算法不同运行方式对聚类结果所产生的影响。轮廓系数越高，表示模型的聚类效果越好。定义每个样本的轮廓系数，由两个分值组成:

  - a: 样本与同一集群中所有其它点之间的平均距离。
  - b: 样本与最近集群中所有其它点之间的平均距离。

$$SC = \frac{b-a}{max(a,b)} \tag{7}$$

- Calinski-Harabasz Index（CHI）

  CHI 是所有集群的集群内离散度和集群间离散度之和的比值[1]:

$$\begin{aligned} CBI &= \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k-1} \\ W_k &= \sum_{q=1}^{k} \sum_{x \in C_i} (x-c_q)(x-c_q)^T \\ B_k &= \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \end{aligned} \tag{8}$$

  CHI 越大，聚类效果越好。

- Davies-Bouldin 指数（Davies-Bouldin Index, DBI）

  DBI 表示集群之间的平均“相似度”，相似度是比较集群之间的距离和集群本身的大小的度量。因此，Davies-Bouldin 指数越小，模型分离集群的能力越好。

  - s_i: 集群的每个点与该集群的质心之间的平均距离。
  - d_ij: 集群 i 和集群 j 质心之间的距离。

$$\begin{aligned} R_{ij} &= \frac{s_i + s_j}{d_{ij}} \\ DBI &= \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij} \end{aligned} \tag{9}$$

[1] 离散度: 距离的平方和

5.2 迭代次数 n_iters

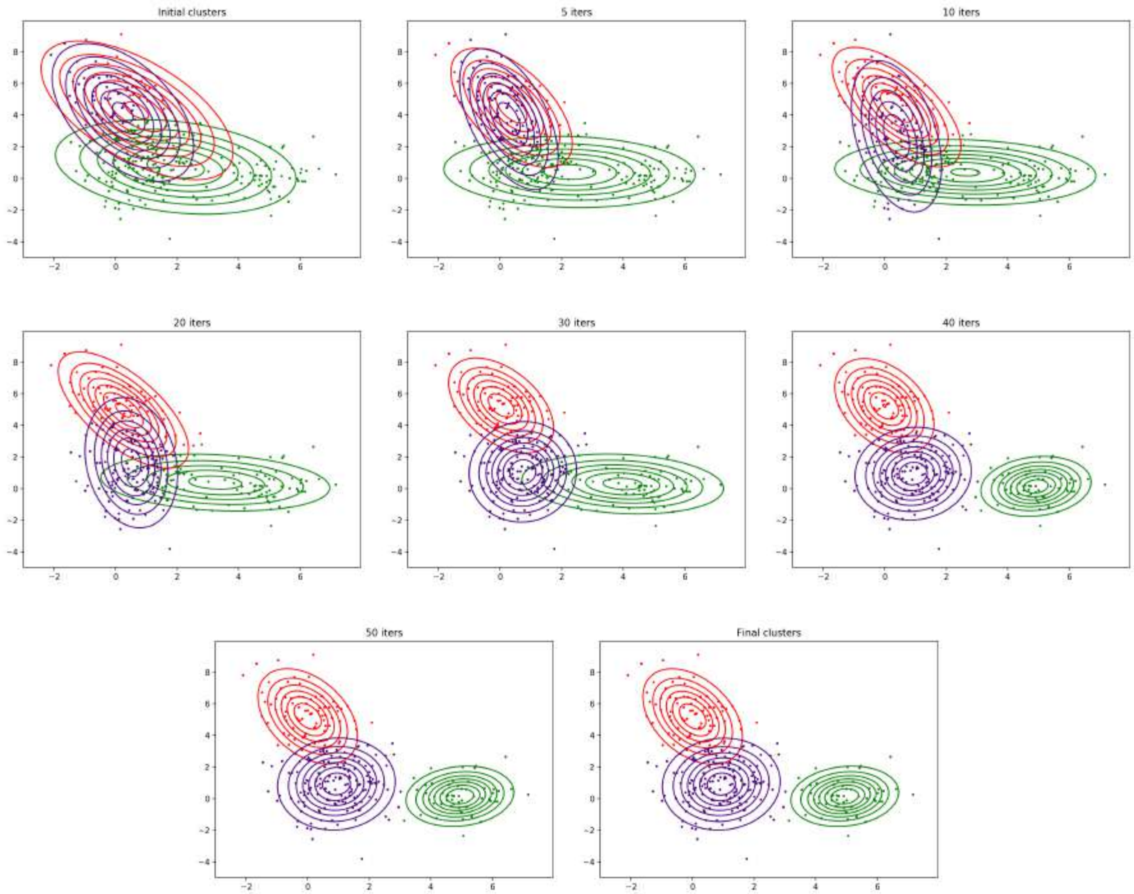

图 2: Caption

从 EM 算法不同迭代的图中可以看出，高斯模型被不断地更新和细化，以适应数据，算法在达到我们指定的最大迭代（$n\_iters = 100$）之前收敛 ($tol = 1e-4$)。

5.3 集群数 n_clusters

与 K-means 类似，GMM 要求用户在训练模型之前指定集群（clusters）的数量。在这里，我们可以使用 赤池信息量准则 (Akaike information criterion, AIC) 或 贝叶斯信息量准则 (Bayesian information criterion, BIC) 来帮助我们做出这个决策。设 L 为模型似然函数的最大值，k 为模型中估计参数的个数，n 为数据点的总数。

AIC 和 BIC 的计算公式如下：

$$
\begin{aligned}
AIC &= 2k - 2In(\hat{L}) \\
BIC &= kIn(n) - 2In(\hat{L})
\end{aligned}
\tag{10}
$$

集群数为 3 时，AIC 和 BIC 都取得最小值，即 k=3 时，集群效果较好。由第 2 节的数据产生可知，该结论正确。

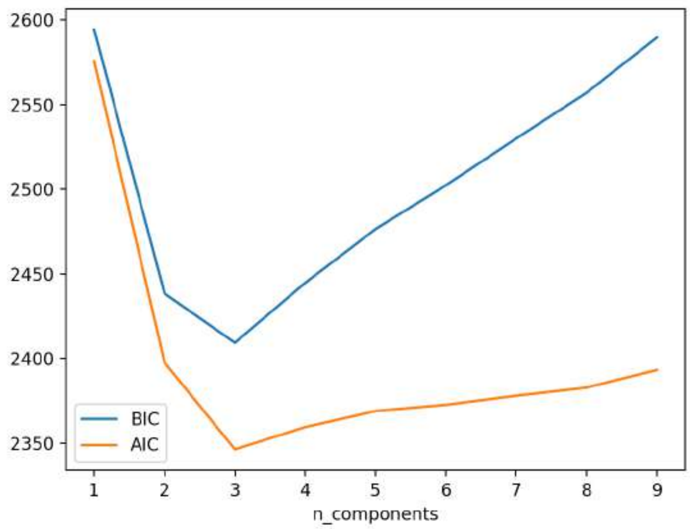

图 3: Caption

5.4 初始化参数

EM 算法与初值的选择有关，选择不同的初值可能得到不同的参数估计值。EM 算法只能保证参数估计序列收敛到对数似然函数序列的稳定点，不能保证收敛到极大值点。所以在应用中，初值的选择非常重要，常用的办法是选取几个不同的初值进行迭代。在本例中，选择不同的随机数种子，高斯分布初始的均值不同 (中心点不同，图4)，对模型的收敛速度（表2）有不同程度的影响，对收敛后的模型性能无显著影响。

seed	converged iteration	SC	CHI	DBI
4	61	0.5159	459.4272	0.6191
10	58			
123	237			
321	46			

表 2: 不同初值选择的对模型性能的影响

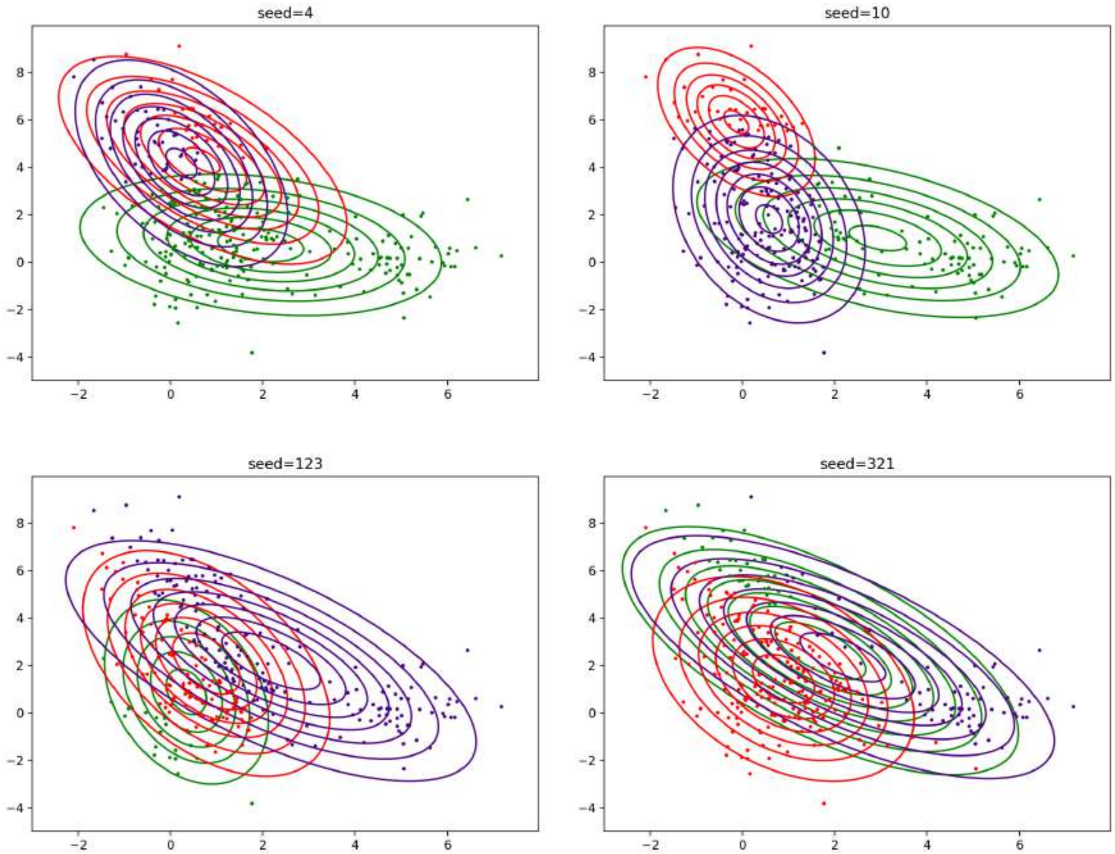

图 4: 随机数种子不同的初始高斯分布

5.5 数据集分布

π_i	n_iters	SC	CHI	DBI
[0.1, 0.1, 0.8]	478	0.1799	185.2766	1.7462
[0.2, 0.2, 0.6]	117	0.5092	399.3886	0.6213
[0.3, 0.3, 0.4]	86	0.492	493.9236	0.6401
[0.33, 0.33, 0.33]	48	0.5326	572.1099	0.6159

表 3: 不同权值对模型性能的影响

GMM 是由多个高斯分布组成的模型，其总体密度函数为多个高斯密度函数的加权组合。由表3可知，高斯分布较均匀时，模型性能更好，收敛速度更快。

5.6 数据集大小

- 随着数据集规模的增大，模型收敛的最大迭代次数抖动剧烈。原因是在数据集规模变化的时候，即使固定了随机数种子，得到的初始化参数是不同的，而 EM 算法对初值是敏感的，因此此时存在两个自变量：数据集大小和初始化参数，无法严格地控制变量。

- 随着数据集规模的增大，SC 的值总体呈下降趋势。原因可能是，数据集规模增大，对集群内的距离 a 影响较大，集群的中心点变化不大，因此对集群间的距离 b 影响较小。由公式7 可知，SC 由 a 值和 b 值决定，分子变小，分母变大，于是 SC 值呈下降趋势。

- 随着数据集规模的增大，DBI 的值总体呈上升趋势。同理，随着数据集规模变大，s_i 变大，d_{ij} 变化较小，R_{ij} 变大，由公式9可知，DBI 变大。

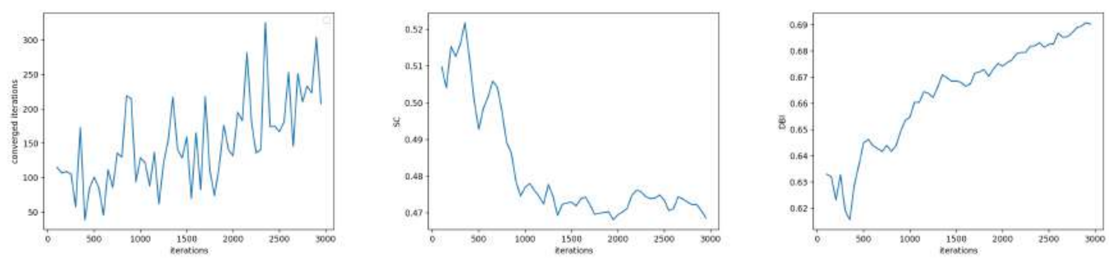

图 5: 数据集大小对模型性能的影响

5.7 数据集重合程度

调整第 2 节中高斯分布的均值，使数据集重合程度逐渐增大（重叠程度：a<b<c）。实验结果表明，随着数据集重合度的增大，数据变得不可分，模型收敛速度变慢，各项评价指标呈变差趋势。

experiment	n_iters	SC	CHI	DBI
a	41	0.5446	633.172	0.5946
b	58	0.5159	459.4272	0.6191
c	161	0.2776	127.1402	1.2578

表 4: 数据集重合程度对模型性能的影响

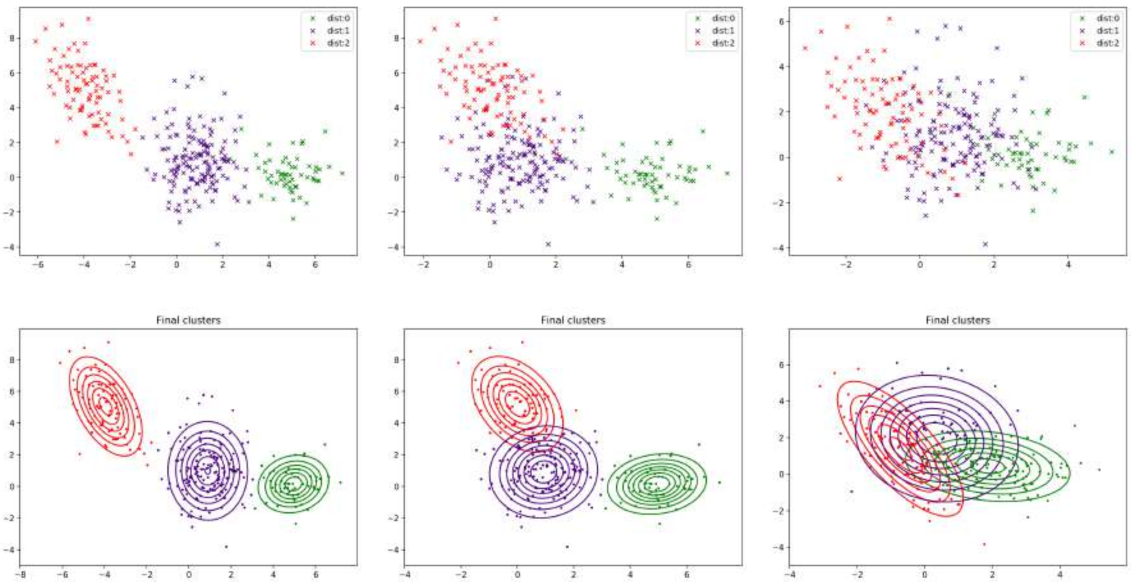

图 6: 数据集重合程度对模型性能的影响

6 代码运行方式

```
python3 source.py
```

7 总结

在第1节中，对问题进行了描述；在第 2 节中，介绍了数据集的生成方式，并给出各个高斯分布的具体参数；在第 3 节中，对高斯混合模型进行了基本介绍；在第 4 节中，详细介绍了 EM 算法的步骤及其代码实现：(1) 初始化 (2)E 步 (3)M 步 (4) 收敛条件；在第 5 节中，首先给出了无监督聚类算法的评价指标，再从迭代次数、集群数、初始化参数、数据集分布、数据集大小、数据集重合程度 6 个方面对模型性能进行详细的评估，并对给出相应的分析。最后在第 6 给出代码的执行方式。