Report of Assignment 3

PRML-Spring20-FDU

Part 1：构造无标签的聚类数据集

描述：

使用三个二维高斯分布。其中，高斯分布 A 的默认参数为均值μ=[0,0]，协方差矩阵Σ = [[10, 0],[0, 10]]，高斯分布 B 的默认参数为μ=[15,12]，Σ=[[3, 0.5], [0.5, 2]]，高斯分布 C 的默认参数为μ=[7, -15]，Σ=[[3, 2], [2, 3]]。默认情况下，从三个高斯分布总**共采样 100000 次**，设从高斯分布 A、高斯分布 B、高斯分布 C 采样的概率分别为 0.28、0.48、0.24。

为了方便数据集构造，实际上先计算从高斯分布 A、B、C 采样的次数。即在 [0,1]上的均匀分布进行 100000 次采样，记 size_A 为其中小于 0.28 的样本个数，size_C 为其中大于 1-0.24 的样本个数，size_B 为剩下的样本个数。然后，使用 numpy 内置的 random.multivariate_normal 依次生成服从 A、B、C 参数、样本个数为对应 size 的正态分布。

由于邱老师在上课时提到，EM 聚类中的每个样本点，并不单独属于某一个类，而是以一定的概率属于某一个类。为了考察**高斯分布之间重合较大**的情形，**额外构造了一套参数**，其中，高斯分布 A 的默认参数为均值μ=[0,0]，协方差矩阵Σ = [[10, 0],[0, 10]]，高斯分布 B 的默认参数为μ=[8,5]，Σ=[[3, 0.5], [0.5, 2]]，高斯分布 C 的默认参数为μ=[5 -6]，Σ=[[3, 2], [2, 3]]。

由于 Requirements 里要求提供不大于 20Kb 的 data 文件，故提供了参数为默认套餐/第二套参数，而采样次数变为 **25000 次**的数据集 dataset_default_small.data 和 dataset_second_small.data。2 个数据集的大小加起来**不超过 20Kb**。

下列图像中，左边呈现了数据点的标签，而**右边是舍弃标签的结果**。

数据集可视化：

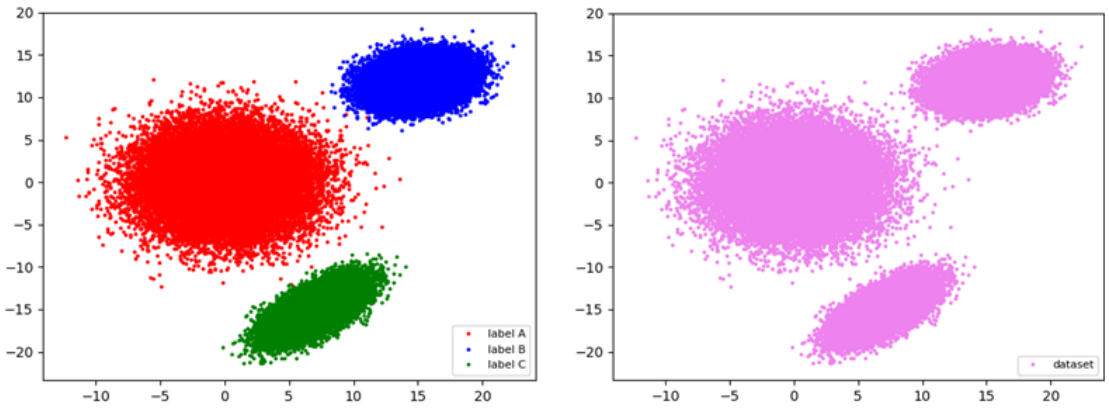


图1. 默认参数，从高斯分布A、B、C中共采样100000次的结果

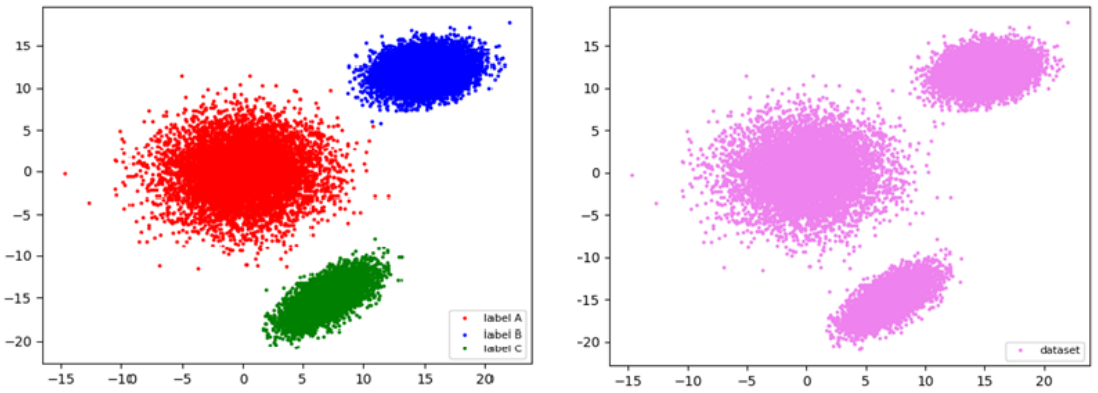


图2. 默认参数，从高斯分布A、B、C中共采样25000次的结果

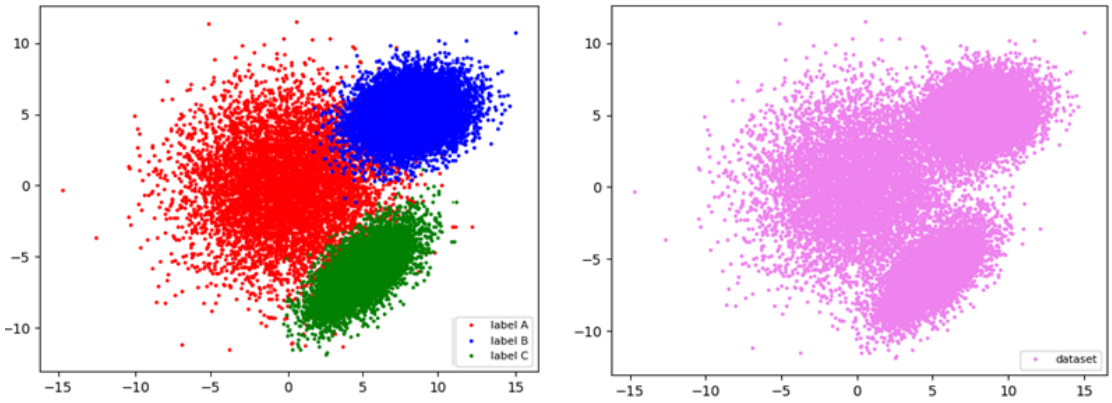


图4. 第二套参数，从高斯分布A、B、C中共采样25000次的结果

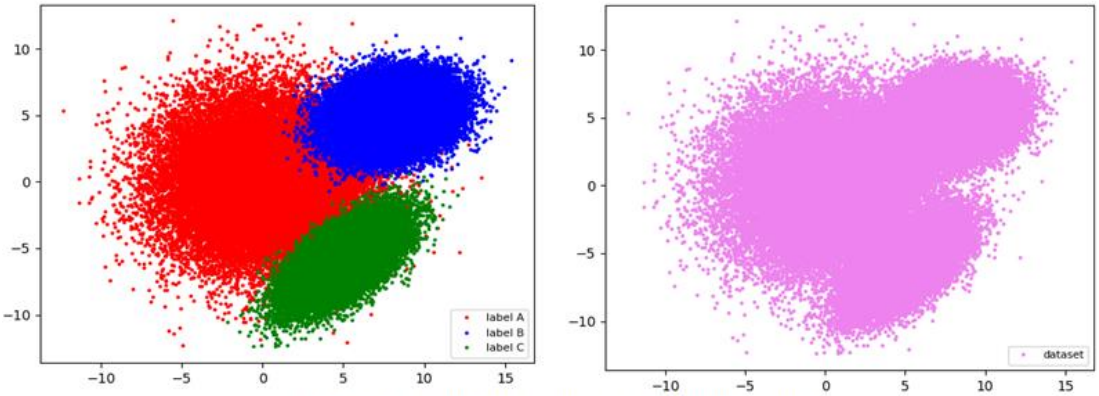


图3. 第二套参数，从高斯分布A、B、C中共采样100000次的结果

Command Lines:

e.g. python source.py 按照上述**默认参数**，进行 100000 次采样，**并运行 GMM**。

e.g. python source.py --load_data --dataset dataset_default_small.data 读取按照默认参数采样 25000 次的小数据集，并运行 GMM。

e.g. python source.py --mean_B 8 5 --mean_C 5 -6 使用第二套参数，进行 100000 次采样并运行 GMM。

e.g. python source.py --load_data --dataset dataset_second_small.data 读取按照第二套参数采样 25000 次的小数据集，并运行 GMM。

相关参数解释：

--mean_A --mean_B --mean_C：高斯分布 A、B、C 的均值μ，--mean_A x y 即设置高斯分布 A 的均值为[x, y]

--cov_A --cov_B --cov_C：高斯分布 A、B、C 的协方差矩阵Σ，--cov a b c d 即设置高斯分布 A 的协方差矩阵为[[a, b], [c, d]] 。其中 a、b、c、d 应满足协方差矩阵的性质，如 b=c。

--load_data：如果使用该参数，即从数据集文件里读取数据集，否则将生成数据集并存入文件

--dataset：使用 load_data 参数时，读取 dataset 的文件名，或不适用 load_data 参数时，生成数据存储的文件名。

Part 2：使用 GMM 进行无监督聚类

描述：

使用高斯混合模型，对上面构造的无标签混合高斯分布样本进行聚类，即反复进行 EM 聚类中的 E 步、M 步，直到模型预测的结果收敛。为了**定义模型收敛**，定义**指标 L 为 2 次预测结果间，3 个正态分布上数据点个数之差的绝对值之和**，并定义**当 L < 数据集大小 / 100000 时，模型收敛**。这个指标在未收敛时**无法表示聚类的优良性**，但却**可以体现模型是否收敛**。

在具体的实现中，流程主要分为以下几步：

1. **生成**数据集，见 Part1
2. **初始化** GMM 参数，主要包括：
   （1）高斯分布的维数 input_dims = 2
   （2）高斯分布的个数 num_classes = 3
   （3）3 个高斯分类的均值 mean，用**二维正态**分布 N（**0**, **I**）生成，其中 0 是二维零向量，I 是 2x2 的**单位阵**。
   （4）3 个高斯分布的协方差矩阵 cov，均初始化为 **2x2 的单位阵**。
   （5）3 个高斯分布的先验，均初始化为 1/num_classes = 1/3
3. **E 步**，根据当前的先验概率、高斯分布参数，计算第 i 个样本属于类别 k 的后验概率。

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

其中，γ_{ik} 代表第 i 个样本属于类别 k 的后验概率，π_k 为类别 k 的先验概率，$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ 为第 i 个样本在第 k 个正态分布下的概率密度，$\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ 分别表示其均值、协方差矩阵。

4. **M 步**，更新参数，包括：高斯分布的先验概率、均值、协方差，以及属于各类别样本个数的期望值。

$$n_k = \sum_{i=1}^{N} \gamma_{ik}$$

$$\mu_k = \frac{1}{n_k}\sum_{i=1}^{N}\gamma_{ik}x_i$$

$$\Sigma_k^2 = \frac{1}{n_k}\sum_{i=1}^{N}\gamma_{ik}(x_i-\mu_k)^2$$

$$\pi_k = \frac{n_k}{N}$$

其中，n_k 为属于类别 k 的样本个数的期望值。

5. 判断模型是否收敛。即计算 L = np.sum(abs(self.nk - self.pre_nk))，如果 L < 数据集大小 / 100000，则收敛。

6. 预测与绘图（如果需要），即利用 E 步计算出来的后验概率，取其中最大项作为预测的类别，并进行绘图。

7. 重复 3-6 步，**直到模型收敛**。

Command Lines:

e.g. python source.py 按照默认参数，进行 100000 次采样，并运行 GMM。仅当模型收敛后，绘制预测结果。

e.g. python source.py --load_data --dataset dataset_default_small.data 读取按照默认参数采样 25000 次的小数据集，并运行 GMM。仅当模型收敛后，绘制预测结果。

e.g. python source.py --mean_B 8 5 --mean_C 5 -6 使用第二套参数，进行 100000 次采样并运行 GMM。仅当模型收敛后，绘制预测结果。

e.g. python source.py --load_data --dataset dataset_second_small.data 读取按照第二套参数采样 25000 次的小数据集，并运行 GMM。仅当模型收敛后，绘制预测结果。

（以上同 Part. 1）

e.g. python source.py --print_every 按照默认参数，进行 100000 次采样，并运行 GMM。在每一次 EM 后，都绘制预测结果。

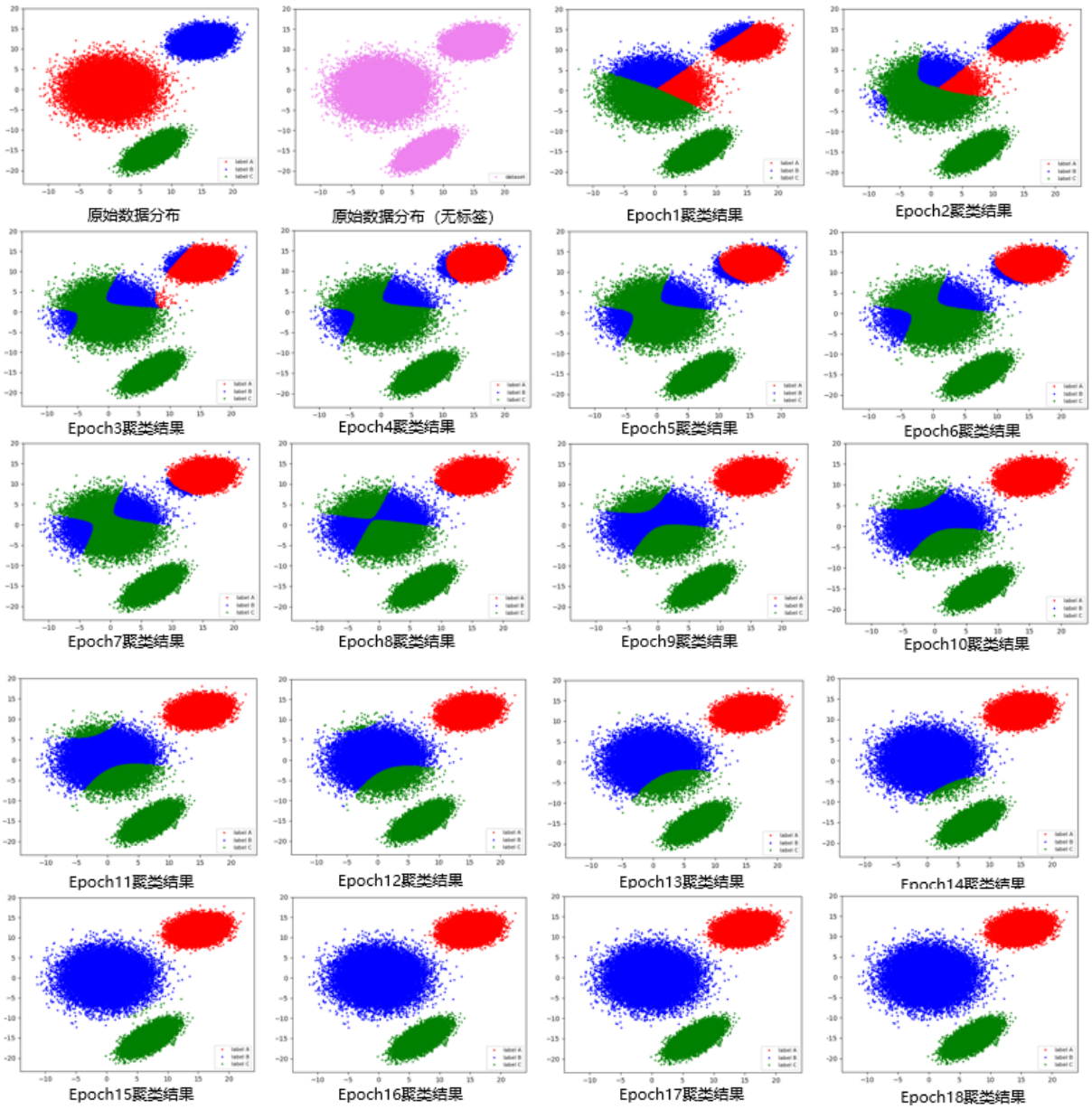
相关参数解释：

--print_every 默认情况下，仅当模型收敛后，绘制预测结果。如果使用 print_every，则在每次 EM 后，都会绘制预测结果。

实验结果及分析:

我们**称一次 E 步、M 步为 1 个 EPOCH**。

按照默认参数，进行 100000 次采样，并运行 GMM，使用 print_every 显示聚类的过程。



模型在第 10 次 EM 后，可以较好地分类出右上角的高斯分布；在第 17 次 EM 后，可以较好地区分蓝色和绿色的高斯分布的边界（第 16 次 EM 后，边界上还是蓝绿混杂。）第 18 次聚类后，L<1，收敛。

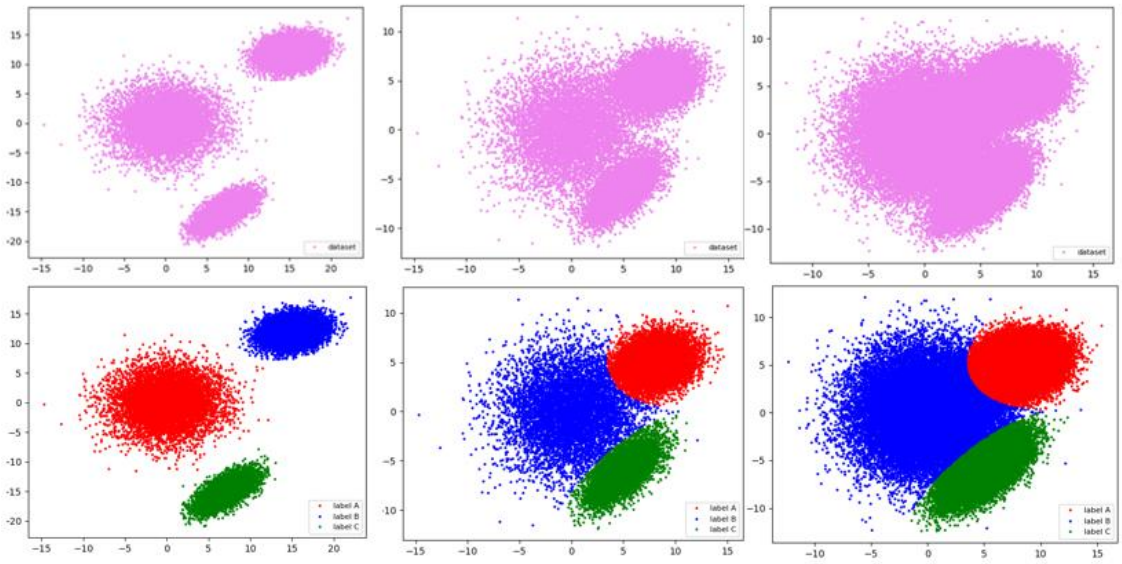
由于这是一个无监督的聚类问题，因此最终模型虽然较好地对混合高斯分布进行了聚类，但无法还原原样本的标签顺序。

Epoch	1	2	3	4	5	6	7	8	9
L	4250.56	5283.53	7952.59	8822.06	3942.80	2227.78	2312.08	3678.90	4898.87
Epoch	10	11	12	13	14	15	16	17	18
L	4810.16	4472.18	4396.14	4514.21	4441.94	3378.33	1089.71	43.03	0.59

按照默认参数，进行 100000 次采样，各 EM 阶段后的收敛指标 L

需要重申的是，**上面定义的 L 不能作为模型未收敛时评价模型聚类能力的标准，仅能作为评价模型是否收敛的标准。**

调整采样次数，并切换第二套参数，进行聚类，得到的结果如下：



可以看到，在采用第二套参数时，由于不同正态分布之间的边界模糊，有重叠，导致分类困难，因此模**型收敛所花的 Epoch 比默认参数多**，约为其 3-4 倍。而数据集大小并不会影响收敛所需的 Epoch（这基于我们对收敛的定义）。

对于重叠部分，在图像上看到的结果是面积较小的高斯分布盖住了面积较大的高斯分布。这是因为面积较大意味着协方差较大，虽然覆盖范围广，但离中心越远，其概率密度越小，导致在重叠部分，面积大的高斯分布的概率密度远小于面积小的高斯分布的概率密度，因此呈现出面积小的高斯分布覆盖住面积大的高斯分布的结果。

总的来说，**GMM 模型可以较好地完成无标签（混合高斯分布）数据的聚类问题**，并得到良好平滑的边界。当不同高斯分布间的边界清晰时，聚类的结果边界也相当清晰。

实验感想：

纸上得来终觉浅，在课上听 EM 模型，怎么都觉得数学公式抽象，模模糊糊，但实际动手实现 GMM 模型后，发现 GMM 模型也不过如此，迭代时也只需要几步。通过本次实验，我得以对 EM 模型有了较为深入的理解。

此外，在做本次实验时，感觉对矩阵运算（尤其是 numpy 中的广播、按元素乘、矩阵乘的使用）较为得心应手，而以前我虽然能使用 CNN、RNN、Transformer 等模型，实际上却不过是调包，对内部的数学运算较为生涩。想来这都得益于在 Assignment-1 中自己动手实现了 FNN（尤其是其反向传播）的矩阵运算。这几个实验，都让我学习到了很多。