PRML Assignment-3 Report

> This document contains the report of assignment-3 in PRML course.
>
> In this assignment, you are required to design a Gaussian Mixture Model.

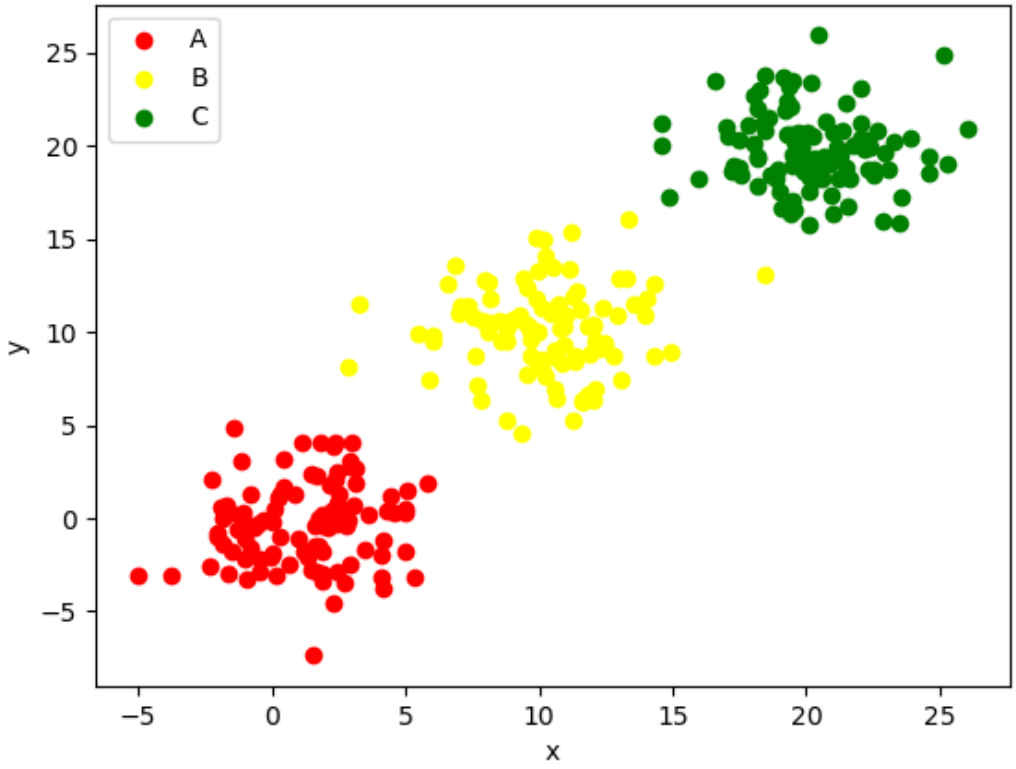
Table of Contents

Part I. Generate Data

> In this part, you are required to construct a clustering dataset without any labels.

考虑到最终要构建的模型是高斯混合模型，为了方便，我直接使用了assignment-1的生成数据函数，生成若干个高斯分布的二维数据集，为了最后检验聚类的效果，我还是带上了初始的标签。

生成三个100个点的数据集可视化结果如图所示：



Part II. Design GMMs

> In this part, you are required to design Gaussian Mixture Models to finish the unlabeled clustering task.

这部分的任务就是构建一个高斯混合模型对生成的数据集进行无标签分类。

1. 模型概述

高斯混合模型(Gaussian Miture Models)是一种无监督聚类模型。GMM认为不同类别的特征密度函数是不一样的；因此，GMM为每个类别下的特征分布都假设了一个服从高斯分布的概率密度函数：

$$P(x|c_k) = \frac{1}{\sqrt{2\pi}\sigma_k} exp(-\frac{(x-\mu_k)^2}{2\sigma_k^2})$$
$$P(x|c_k) \sim N(\mu_k, \sigma_k)$$

而对于数据集来说，数据集可能由多个类混合而成，所以数据中特征的概率密度函数可以使用多个高斯分布的组合来表示：

$$\begin{aligned} P(x) &= \sum_{k=1}^{K} P(c_k)P(x|c_k) \\ &= \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k) \end{aligned}$$

其中π_k为类分布概率，也可以看作是各个高斯分布的权重系数，也称为混合系数，其中$\sum_{k=1}^{K} \pi_k = 1$。

2. Expectation-Maximization

上面已经给出了模型，现在给定一组数据X，需要得到一组参数μ, σ，使得在这个组参数下观测数据出现的概率最大。通过最大似然估计可以得到：

$$\prod_{i=1}^{N} P(x_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k)$$

取对数，

$$\sum_{i=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(x_i|\mu_k \sigma_k)\}$$

即可得到参数。然后可以计算某个样本对应的类，由贝叶斯公式得到：

$$\begin{aligned} P(c_k|x_i) &= \frac{P(c_k, x_i)}{P(x_i)} \\ &= \frac{P(x_i|c_k)P(c_k)}{P(x_i)} \\ &= \frac{\pi_k N(x_i|\mu_k, \sigma_k)}{\sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k)} \end{aligned}$$

得到后验概率之后，我们可以得到某个类别的分布概率与该类别下的统计量：

$$\begin{aligned} N_k &= \sum_{i=1}^{N} P(c_k|x_i) \\ \pi_k &= \frac{N_k}{N} = \frac{1}{N} \sum_{i=1}^{N} P(c_k|x_i) \\ \mu_k &= \frac{1}{N_k} \sum_{i=1}^{N} P(c_k|x_i) x_i \\ \sigma_k &= \sqrt{\frac{1}{N_k} \sum_{i=1}^{N} P(c_k|x_i)(x_i - \mu_k)^2} \end{aligned}$$

其中N_k是类别k的频率期望。

以上两步计算实际对应了期望最大化算法的E-step跟M-step。

3. 训练模型

随机生成k个高斯分布，然后不断迭代EM算法直至似然函数变化不再明显或者到达最优迭代次数。

3.1 E-step

在给定的多维高斯分布下，计算各样本属于各个类别的概率：

$$P(c_k|x_i) = \frac{\pi_k P(c_k|x_i)}{\sum_{k=1}^{K} \pi_k P(c_k|x_i)}$$

3.2 M-step

根据概率重新计算更优的高斯参数：

$$N_k = \sum_{x=1}^{N} P(c_k|x_i)$$
$$\pi_k = \frac{N_k}{N}$$
$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} P(c_k|x_i) x_i$$
$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} P(c_k|x_i)(x_i - \mu_k)^T (x_i - \mu_k)$$

4. 预测

在最开始训练的时候，将数据集按照一定比例分割为训练集和测试集，测试集专门用于预测。训练好的模型，将测试集的数据代入已经训练好的模型，选择使得其概率最大的某个模型作为该测试数据所属的类别。得到训练结果。

5. 评估

对于一个良好的无监督聚类模型，其评估方式有许多种，这里我选择的是最简单朴素的一种方法——判定预置标签与预测标签是否相同，统计测试集的正确率。但是此处需要注意一个问题，对于无监督聚类得到的结果，无法与预置标签的名称一一对应。即在预置类别标签中属于A类的数据，很有可能在分类的时候被分到一个名称叫B的类别中，如果此时直接用B跟预置标签的B进行比较统计准确率，显然会得到错误的结果，无法反映真实的聚类结果。于是我的想法是，对已经预测好的结果与类别名称的排列组合分别进行比较，选择与预置标签重合程度最高的匹配方式作为最终的结果。

此外，除了这种朴素的方法以外，还有一种方法是计算类内距离和类间距离，与初始数据进行比较。这种方法的好处就是类与类之间不必区分谁是谁，只关心类内的紧凑程度和类间的疏远程度。

另外，还有余弦相似度等等方法用以评估聚类算法的精确性。此处受限于时间关系并未一一实现。

6. 结果

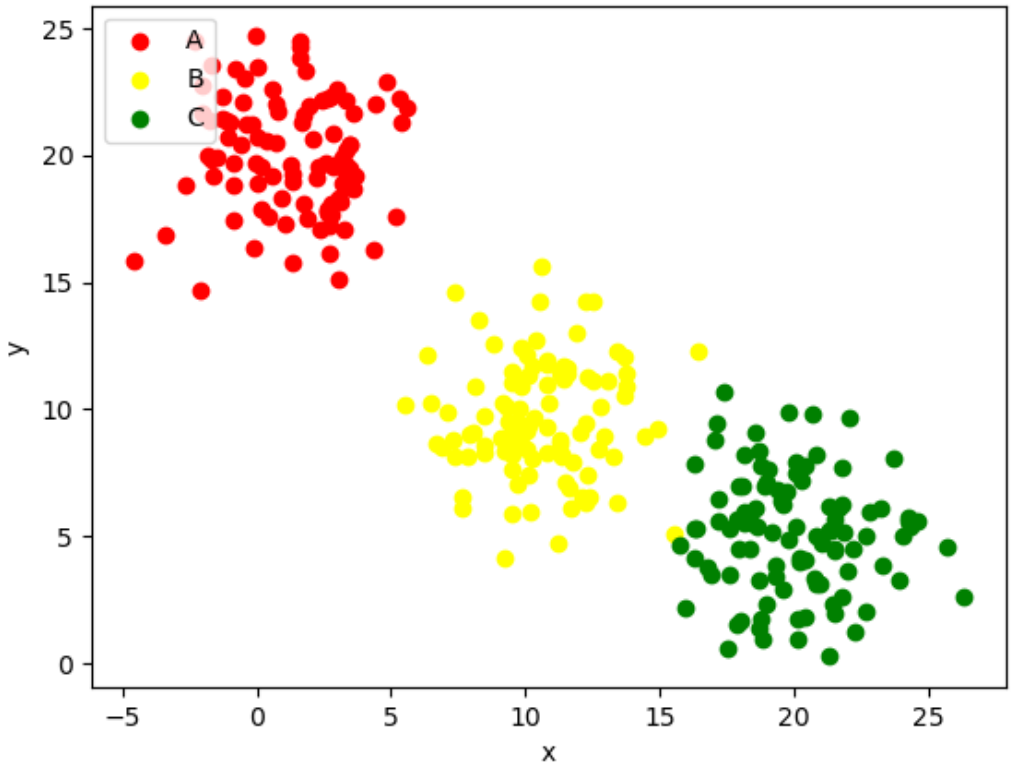
经过反复实验，我得到了以下结果：

6.1

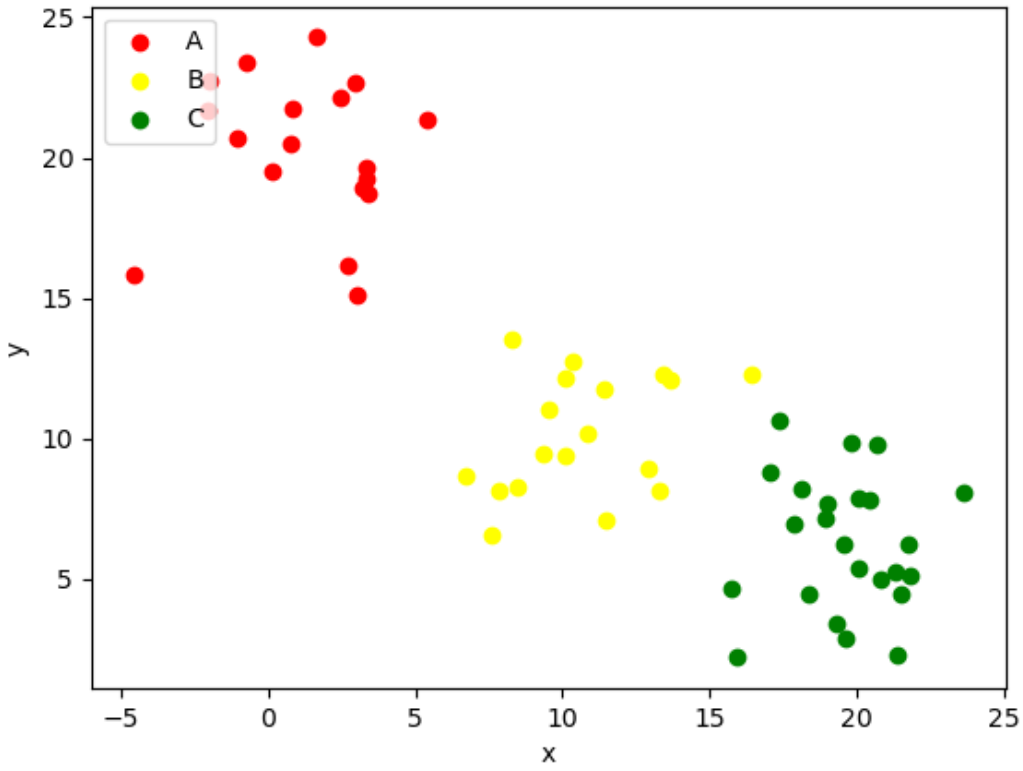
当数据集类与类之间间隔比较清晰时，得到的结果比较好。

```
    mean1 = [1, 20]
    mean2 = [10, 10]
    mean3 = [20, 5]
    d = 5
    cov1 = [[d, 0], [0, d]]
    cov2 = [[d, 0], [0, d]]
    cov3 = [[d, 0], [0, d]]
```

初始数据():



预测结果(20%测试集):




```
C:\ProgramData\Anaconda3\envs\PRML\python.exe C:/Users/xrnie/PycharmProjects/
Iter: 0
Accuracy: 44/60, 0.733333
Iter: 100
Accuracy: 60/60, 1.000000
[1 2 2 1 0 0 2 0 0 0 1 2 0 2 2 1 1 0 0 0 2 1 0 1 2 1 0 2 0 2 2 1 1 0 0 1 0
 0 2 2 2 2 1 1 2 0 0 1 1 2 0 1 0 0 0 1 2 1 0 0]
Accuracy: 60/60, 1.000000
```

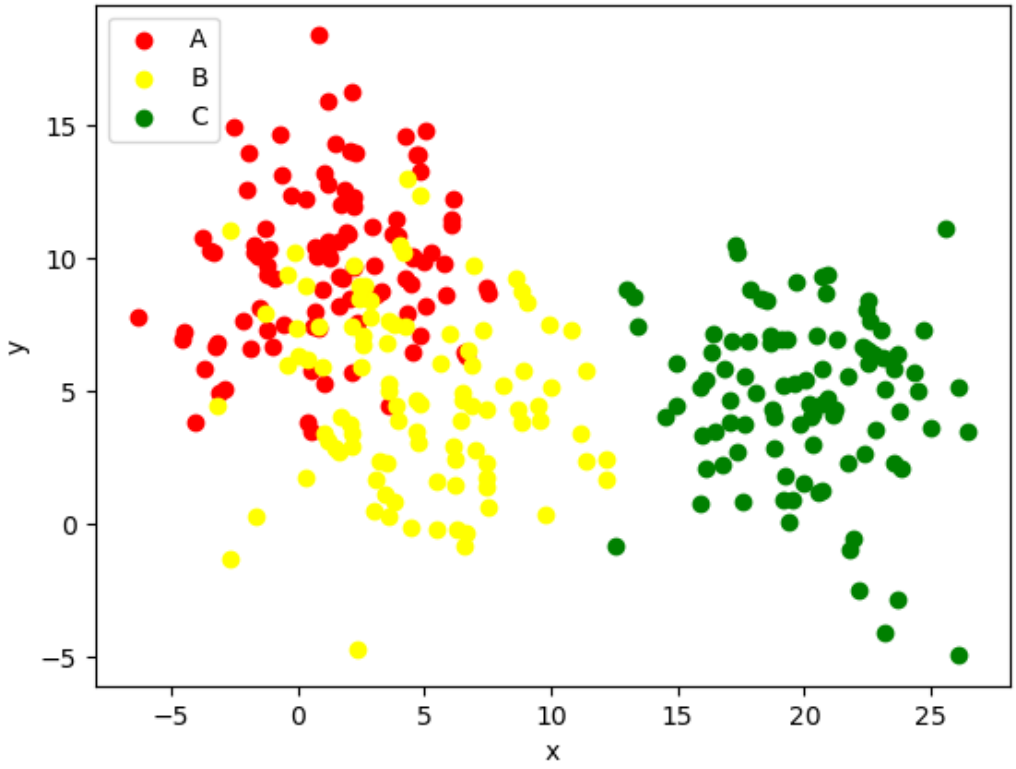
注：上述结果训练执行了200个iteration，每隔100个iteration对模型进行一次测试，可以看到100次iter之后，模型在测试集上已经可以达到100%的正确率。

6.2

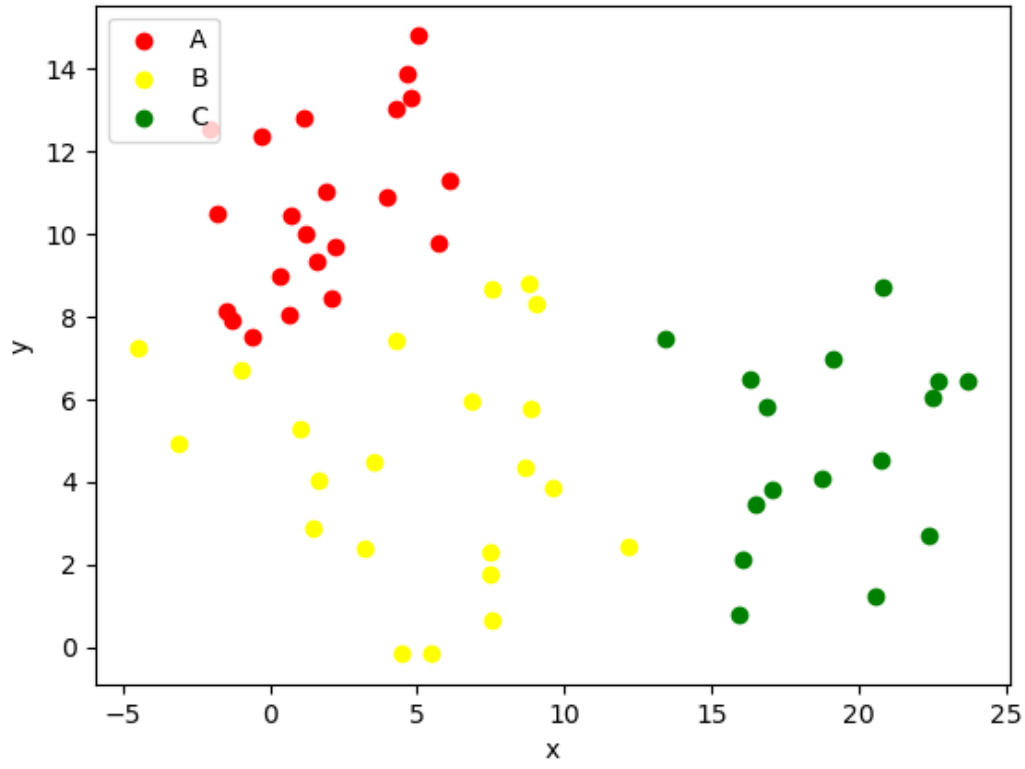
当数据集的不同类别之间距离比较近，或者不同类之间的数据混合程度比较高时，得到的结果只能算差强人意。

```
    mean1 = [1, 10]
    mean2 = [5, 5]
    mean3 = [20, 5]
    d = 10
    cov1 = [[d, 0], [0, d]]
    cov2 = [[d, 0], [0, d]]
    cov3 = [[d, 0], [0, d]]
```

数据集：


A
B
C
15
10
5
0
−5
y
−5
0
5
10
15
20
25
x

预测结果(20%测试集):


A
B
C
14
12
10
8
6
4
2
0
y
−5
0
5
10
15
20
25
x

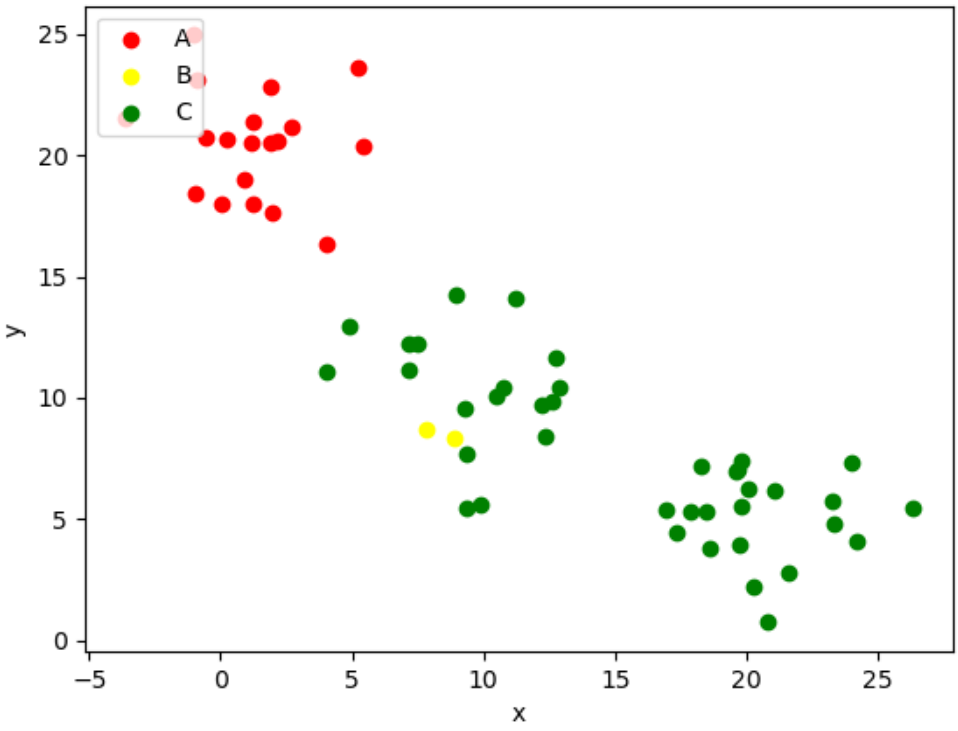
```
C:\ProgramData\Anaconda3\envs\PRML\python.exe C:/Users/xrnie/PycharmProjec
Iter: 0
Accuracy: 39/60, 0.650000
Iter: 100
Accuracy: 51/60, 0.850000
[2 2 2 0 0 2 0 0 0 1 0 0 0 2 1 0 1 1 0 1 2 2 2 0 2 1 2 1 0 2 0 2 2 0 2 0 1
 1 2 1 1 1 2 1 0 2 0 1 0 1 2 0 0 1 2 2 0 0 2 2]
Accuracy: 51/60, 0.850000

Process finished with exit code 0
```

可以看到，此时的准确率只有85%。

7. 思考和提升

综合来看，模型取得了不错的结果，但其精确度仍然很大程度上受制于数据集本身的特点。在实验过程中，我发现了几个需要注意的问题：

1. 模型的**初始化参数**极其重要，查阅资料后，我将高斯分布的初始参数设置成数据集中最小值和最大值中间的随机整数，取得了不错的效果。
2. 对于**模型的评估方法有待提高**，此处受限于时间关系我只用了原始的标签匹配的方法评估模型的精确度，虽然通过可视化预测结果可以看到，这种评估方法的准确度和相关性还是挺好的。但是这种方法的弊端在于，为了使得预测的结果中的类别与实际类别匹配，必须消耗`k!`的复杂度进行匹配，k为类别的个数。在大规模的聚类分析中，这种评估方法显然是十分低效的。
3. 当模型面对的数据比较分散，或者不同类别之间的交叉区域比较模糊的时候，模型容易陷入**局部最优解**。最后预测出来的结果出现较大的失误，比如：



从这个例子我们可以看到，明显B类别的诸多数据点均被归类于C中，由此可见消除局部最优解是一件亟待解决的事情。