# PRML Assignment 3 Report

## 任务

在这次 assignment 中，我们需要先根据一些参数生成高斯混合模型的数据。然后写出一个模型来在没有标签的点集上进行训练，给出该点集的一个分割。

## 生成数据

使用 gen_data 来生成数据。默认从三个高斯分布中抽取样本，高斯分布的参数取 source.py 中的值。save_data 来把生成的样本点和标签保存到 data.data 和 label.data 文件，使用 load_data 函数来读取文件。

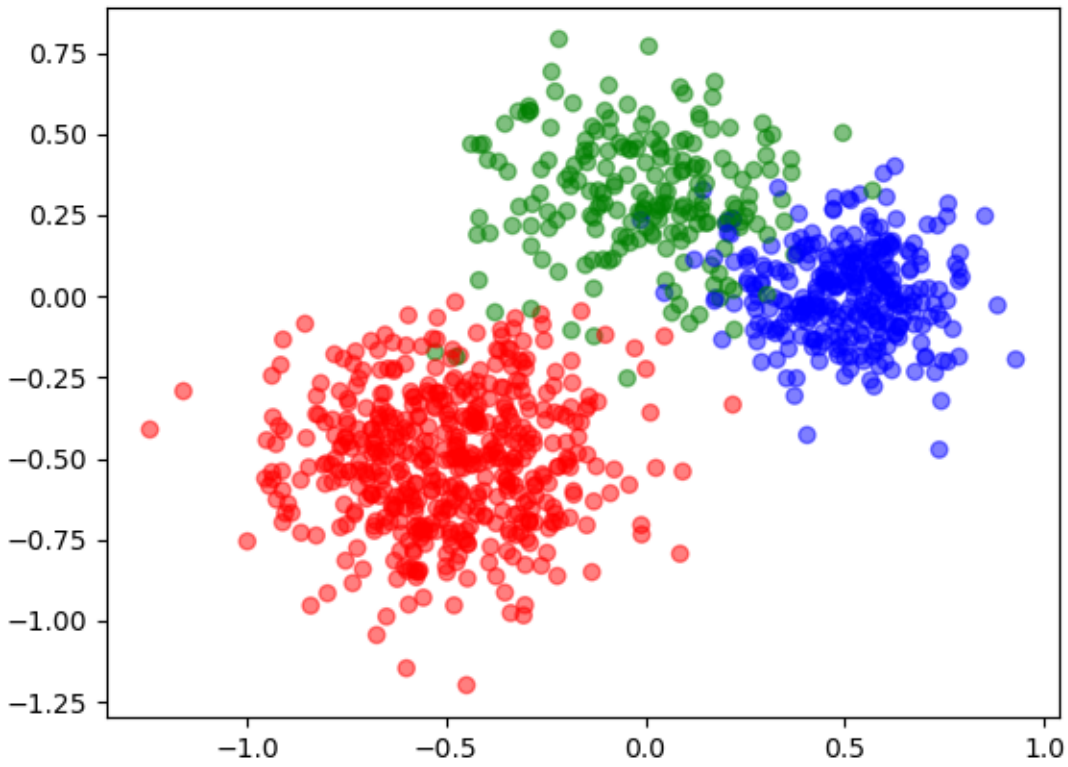



*Figure 1 原始数据。颜色代表归于哪一类，但是分类是模型学习时未知的*

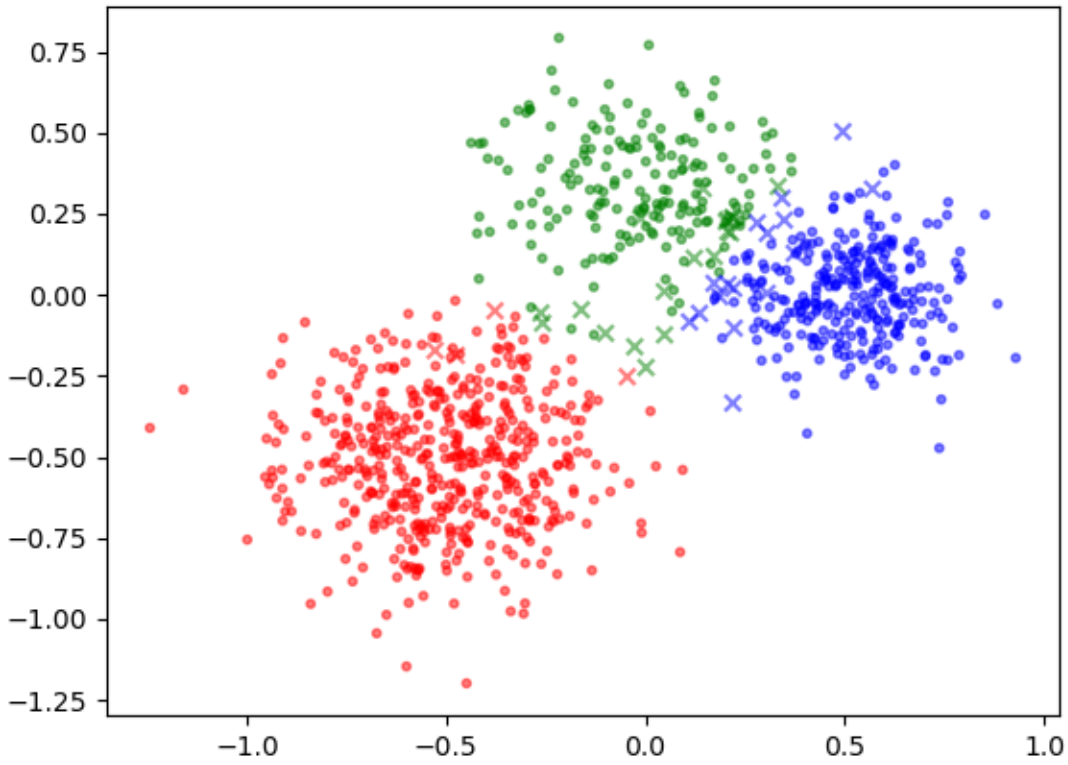
## K means 方法

K means 是一个简单的 GMM 解法，在训练时先选若干个点作为中心，离某个点最近的中心成为该点的归类。下一次迭代时使用某一类点的均值作为新的中心。判断时同样选取离点最近的中心作为标签。

在具体实现时需要注意为了防止某一类初始没有点归入，需要在原来的数据集上选取若干个不同点作为初始中心。

老师上课说 k_means 方法会倾向于给出每个类大小差不多的分布，得到了实际测试的验证。当每个类的协方差接近时效果比较好，反之效果就会不太好。

K means 迭代不到十次就达到稳定了所以这里不展示准确率曲线。除了有时会收敛到完全错误的结果外，最终正确率 0.961




## EM 算法

EM 算法的实现使用 k means 的对点的判别作为初始值。由于 k means 给出的结果可以作为$\gamma_{nk}$，我们需要从 M 步开始训练。

训练时每十步输出一次预测结果与正确值的 KL 散度。最终正确率 0.966，较之 K means 算法稍高

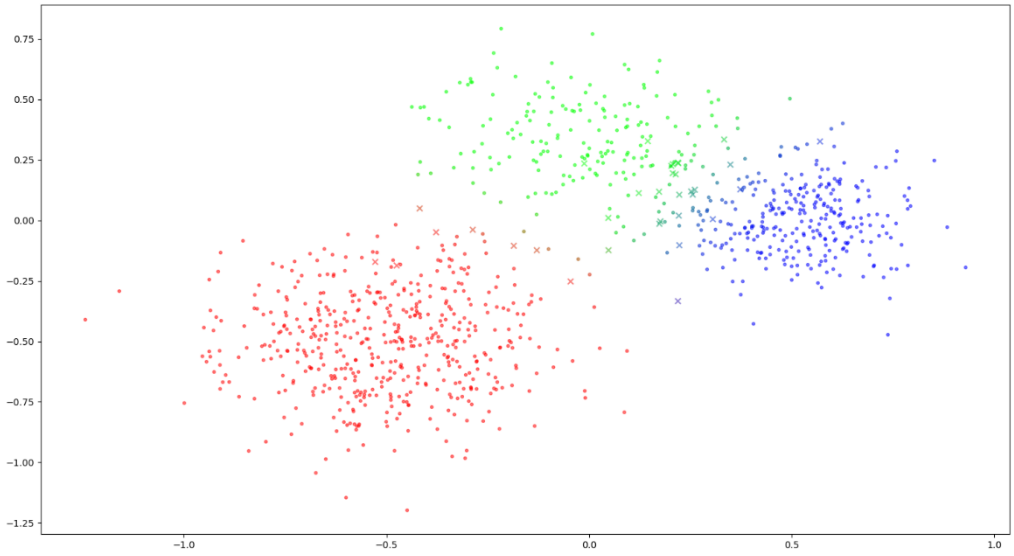



*Figure 2 EM 算法结果。颜色纯度越高致信率越大*

## KM 较劣的情况

让三个分布的协方差相差较大的同时加大样本难度。

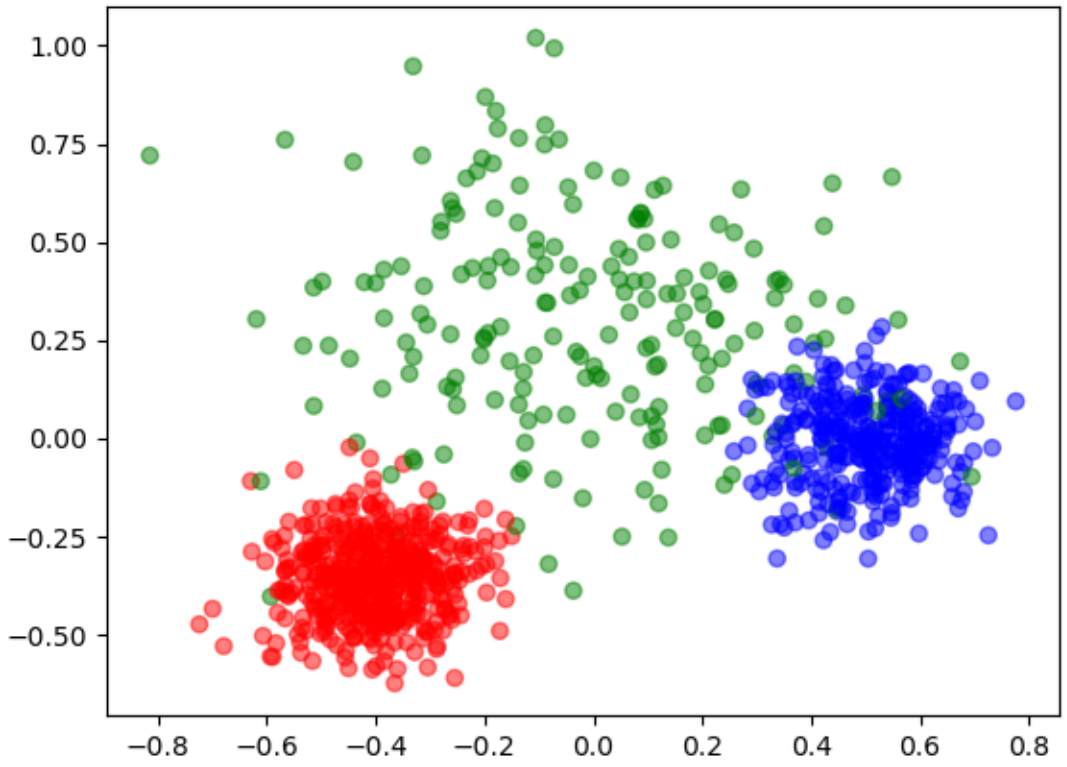



*Figure 3 KM 较差的原始样本*

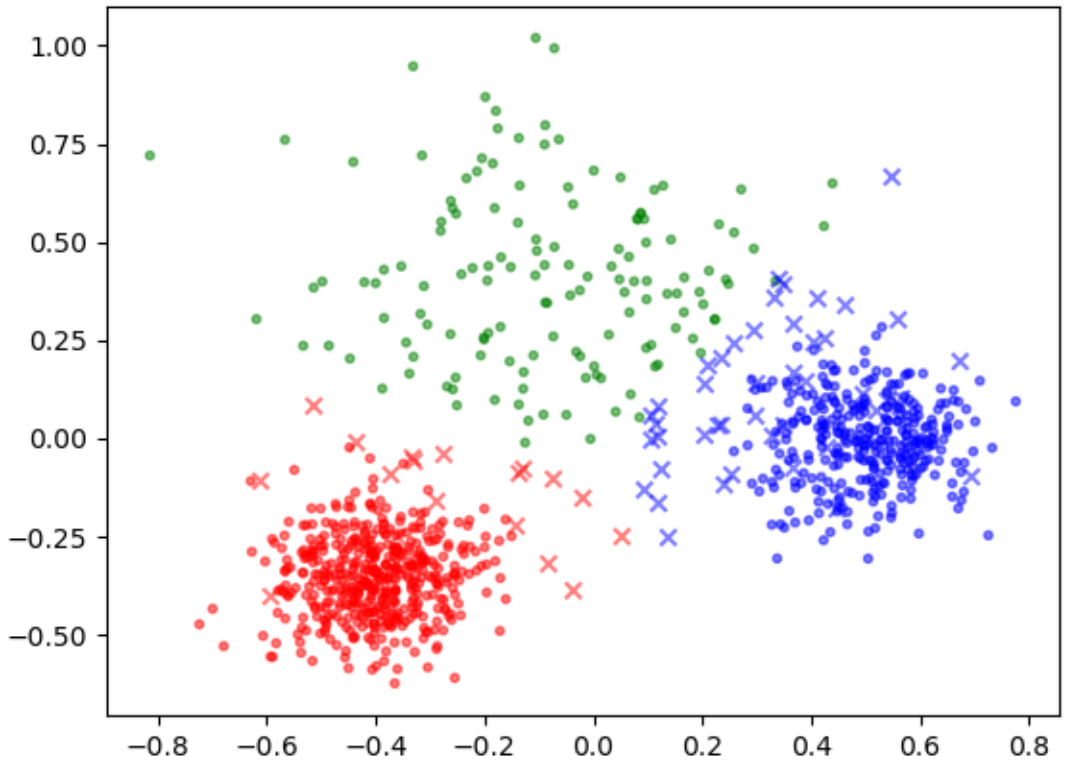



*Figure 4 KM 结果 正确率 0.936 可以看到红蓝占了过多的面积。*

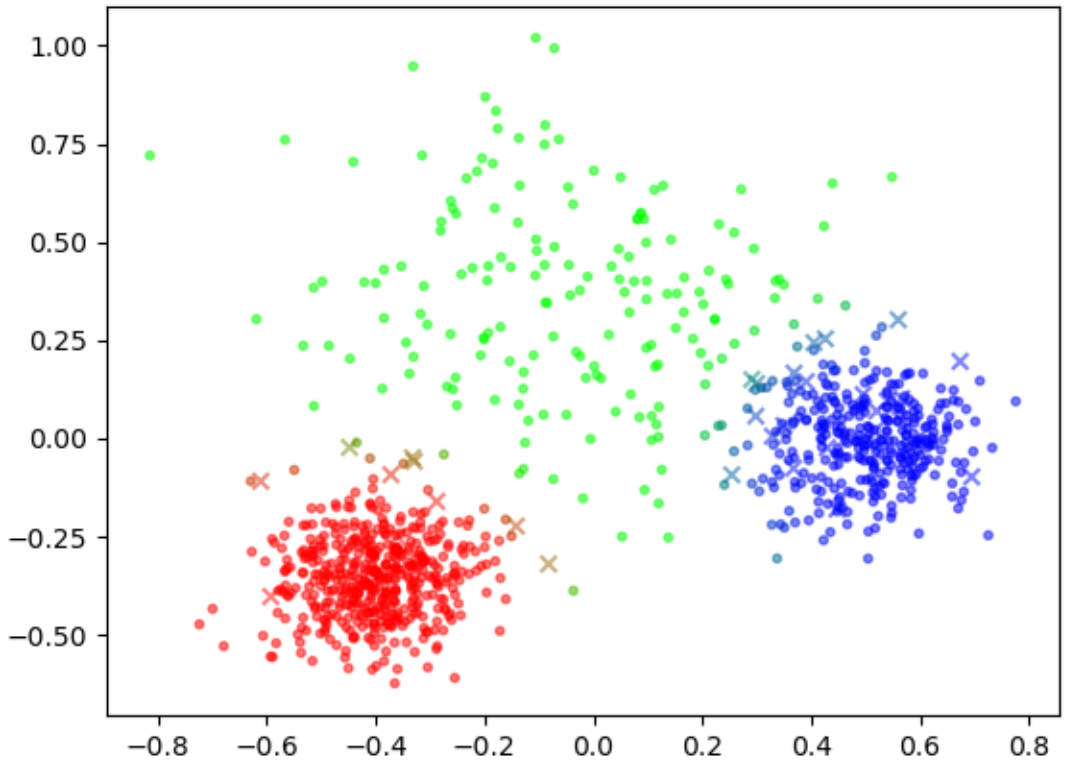



*Figure 5 EM 的结果。正确率 0.968.看起来非常接近最优解。*

## 运行代码

Python3 source.py

或者

```
from source import *
tmp = load_data()
k = 3
data, Glabel = tmp
n = len(data)
for kmResult in kmeans_train(data, k):
    pass
print(kmResult)
for pi, miu, sigma in em_train(data, k, kmeans_label):
    pass
print(pi, miu, sigma)
```

## 总结

Kmeans 解法实现简单但是有缺陷使得不能很好的处理所有的 GMM 数据。EM 算法处理 GMM 模型需要依赖一个较好的初始值，但是效果总是比 kmeans 更好。

## Reference

chap9-EM-GMM.pdf provided as in FDU