PRML Assignment3 Report

Part I 生成聚类数据集

代码见handout/datatools.py

在datatools.py中实现了数据的save，load，生成(gen)，画图(draw)等功能。

数据工作继承在了的datatools.workdata()方法中，其中带有参数need_gen 和 need_save，分别表示是否需要生成新的数据和是否需要保存生成的数据。

通过*numpy.random*中的*multivariate_normal*即可在给定的高斯分布上取样。

先生成一个简单的没有重合的高斯分布

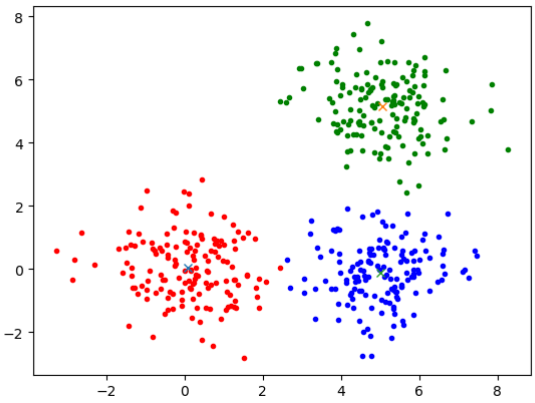
三个高斯分布的均值分别为

$$mean1 = (0, 0)$$
$$mean2 = (0, 5)$$
$$mean3 = (5, 5)$$

三个高斯分布的协方差矩阵均设置为

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

由于限制 .data文件大小不超过20k，所以在生成需要保存的数据时，分别在三个高斯分布上采样150个点，在实验中可以使用另外单独生成的更大的数据。



三个高斯分布的采样如上图所示。

Part II 生成高斯混合模型

本次实验设计使用K-means + EM算法进行聚类。

具体流程如下：

1. 从所有点中随机取3个点作为每个高斯分布的均值
2. 进行10次K-means的迭代过程
3. 进行20次EM算法的迭代过程

这里进行10次k-means算法是为了得到一个较好的初始值，帮助EM算法更快收敛。

另外设置一组对照组，只进行20次EM算法的迭代过程，不使用K-means算法。

简述一下K-means和EM算法：

K-means：

1. 随机选k个点作为中心。
2. 每次把每个点分给这k个点中距离最近的点。
3. 计算属于每个中心的点的平均值作为中心。
4. 重复2-3步，直到收敛。

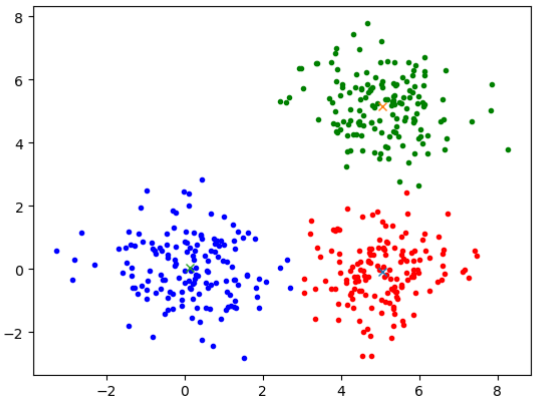
EM：

1. 随机选k个点作为高斯分布的均值。
2. 算出每个点属于每个高斯分布的概率。
3. 用概率进行加权平均，算出每个高斯分布的先验概率，均值和协方差矩阵。
4. 重复2-3步，直到收敛。

实验一 使用Part I生成的数据

这一组数据三个高斯分布数量都较少，并且没有明显的重合部分。

K-means + EM组

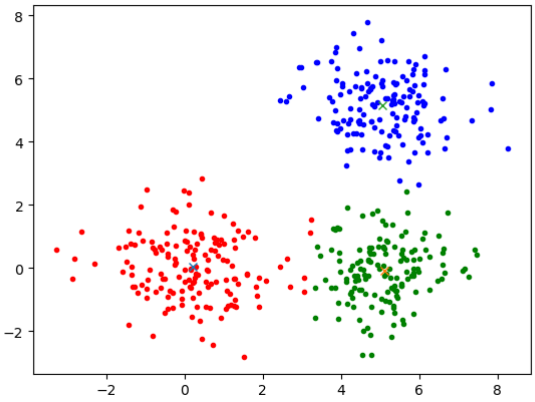


$$mean1 = (5.04617656, -0.08591989)$$
$$mean2 = (5.04315818, 5.15504626)$$
$$mean3 = (0.11859660.03575082)$$

$$cov1 = \begin{bmatrix} 0.86229846 & 0. \\ 0. & 0.93729302 \end{bmatrix}$$
$$cov2 = \begin{bmatrix} 1.0249532 & 0. \\ 0. & 0.91980852 \end{bmatrix}$$
$$cov3 = \begin{bmatrix} 1.18129525 & 0. \\ 0. & 1.01867894 \end{bmatrix}$$

先验概率（每个高斯分布的占比）
$p1 = 0.33223786$
$p2 = 0.33059483$
$p3 = 0.33716731$

EM组



$$mean1 = (0.30955325, 0.0365708)$$
$$mean2 = (5.14431927, -0.09481982)$$
$$mean3 = (5.04307149, 5.15486069)$$

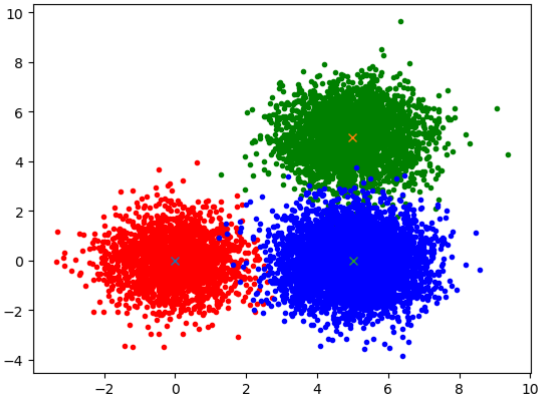
$$cov1 = \begin{bmatrix} 1.75063133 & 0. \\ 0. & 1.01096767 \end{bmatrix}$$
$$cov2 = \begin{bmatrix} 0.73253483 & 0. \\ 0. & 0.93930594 \end{bmatrix}$$
$$cov3 = \begin{bmatrix} 1.02495855 & 0. \\ 0. & 0.92037054 \end{bmatrix}$$

先验概率（每个高斯分布的占比）
$p1 = 0.35722213$
$p2 = 0.31216271$
$p3 = 0.33061517$

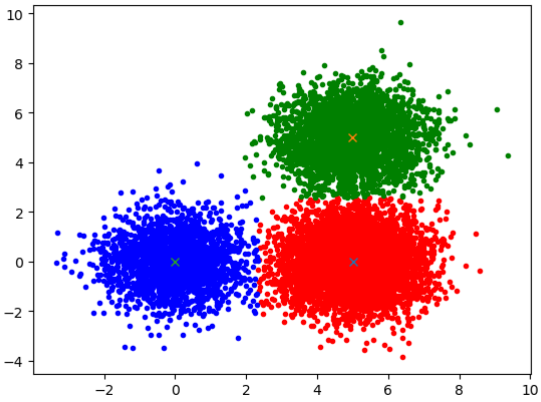
可以发现单纯的EM效果略微不如K-means+EM。

实验二 使用规模更大的数据

把实验一中的每个高斯分布的采样数分别增加到2000,3000和5000，其余保持不变



K-means + EM组

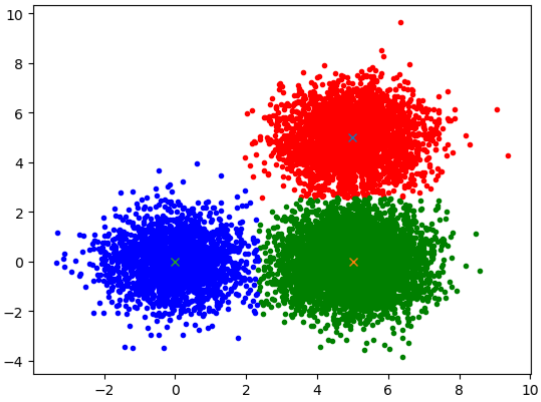


$$mean1 = (5.00970273, -0.00057948)$$
$$mean2 = (4.99734922, 4.97801927)$$
$$mean3 = (0.00296928, 0.00573215)$$

$$cov1 = \begin{bmatrix} 1.01952373 & 0. \\ 0. & 1.02811722 \end{bmatrix}$$
$$cov2 = \begin{bmatrix} 1.01952373 & 0. \\ 0. & 1.02811722 \end{bmatrix}$$
$$cov3 = \begin{bmatrix} 0.96948848 & 0. \\ 0. & 1.00573272 \end{bmatrix}$$

$$p1 = 0.49874010$$
$$p2 = 0.30017891$$
$$p3 = 0.20108098$$

EM组



$$mean1 = (4.99735808, 4.97790644)$$
$$mean2 = (5.01190117, -0.00055060)$$
$$mean3 = (0.00826209, 0.00548536)$$

$$cov1 = \begin{bmatrix} 1.01947804 & 0. \\ 0. & 1.02836869 \end{bmatrix}$$
$$cov2 = \begin{bmatrix} 0.96653607 & 0. \\ 0. & 1.00996156 \end{bmatrix}$$
$$cov3 = \begin{bmatrix} 0.98100376 & 0. \\ 0. & 1.00615513 \end{bmatrix}$$

$p1 = 0.30019226$
$p2 = 0.49829442$
$p3 = 0.20151330$

可以发现用EM和K-means+EM的差距缩小了。

还可以发现EM算法不能很好的处理有重合部分的数据，这一点也很好理解。因为最终通过一个点在某个高斯分布的概率乘上先验概率进行比较，来判断这个点使用哪一个高斯分布生成的，所以会有鲜明的决策分割线，因此不能很好的处理有重合部分的数据。

总结

这次实验加深了我对GMM模型，K-means算法和EM算法的认识。

通过实践，我明白了之前学习的时候看上去晦涩难懂的公式其实是很自然的得到的，能清晰的理解公式的含义。

了解了GMM的优点：容易实现的方便简单的聚类，也找到了GMM的缺点：不能很好的处理有重合的数据。