Assignment3 高斯混合分布模型

Part 1 生成高斯分布

从高斯混合模型中生成一个样本x可以分为两步：

1. 根据多项式分布随机选取一个高斯分布
2. 假设选中第k个高斯分布，再从该高斯分布中选取样本x

整体高斯分布的样本生成可以近似等价于对第k个高斯分布选取N*P(k)个样本进行混合，每一个高斯分布的多项分布和P(k)相关

所以本实验中可以通过自己定义K(定义高斯分布的个数)，mu，sigma(定义每一个高斯分布)，n(每一个高斯分布的采样数)，生成无标签的数据分布。

为了便于可视化，以下实验都使用二维高斯分布。

Part 2 使用EM模型进行高斯分布拟合

2.1 初始化参数的方法

在训练前，需要初始化一下参数

- k:cluster的个数
- pi: 每个cluster的先验概率
- mu:每个cluster的初始mu值
- sigma:每个cluster的初始sigma

2.1.1 如何确定超参数K

可以通过遍历不同的K值，使用EM算法，通过贝叶斯信息准备挑选最符合当前数据的K值。
本实验中直接定义为真实K值。

2.1.2 如何初始化超参数mu,sigma,pi

由于EM算法对初始化参数敏感，所以尝试使用两种初始化方法。

随机初始化

通过init_theta初始化pi,mu,sigma

- pi:初始化为1/K
- mu:随机挑选K个点作为K个cluster的mu
- sigma:初始化为np.eye(K)

使用k-means初始化

通过init_kmeans_theta初始化pi,mu,sigma
通过scipy.cluster.vq.kmeans2库，对数据进行k-means分类。

- pi:初始化为k-means分类后每个类的样本数占比

- mu:初始化为k-means分类后每个类的中心
- sigma:初始化为np.eye(K)

2.2 EM算法

2.2.1EM过程

1. E
   使用固定的mu(t),sigma(t),pi(t)，计算先验概率p(Z|X)

$$\begin{aligned}\gamma_{nk} &\triangleq p(z^{(n)}=k|x^{(n)}) \\ &= \frac{p(z^{(n)})p(x^{(n)}|z^{(n)})}{p(x^{(n)})} \\ &= \frac{\pi_k \mathcal{N}(x^{(n)}|\mu_k,\sigma_k)}{\sum_{k=1}^{K}\pi_k \mathcal{N}(x^{(n)}|\mu_k,\sigma_k)},\end{aligned}$$

2. M
   使用固定的pi(t),进行mu(t+1),sigma(t+1),pi(t)的更新

$$\begin{aligned}&ELBO(\gamma,\mathcal{D}|\pi,\mu,\sigma) = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\log\frac{p(x^{(n)},z^{(n)}=k)}{\gamma_{nk}} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\left(\log\mathcal{N}(x^{(n)}|\mu_k,\sigma_k)+\log\frac{\pi_k}{\gamma_{nk}}\right) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\left(\frac{-(x-\mu_k)^2}{2\sigma_k^2}-\log\sigma_k+\log\pi_k\right)+C,\end{aligned}$$

   最大化ELBO，可得：

$$N_k = \sum_{n=1}^{N}\gamma_{nk}.$$

$$\begin{aligned}\pi_k &= \frac{N_k}{N}, \\ \mu_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}x^{(n)}, \\ \sigma_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}(x^{(n)}-\mu_k)^2\end{aligned}$$

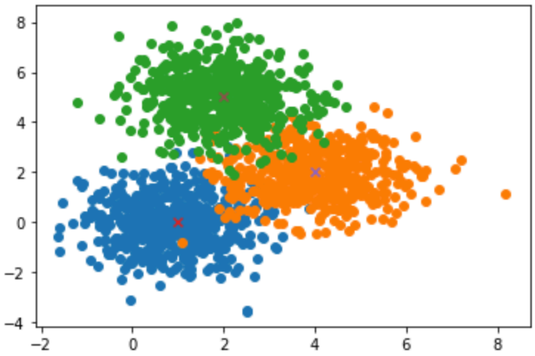
2.2.2EM算法终止条件

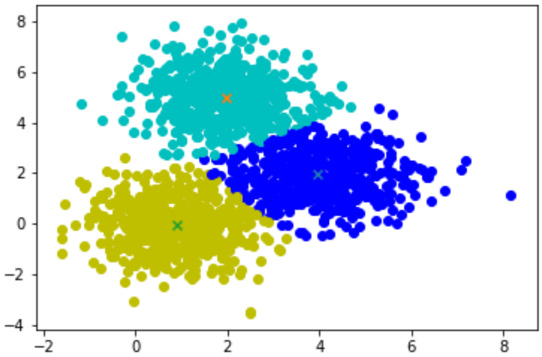
1. 迭代次数超过超参数m次
2. ELBO的值两次迭代之间的差小于1e-3

3 实验结果

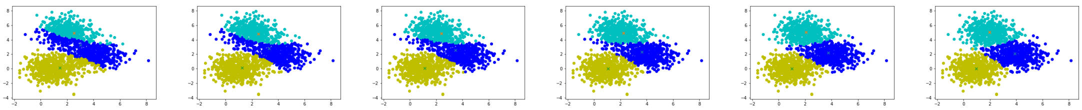
3.1 分布较分散数据实验结果

高斯分布参数:
K=3,mean = [[1,0],[4,2],[2,5]],cov = [[[1,0],[0,1]],[[1,0],[0,1]],[[1,0],[0,1]]],N=[500,500,500]

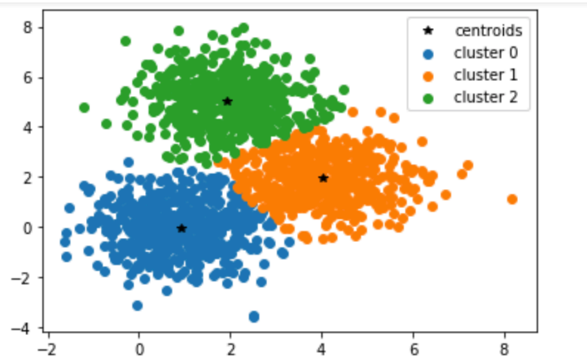


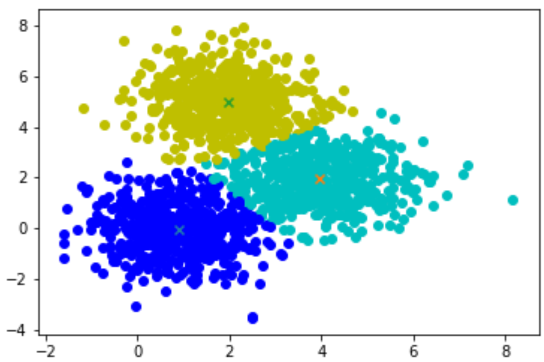


左图为样本真实分类，右图为随机初始化后进行迭代至收敛的EM算法结果， 迭代次数=44



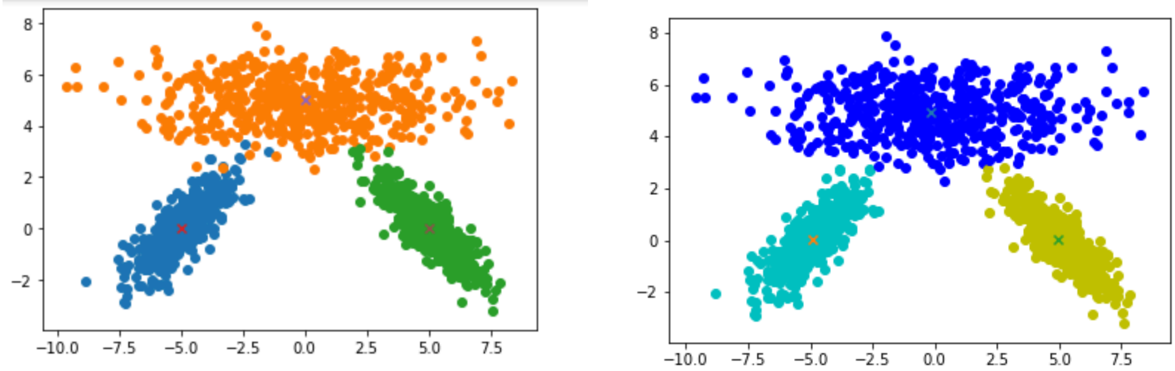
上图分别为迭代次数为0,2,4,6,8,10时的分类结果，可以看出EM算法在迭代的过程中学习到真实数据的高斯分布。



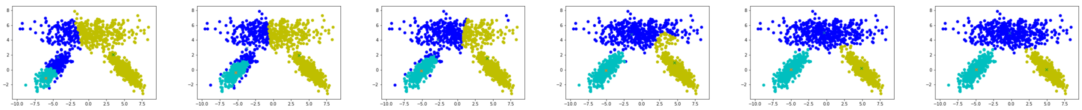


左图为使用k-means初始化时的预测结果，右图为EM算法收敛时的结果，迭代次数为40次
可以看出在不同类别的样本重叠比较小且样本属于的高斯分布cov为对角阵时EM算法和k-means算法都能较好地多原分布进行模拟。

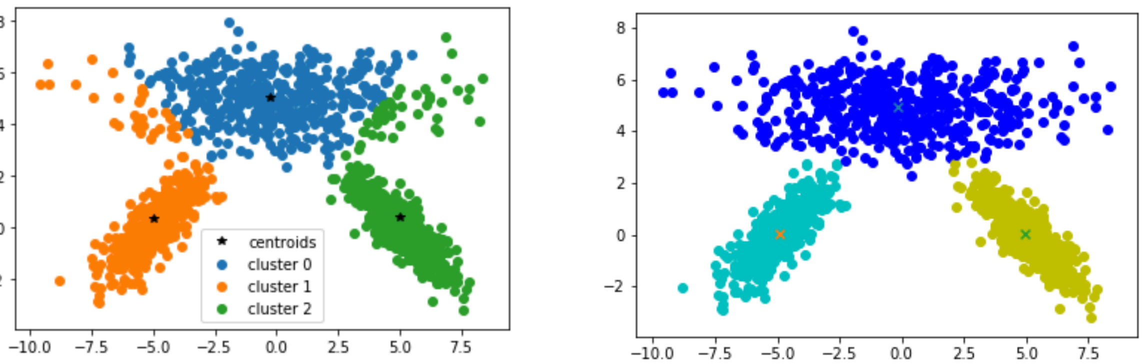
3.2 分布方差值不对称



左图为样本真实分类，右图为随机初始化后进行迭代至收敛的EM算法结果



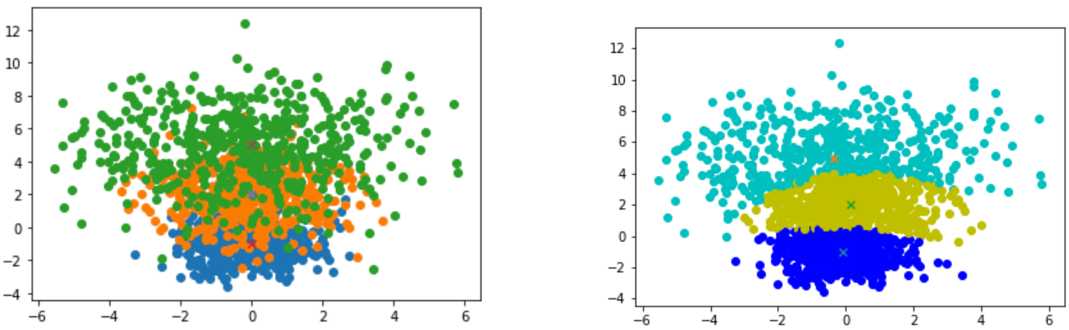
上图分别为迭代次数为0,2,4,6,8,10时的分类结果，可以看出EM算法在迭代的过程中的学习考虑到节点属于某个高斯分布的概率，分类的边缘有一定弧度，和kmeans之间按照距离的迭代不同。



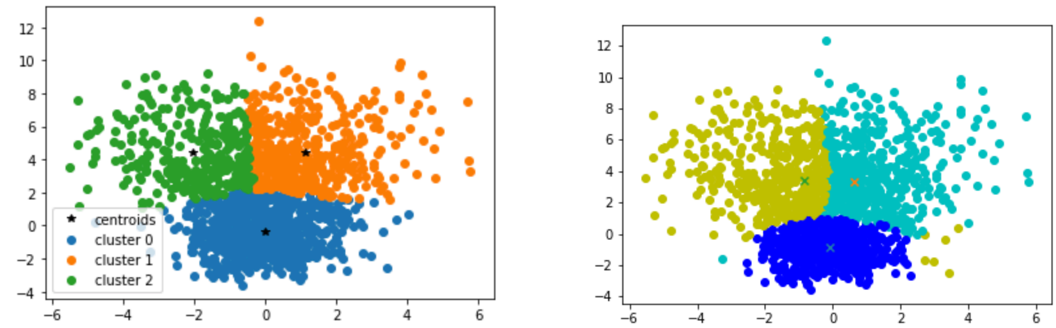
左图为使用k-means初始化时的预测结果，右图为EM算法收敛时的结果。

可以看出由于k-means只考虑节点到中心节点的距离，而没有考虑高斯分布本身的性质，对于此类方差值不对称的数据很容易造成错误的分类，而EM算法则可以较好地进行预测。

3.3 分布重叠性较高



左图为样本真实分类，右图为随机初始化后进行迭代至收敛的EM算法结果

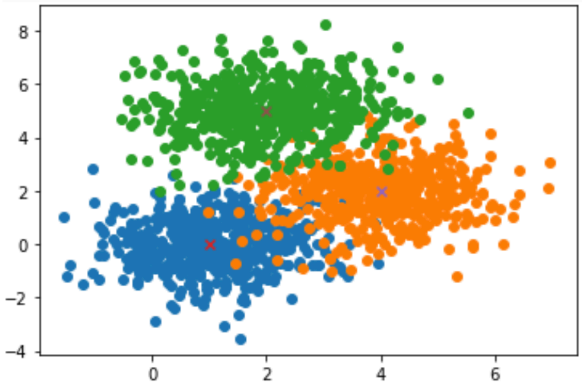


左图为使用k-means初始化时的预测结果，右图为EM算法收敛时的结果。

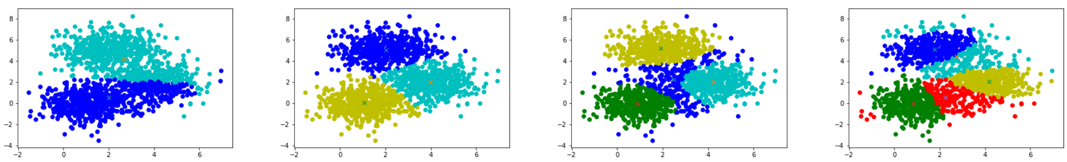
可以看出k-means和EM算法都没有很好得模拟真实分布，而且由于EM算法的初始化敏感，两种初始化办法得到的分类结果差距很大。可以看出对与重叠度较大的GMM模型，两种分类算法都给出了合理但不是真实的结果。

3.4 K值不固定的情况下

观察初始化参数 K=2/3/4/5时对同样样本的分类



上图为样本真实分类



上图为给定不同K值时，使用K-means初始化后进行迭代至收敛或迭代至50步的EM算法结果。
可以看出在K=2/3时得到的分类都比较自然，且在K=3时的迭代次数所需最少，在K=5时，迭代50次后仍未收敛。

4 总结

EM算法在对GMM模型进行聚类上有着很良好的表现，尤其是在对高斯分布的方差不对称的情况下，EM算法比K-means算法的结果好很多。但是由于EM算法的对初始化敏感，可以使用K-means算法进行EM算法的初始化，或者尝试多次随机初始化后，选取ELBO值最高的一组作为结果。

5 程序运行

可以使用
python3 source.py 运行程序