

assignment-3 实验报告

实验目的

生成若干个二维的高斯分布数据

使用GMM模型对数据点属于那个高斯模型进行分类

GMM模型

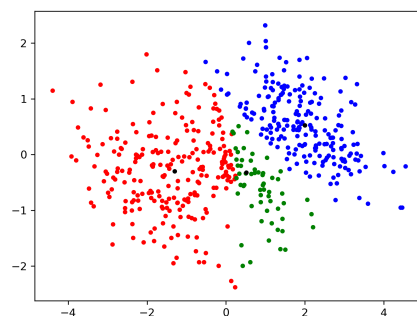
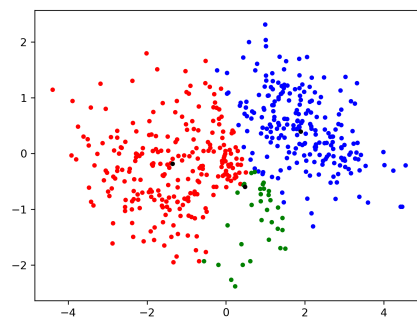
模型介绍

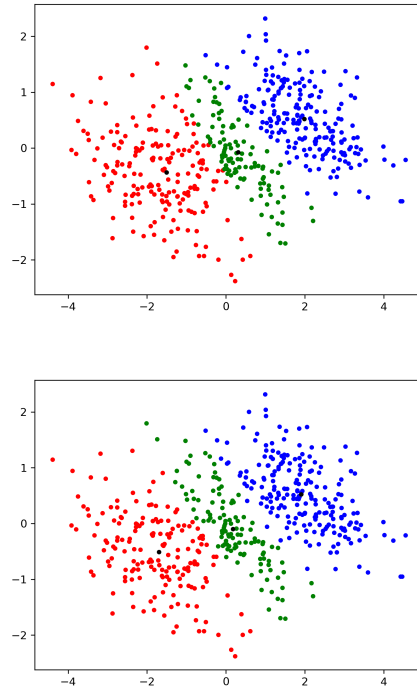
使用高斯混合模型，对于预测的每个高斯分布，需要学习三个参数：平均值 μ ，协方差矩阵 σ 和分布的先验概率 π 。

对于训练数据集，根据EM算法，按以下流程进行训练：

1. 初始化参数平均值 μ ，协方差矩阵 σ 和分布的先验概率 π 。这里令 $\sigma = I$ ， $\pi = 1/k$ ， k 为每个高斯分布的个数
2. E步，根据现有的 μ ， σ 和 π 进行计算每个点的后验概率 $\gamma_{n,k}$
3. M步，根据现有的后验概率 $\gamma_{n,k}$ 计算使证据下界最大化时的参数 μ ， σ 和 π
4. 重复E步和M步，迭代若干次

基于EM算法，模型可以根据当前值向极小值收敛，最终达到局部最优。

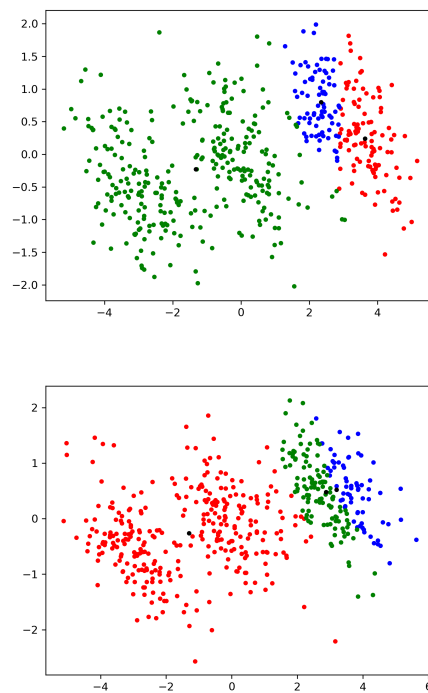


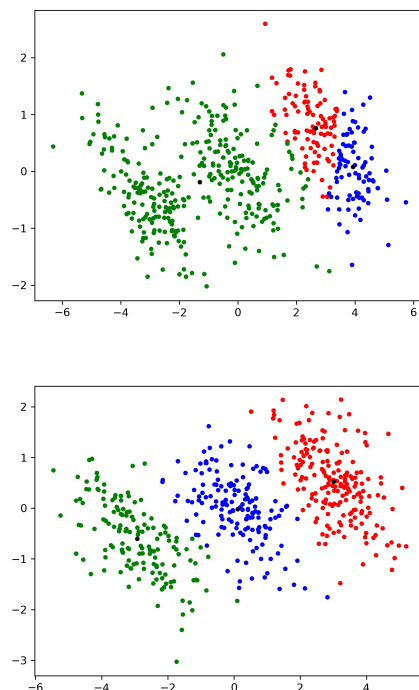


如图所示，随着迭代次数增加，每个分布的均值 μ （黑色标注）会向着极小值点收敛，最终使得证据下界达到极小值。

初值选取

在训练过程中，发现对于同样的分布数据，如果初值选取适当，较为分散，会有较好的训练结果；如果初值选在同一个聚类中，而且每个分布的均值之间又相隔较远，会出现很大的误差。





上图为执行100次迭代之后的分类结果，可以看出当三个均值的预测点都落在不同的聚类时，可以获得较好的效果；否则，如果某个聚类包含了多个初始均值点，很容易被预测成多个高斯分布，误差很大。

通过计算可知，在 k 个聚类中随机选取 k 个初始均值点，若每个聚类点数相同，只会有 $\frac{k!}{k^k}$ 的概率使得选取的这些均值点对应不同的分布。当 k 变大时，这个概率会迅速变小。

kmeans++

这里，使用kmeans++的方法进行初始化。先用k-means选出可能的均值点，在用GMM模型进行训练。

k-means也是运用EM算法，会先找出 m 个分布较分散的点作为初始的中心点，E步根据每个点到各中心点的距离对每个点进行分类，在M步根据分类好的结果更新每个分布的中心点。

这里，令初始化时先用k-means迭代10次，把迭代后的中心点当作GMM中初始的均值点，再按照原先的步骤进行训练。

令使用随机法和kmeans+的模型分别迭代50次，每10次迭代后输出准确率，取5次训练的平均值。

| | 随机 | KMEANS++ |
|----------|--------|----------|
| epoch=0 | 0.6284 | 0.9795 |
| epoch=10 | 0.7184 | 0.988 |
| epoch=20 | 0.7472 | 0.988 |
| epoch=30 | 0.7804 | 0.988 |
| epoch=40 | 0.8136 | 0.988 |
| epoch=50 | 0.8240 | 0.988 |
| epoch=60 | 0.8240 | 0.988 |

可见，使用kmeans++方法得到初始值，可以避免在同一个聚类中在初始化时被分到多个高斯分布中的情况，提高了分类效果。

为了测试k-means对于呈高斯分布的聚类的预测情况，令使用kmeans+的GMM模型和k-means模型分别迭代50次，每10次迭代后输出准确率，取5次训练的平均值。

| | KMEANS++ | K-MEANS |
|----------|----------|---------|
| epoch=0 | 0.9795 | 0.9068 |
| epoch=10 | 0.988 | 0.9556 |
| epoch=20 | 0.988 | 0.9556 |
| epoch=30 | 0.988 | 0.9556 |
| epoch=40 | 0.988 | 0.9556 |
| epoch=50 | 0.988 | 0.9556 |
| epoch=60 | 0.988 | 0.9556 |

对于高斯分布的聚类，k-means的方法只能按照距离进行分类，对于某些分布较为分散的聚类并不能起到很好的分类作用。

代码运行说明

```
1 python source.py
```

默认情况下，会创建中心在 $\{(0, 0), (-3, 0)(4, 0)\}$ ，协方差为 $\{[[1, 0], [0, 1]], [[1.5, 0], [0, 1.5]], [[0.8125, -0.325], [-0.325, 0.4375]]\}$ 的高斯分布数据点共500个，按照 $epoch_{gmm} = 100, epoch_{kmeans} = 10$ 进行训练。