Report

问题描述

part1：构建一个没有任何标签的数据集用于聚类。

part2：设计高斯混合模型，对构建的数据集进行标记，完成聚类任务。

代码说明

1、代码运行方式：

在命令行中定位到相应文件夹下，输入命令 python source.py，即可运行程序。程序开始运行后，会展示未分类前数据的图像。参数学习开始后，每隔 10 个 step，命令行中会显示当前 Q 函数的值。学习完成后，程序会展示最终的聚类结果图像。

2、其它说明：

生成数据集的函数也包含在了 source.py 中，由于已经提供了小数据集 gauss.data，生成数据部分的函数不会在运行程序时被调用。

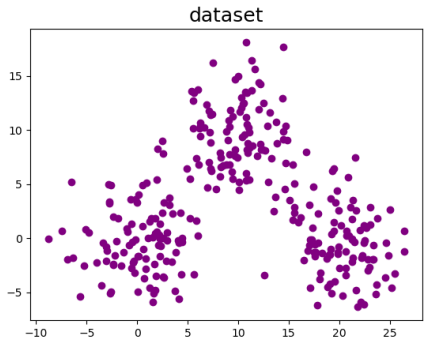
Part 1

为了便于作图观察，生成数据集时选择生成二维平面中的点。

使用三个二维高斯分布来生成数据。三个高斯分布的均值分别为 (0,0), (10,10), (20,0)，协方差矩阵均为$\begin{bmatrix}10 & 0\\ 0 & 10\end{bmatrix}$，样本数量均为 100。在抽样完成后，对所有的样本进行随机排列，使不同高斯分布生成的

样本均匀分布在数据集中。

在生成得到的数据集中，只包含了这 300 个样本点的横坐标和纵坐标，不包含其它任何信息。提交的 gauss.data 文件中的数据如下图所示：



可以看到，图中的点大致分成 3 个部分，且各个部分之间没有明显的界限，适合作为聚类所用的数据集。

Part 2

本次设计的高斯混合模型(GMM)使用二维高斯分布作为参数模型，使用 EM 算法对模型进行训练。

1、初始化

训练之前，需要设定子模型的数量 K，即聚类后需要得到的类别数量。K 个子模型均为二维高斯分布。

对每个子模型，需要初始化的参数有 3 个：这个子模型的先验概率π，二维高斯分布的均值μ，二维高斯分布的协方差矩阵Σ。由于 EM 算法对模型的初值敏感，所以初始化时的参数也要尽可能合理。先验概率π在 0.1~1 之间随机；均值μ在一个矩形范围内随机，矩形的左下角为(0,0)，右上角为(20,20)，这样的范围大小可以让初始时的模型更

好，有助于模型更快达到收敛状态；协方差矩阵Σ为$\begin{bmatrix} \mathrm{k} & 0 \\ 0 & \mathrm{k} \end{bmatrix}$，其中 1≤k≤21，这样能够保证后续求逆矩阵或求行列式的值时，协方差矩阵合法。

2、E 步

E 步计算每个样本属于不同高斯分布的概率。

均值为μ_k，协方差矩阵为Σ_k的二维高斯分布，对应的似然函数为：

$$N(x|\mu_k,\Sigma_k)=\frac{1}{2\pi*|\Sigma_{\mathrm{k}}|^{\frac{1}{2}}}\exp\{-\frac{1}{2}(\mathrm{x}-\mu_{\mathrm{k}})^{\mathrm{T}}\Sigma_{\mathrm{k}}{}^{-1}(\mathrm{x}-\mu_{\mathrm{k}})\}$$

定义γ_{nk}为样本$x^{(n)}$属于第 k 个二维高斯分布的后验概率

$$\gamma_{nk}=\frac{p(z^{(n)})p(x^{(n)}|z^{(n)})}{p(x^{(n)})}=\frac{\pi_k N(x^{(n)}|\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\pi_k N(x^{(n)}|\mu_k,\Sigma_k)}$$

3、M 步

M 步根据 E 步得到的结果，对模型的参数进行更新。

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

$$\pi_k=\frac{N_k}{N}$$

$$\mu_k=\frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}\,x^{(n)}$$

$$\Sigma_{\mathrm{k}}=\frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}(x^{(n)}-\mu_{\mathrm{k}})\left(x^{(n)}-\mu_{\mathrm{k}}\right)^{\mathrm{T}}$$

4、Q 函数

完全数据的对数似然函数$\log P(Y,Z|\theta)$关于在给定观测数据 Y 和当前参数$\theta^{(i)}$下对未观测数据 Z 的条件概率分布$P(Z|Y,\theta^{(i)})$的期望称为 Q 函数，即

$$Q(\theta,\theta^{(i)}) = E_z[\log P(Y,Z|\theta)|Y,\theta^{(i)}]$$

在这个模型下，计算得到 Q 函数为

$$Q(\theta,\theta^{(i)}) = \sum_{k=1}^{K}\{n_k log\pi_k + \sum_{n=1}^{N}\gamma_{nk}[-log\sqrt{2\pi|\Sigma_k|} - \frac{1}{2|\Sigma_k|}(x^{(n)}-\mu_k)^T(x^{(n)}-\mu_k)]\}$$

EM 算法是通过求解完全数据的对数似然函数$\log P(Y,Z|\theta)$的期望的极大似然估计，也就是使 Q 函数$Q(\theta,\theta^{(i)})$达到极大值，来得到问题的结果的。

5、参数学习

参数学习的具体步骤如下：

① 通过随机的方法得到模型的初始值

② E 步：计算每个样本属于不同高斯分布的概率γ_{nk}

③ M 步：对模型的参数π_k,μ_k,Σ_k进行更新。

④ 重复②③步，直至满足终止条件时停止迭代

⑤ 根据γ_{nk}，将每个样本标注为概率最大的类，得到结果。

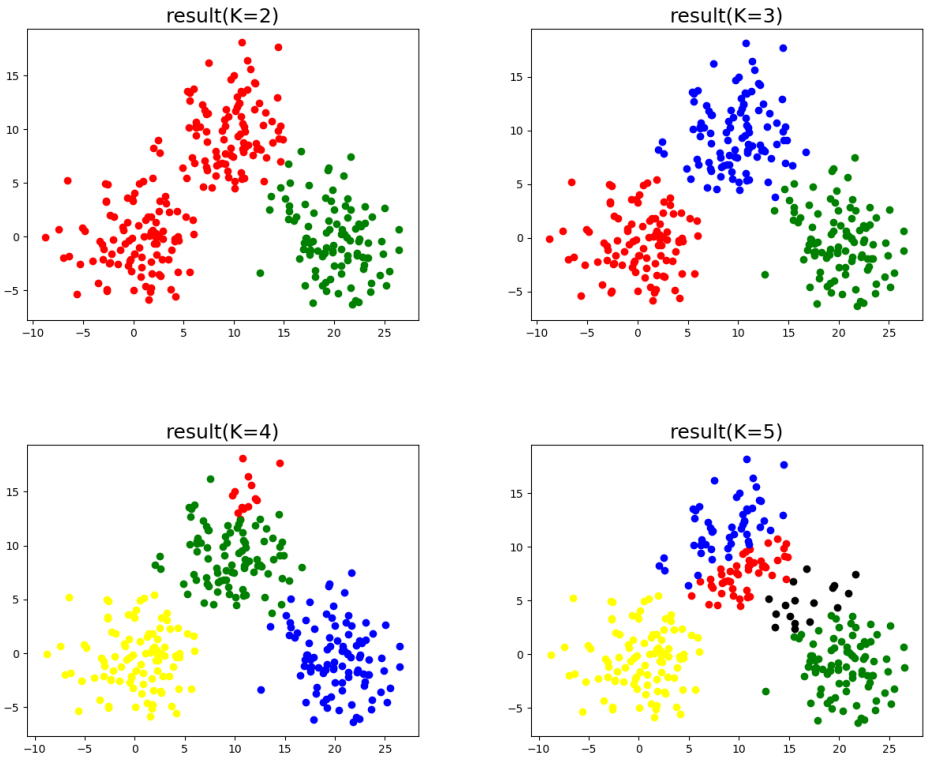
在实际的代码实现中，学习的终止条件有两种：一是迭代超过 200 个 step；二是模型达到收敛状态，即

$$\left||Q(\theta^{(i+1)},\theta^{(i)}) - Q(\theta^{(i)},\theta^{(i)})\right|| < \varepsilon \quad \varepsilon = 10^{-4}$$

6、结果

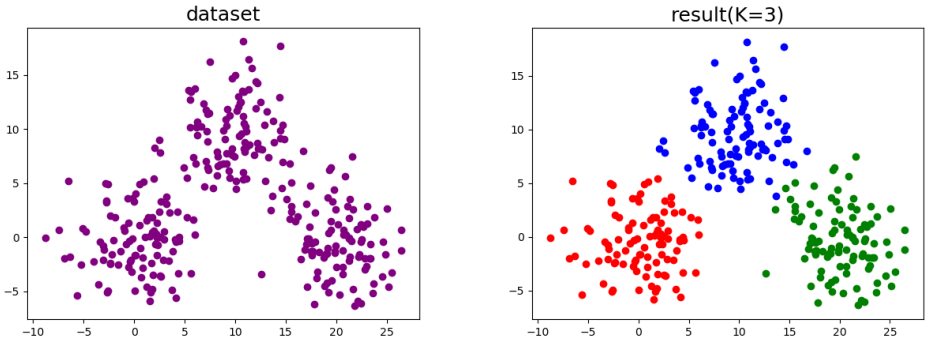
GMM 是一种无监督的方法，对结果的评估只能依靠肉眼的观察和比较。EM 算法虽然能保证收敛，但是并不能保证找到全局最大值，而是有可能找到局部最大值。所以，通过多次运行程序的方式，取其中 Q 函数值最大的结果，作为最终结果。

分别取 K=2,3,4,5 进行测试，对每一个 K 的取值运行 5 次程序，得到的结果如下图：



观察上面不同的 K 值对应的结果，可以发现，K=2 或 3 都能得到不错的结果。K=4 或 5 时，得到的结果与 K=3 的结果相似，只是进行了更细的划分。在测试过程中还发现，只有 K=3 得到的结果比较稳定，K 取其它值时结果的区别都比较大，这也说明 K=3 是最适合该数据集的超参数。

最终的聚类结果如下图所示。肉眼观察可以发现，得到的结果是非常合理的。



总结

高斯混合模型(GMM)是一种使用多个高斯分布来刻画数据分布的模型，它通常使用 EM 算法来进行参数学习。

由于 EM 对初值敏感的特点，在初始化数据时，需要进行一些合理的设置，例如在生成初始高斯分布的均值时，扩大随机的范围。

EM 算法具备收敛性，但是并不一定能找到全局最大值，有可能会找到局部最大值。一种可行的解决方案是使用不同的初始化参数进行迭代，取其中结果最好的。