第二次作业报告

学号：21375255 姓名：王鑫超

一、GMM模型

高斯混合模型（Gaussian Mixture Model，GMM）是一种基于概率密度函数的聚类方法，它假设每个聚类都是由多个高斯分布组成的混合分布。GMM的目标是通过最大化似然函数来估计模型参数，包括每个高斯分布的均值、方差和混合系数，以及数据点属于每个聚类的概率。在聚类时，GMM将数据点分配到概率最大的聚类中，而不是像K-Means那样将数据点硬性分配到某个聚类中。GMM在许多应用中都表现出色，尤其是当数据点不是明显分离的时候。

GMM模型的优势在于：

- GMM可以处理复杂的数据分布，因为它可以用多个高斯分布来近似描述数据分布；
- GMM可以自适应地调整簇的数量和大小，从而更好地适应不同的数据分布；
- GMM可以用于生成新的数据样本

GMM模型的概率密度函数如下：

$p(x) = \sum_{i=1}^{n} \pi_i N(x|\mu_i, \sigma_i)$

其中$\sum_{i=1}^{n} \pi_i = 1$

那么，要建立GMM模型，重点就在于如何估计模型中的π、μ、σ参数。

二、EM算法

针对GMM模型的参数估计问题，可以引入EM算法进行解决。

最大期望算法（Expectation-maximization algorithm）在统计中被用于寻找，依赖于不可观察的隐性变量的概率模型中，参数的最大似然估计。

EM算法的通用步骤如下：
1、初始化模型参数
2、重复进行以下步骤直到收敛：

- E-step：根据参数的假设值，给出未知变量的期望估计，应用于缺失值。
- M步骤：根据未知变量的估计值，给出当前的参数的极大似然估计。

针对GMM模型，使用EM算法推导其模型参数的推导过程如下：

EM算法推导

设现有的数据集为 $X=\{x_i \mid i=1,2,\cdots,n\}$，表示现有的大学生身高数据.

对GMM模型:

$\theta=(\pi_1,\pi_2,\cdots,\pi_n,\mu_1,\mu_2,\cdots,\mu_n,\sigma_1^2,\sigma_2^2,\cdots,\sigma_n^2)$

$p(x|\theta)=\sum_{i=1}^{m}\pi_i N(x|\mu_i,\sigma_i)$ 为GMM模型的概率密度函数,

对数据集X，估计θ参数，利用最大似然函数估计的思路，最大似然函数

$$L(X,\theta)=\sum_{i=1}^{n}\ln p(x_i|\theta)=\sum_{i=1}^{n}\ln\sum_{j=1}^{m}\pi_j N(x_i|\mu_j,\sigma_j)$$

引入隐变量z，$P(z=k)=\pi_k$，设 $P(x_i|z=k,\mu_k,\sigma_k)=N(x_i|\mu_k,\sigma_k)$

则 $L(X,\theta)=\sum_{i=1}^{n}\ln\sum_{j=1}^{m}P(z=j)\cdot P(x_i|z=j,\mu_j,\sigma_j)$

$$=\sum_{i=1}^{n}\ln\sum_{j=1}^{m}p(z=j|x_i,\mu_j,\sigma_j)\cdot\frac{p(z=j)\cdot p(x_i|z=j,\mu_j,\sigma_j)}{p(z=j|x_i,\mu_j,\sigma_j)}$$

由Jensen不等式,

$$L(X,\theta)\geq\sum_{i=1}^{n}\sum_{j=1}^{m}p(z=j|x_i,\mu_j,\sigma_j)\cdot\ln\frac{p(z=j)\cdot p(x_i|z=j,\mu_j,\sigma_j)}{p(z=j|x_i,\mu_j,\sigma_j)}$$

$$p(z=j|x_i,\mu_j,\sigma_j)=\frac{p(x_i|z=j,\mu_j,\sigma_j)}{\sum_{j=1}^{m}p(x_i|z=j,\mu_j,\sigma_j)}=\frac{\pi_j N(x_i|\mu_j,\sigma_j)}{\sum_{j=1}^{m}\pi_j N(x_i|\mu_j,\sigma_j)}$$

令 $w_{ij}=p(z=j|x_i,\mu_j,\sigma_j)$，则 $L(X,\theta)\geq\sum_{i=1}^{n}\sum_{j=1}^{m}w_{ij}\ln\frac{\pi_j N(x_i|\mu_j,\sigma_j)}{w_{ij}}$

令 $Q(\theta,\theta^{(k)})=\sum_{i=1}^{n}\sum_{j=1}^{m}w_{ij}^{(k)}\ln\frac{\pi_j N(x_i|\mu_j,\sigma_j)}{w_{ij}^{(k)}}$，$\theta^{(k)}$ 表示第k次迭代的参数

由于 $Q(\theta,\theta^{(k)})\leq L(X,\theta)$，通过设定 $\theta^{(0)}$，每次取 $\theta=\theta_k^{\cancel{(k+1)}}$，使得

$\theta=\arg\max Q(\theta,\theta^{(k)})$ 再令 $\theta^{(k+1)}=\theta_k$ 多次迭代后，便可得到θ的MLE

$$w_{ij}^{(k)} = \frac{\pi_j^{(k)} N(x_i \mid \mu_j^{(k)}, \sigma_j^{(k)})}{\sum_{j=1}^{m} \pi_j^{(k)} \cdot N(x_i \mid \mu_j^{(k)}, \sigma_j^{(k)})}$$

则 $Q(\theta, \theta^{(k)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^{(k)} \cdot \left[\ln \pi_j - \ln w_{ij}^{(k)} - \frac{1}{2} \ln(2\pi\sigma_j^2) - \frac{(x_i - \mu_j)^2}{2\sigma_j^2} \right]$

通过优化 π_j, σ_j^2, μ_j，使上式取得最大值即可

$\frac{\partial Q}{\partial \pi_j} = \sum_{i=1}^{n} \cancel{\sum_{j=1}^{m}} w_{ij}^{(k)} \cdot \frac{1}{\pi_j}$ ，而 $\sum_{j=1}^{m} \pi_j = 1$，

利用Lagrage乘子法， $L(\pi_j, \lambda) = Q(\theta, \theta^{(k)}) + \lambda\left(\sum_{j=1}^{m} \pi_j - 1\right)$

$\frac{\partial L}{\partial \pi_j} = \frac{\sum_{i=1}^{n} w_{ij}^{(k)}}{\pi_j} + \lambda = 0$ ， $\therefore \cancel{\lambda}$

$\therefore \pi_j = -\frac{\sum_{i=1}^{n} w_{ij}^{(k)}}{\lambda}$ ， $\therefore \sum_{j=1}^{m} \pi_j = -\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(k)}}{\lambda} = 1$

$\therefore \lambda = -n$

$\therefore \pi_j = \frac{\sum_{i=1}^{n} w_{ij}^{(k)}}{n}$ 时，$Q(\theta, \theta^{(k)})$ 取最大，

可令 $\pi_j^{(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k)}}{n}$ ，

$\frac{\partial Q}{\partial \mu_j} = \sum_{i=1}^{n} \cancel{\sum_{j=1}^{m}} w_{ij}^{(k)} \cdot 2 \cdot (x_i - \mu_j) = 0$，

$\therefore$ 可取 $\mu_j^{(k+1)} = \sum_{i=1}^{n} x_i \cdot w_{ij}^{(k)} \Big/ \sum_{i=1}^{n} w_{ij}^{(k)}$

$\frac{\partial Q}{\partial \sigma_j^2} = \sum_{i=1}^{n} \cancel{\sum_{j=1}^{m}} w_{ij}^{(k)} \cdot \left[-\frac{\pi}{2\pi\sigma_j^2} + \frac{(x_i - \mu_j)^2}{2(\sigma_j^2)^2} \right] = 0$

$$\therefore (\sigma_j^2)^{(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k)} \cdot (x_i - \mu_j^{(k+1)})^2}{\sum_{i=1}^{n} w_{ij}^{(k)}}$$

按以上方法迭代，即可求出 θ^*

三、核心代码

依照上述的推导过程，设计GMM的参数估计函数，核心的代码如下：

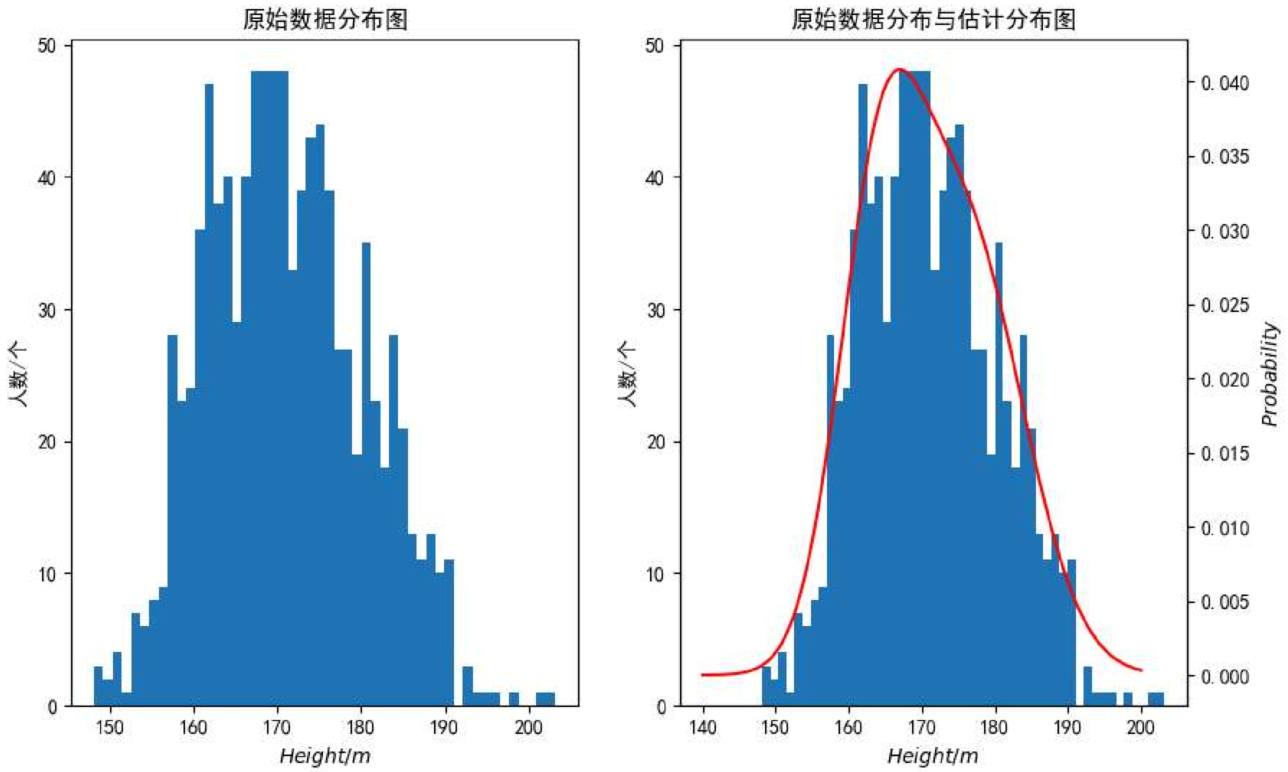
```
def EM_GMM(x, n_samples=1000, k=2):
    pi = np.ones((k, )) / k
    miu = np.random.uniform(150, 180, (k, ))
    sigma = np.random.uniform(5, 10, (k, ))
    omiga = np.zeros((n_samples, k))
    for epochs in range(10):
        #循环算omiga
        for i in range(n_samples):
            sum = 0
            for j in range(k):
                omiga[i][j] = pi[j] * norm.pdf(x[i], miu[j], sigma[j])
                sum = sum + omiga[i][j]
            for j in range(k):
                omiga[i][j] = omiga[i][j] / sum
        #更新参数
        sum_omiga = np.zeros((k, ))
        sum_x_omiga = np.zeros((k, ))
        for j in range(k):
            for i in range(n_samples):
                sum_omiga[j] = sum_omiga[j] + omiga[i][j]
                sum_x_omiga[j] = sum_x_omiga[j] + omiga[i][j] * x[i]
        for j in range(k):
            pi[j] = sum_omiga[j] / n_samples
            miu[j] = sum_x_omiga[j] / sum_omiga[j]
        for j in range(k):
            sum = 0
            for i in range(n_samples):
                sum = sum + omiga[i][j] * (x[i] - miu[j]) ** 2
            sigma[j] = np.sqrt(sum / sum_omiga[j])
    return pi, miu, sigma
```

随后对模型进行检验，按照要求生成对应的训练数据：

```
np.random.seed(1)
male = np.random.normal(176, 8,(600, ))
female = np.random.normal(164, 6,(400, ))
height = np.hstack((male, female))
```

四、结果展示

拟合的效果如下：



最后拟合得到的参数如下：

$\pi_1 = 0.39973408, \pi_2 = 0.60026592$

$\mu_1 = 164.04154896, \mu_2 = 175.7892246$

$\sigma_1 = 5.71203936, \sigma_2 = 8.06707288$

原始的参数如下：

$\pi_1 = 0.4, \pi_2 = 0.6$

$\mu_1 = 164, \mu_2 = 176$

$\sigma_1 = 6, \sigma_2 = 8$

可以看到，拟合的参数与准确值非常接近，说明EM算法估计GMM模型参数的方法非常奏效。