《大数据机器学习》第 7 次作业

姓名：刘培源　　学号：2023214278

题目 1： 如例 9.1 的三硬币模型。假设观测数据不变，试选择不同的初值，例如，$\pi(\theta) = 0.46$，$p(\theta) = 0.55$，$q(\theta) = 0.67$，求模型参数 $\theta = (\pi, p, q)$ 的极大似然估计。

例 9.1（**三硬币模型**）　假设有 3 枚硬币，分别记作 A，B，C。这些硬币正面出现的概率分别是 π，p 和 q。进行如下掷硬币试验：先掷硬币 A，根据其结果选出硬币 B 或硬币 C，正面选硬币 B，反面选硬币 C；然后掷选出的硬币，掷硬币的结果，出现正面记作 1，出现反面记作 0；独立地重复 n 次试验（这里，$n = 10$），观测结果如下：

$$1, 1, 0, 1, 0, 0, 1, 0, 1, 1$$

假设只能观测到掷硬币的结果，不能观测掷硬币的过程。问如何估计三硬币正面出现的概率，即三硬币模型的参数。

答： 我的代码实现如下：

```python
class ThreeCoinEM:
    def __init__(self, prob, tol=1e-5, max_iter=100):
        # 初始化参数：prob_pi，prob_p，prob_q 分别是三个硬币的概率；tol 是收敛阈值；max_iter 是最大迭代次数
        self.prob_pi, self.prob_p, self.prob_q = prob
        self.tol = tol
        self.max_iter = max_iter

    def calc_mu(self, x):
        # 计算隐变量的期望，即E步骤
        pro_1 = self.prob_pi * self.prob_p ** x * (1 - self.prob_p) ** (1 - x)
        pro_2 = (1 - self.prob_pi) * self.prob_q ** x * (1 - self.prob_q) ** (1 - x)
        return pro_1 / (pro_1 + pro_2)

    def fit(self, data):
        # 对数据进行拟合，实现EM算法的主要逻辑
        count = len(data)
        print("Initial prob:")
        print(f"prob_pi={self.prob_pi}, prob_p={self.prob_p}, prob_q={self.prob_q}")
        print("Begin EM:")
        for i in range(self.max_iter):
            mu = [self.calc_mu(data[j]) for j in range(count)]

            # 更新概率值，即M步骤
            prob_pi = 1 / count * sum(mu)
            prob_p = sum(m * d for m, d in zip(mu, data)) / sum(mu)
            prob_q = sum((1 - m) * d for m, d in zip(mu, data)) / sum(1 - m for m in mu)
            print(f"Iteration {i + 1}: prob_pi={prob_pi:.4f}, prob_p={prob_p:.4f}, prob_q={prob_q:.4f}")

            # 判断是否达到收敛条件
            error = abs(self.prob_pi - prob_pi) + abs(self.prob_p - prob_p) + abs(self.prob_q - prob_q)
            self.prob_pi, self.prob_p, self.prob_q = prob_pi, prob_p, prob_q

            if error < self.tol:
                print("Final prob:")
                print(f"prob_pi={self.prob_pi:.4f}, prob_p={self.prob_p:.4f}, prob_q={self.prob_q:.4f}")
                break

data = [1, 1, 0, 1, 0, 0, 1, 0, 1, 1]
init_prob = [0.46, 0.55, 0.67]

em = ThreeCoinEM(prob=init_prob)
em.fit(data)
```

代码的输出如下：

```
Initial prob:
prob_pi=0.46, prob_p=0.55, prob_q=0.67
Begin EM:
Iteration 1: prob_pi=0.4619, prob_p=0.5346, prob_q=0.6561
Iteration 2: prob_pi=0.4619, prob_p=0.5346, prob_q=0.6561
Final prob:
prob_pi=0.4619, prob_p=0.5346, prob_q=0.6561
```

所以，三硬币正面出现的概率分别为 $\pi = 0.4619$，$p = 0.5346$，$q = 0.6561$

题目 2： 已知观测数据 -67，-48，6，8，14，16，23，24，28，29，41，49，56，60，75，试估计两个分量的高斯混合模型的 5 个参数。

答： 我们采用 "sklearn.mixture" 来实现这个功能，代码如下：

```python
from sklearn.mixture import GaussianMixture
import numpy as np
import matplotlib.pyplot as plt

# 初始化观测数据
data = np.array([-67, -48, 6, 8, 14, 16, 23, 24,
                 28, 29, 41, 49, 56, 60, 75]).reshape(-1, 1)

# 聚类
gmmModel = GaussianMixture(n_components=2)
gmmModel.fit(data)

# 预测
labels = gmmModel.predict(data)

# 输出结果
print(f"labels={labels}\n \
        means={gmmModel.means_.reshape(1, -1)}\n \
        covariances={gmmModel.covariances_.reshape(1, -1)}\n \
        weights={gmmModel.weights_.reshape(1, -1)}"
    )

# 可视化结果
plt.scatter(np.arange(len(data)), data, c=labels, s=50)
plt.title('Gaussian Mixture Model')
plt.show()
```

代码的输入如下：

```
labels=[1 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
means=[[ 32.98489643 -57.51107027]]
covariances=[[429.45764867  90.24987882]]
weights=[[0.86682762 0.13317238]]
```

所以两个分量的高斯混合模型的参数如下：

- $\mu_1 \approx 32.98, \mu_2 \approx -57.51$
- $\sigma_1 \approx 429.46, \sigma_2 \approx 90.25$
- $\alpha_1 \approx 0.867, \alpha_2 = 1 - \alpha_1 \approx 0.134$