机器学习报告 # 2

EM 算法*

董晟渤, 统计 91, 2193510853

西安交通大学数学与统计学院

日期：2022 年 4 月

目录

*2021-2022 学年第二学期, 课程: 机器学习, 指导老师: 孟德宇.

1 理论: EM 算法的思想与原理

1.1 EM 算法的思想

EM 算法的目标是, 最大化含有潜在变量的模型的似然函数, 从而得到合理的参数估计. 用 $\boldsymbol{X}$ 表示观测得到的变量, $\boldsymbol{Z}$ 表示观测变量背后的潜在变量 (例如, 某次观测来自哪个分布), 我们通过以下两步来进行 EM 算法:

- 在 $\mathbb{E}$ 步, 设参数的初值为 $\boldsymbol{\theta}^{\text{old}}$, 通过对 $\boldsymbol{Z}$ 的后验分布求期望, 得到似然函数
$$\mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right) = \sum_{\boldsymbol{Z}} \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \cdot p\left(\boldsymbol{Z}\middle|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}\right);$$
- 在 $\mathbb{M}$ 步, 通过最大化似然函数, 得到新参数
$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}\right).$$

通过以上两步不断迭代, 直到 $\left\|\boldsymbol{\theta}^{\text{new}} - \boldsymbol{\theta}^{\text{old}}\right\|$ 充分小, 或者运算次数充分大, 得到参数估计 $\hat{\boldsymbol{\theta}}$. 通常情况下, 为了方便, 记 $\gamma(\boldsymbol{Z}) = p\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}\right)$, 表示 $\boldsymbol{Z}$ 在参数 $\boldsymbol{\theta}^{\text{old}}$ 下的后验分布. 根据 Bayes 公式, 有后验分布的计算公式
$$\gamma(\boldsymbol{Z}) = p\left(\boldsymbol{Z}\middle|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}\right) = \frac{p\left(\boldsymbol{X}\middle|\boldsymbol{Z}, \boldsymbol{\theta}^{\text{old}}\right) \cdot p\left(\boldsymbol{Z}\middle|\boldsymbol{\theta}^{\text{old}}\right)}{\sum_{\boldsymbol{Z}} p\left(\boldsymbol{X}\middle|\boldsymbol{Z}, \boldsymbol{\theta}^{\text{old}}\right) \cdot p\left(\boldsymbol{Z}\middle|\boldsymbol{\theta}^{\text{old}}\right)},$$
其中 $p\left(\boldsymbol{Z}\middle|\boldsymbol{\theta}^{\text{old}}\right)$ 表示 $\boldsymbol{Z}$ 在参数 $\boldsymbol{\theta}^{\text{old}}$ 下的先验分布.

1.2 EM 算法的原理

EM 算法实际上实现了对观测数据的极大似然估计, 也即最大化观测数据的似然函数 $\mathcal{L}(\boldsymbol{\theta})$. 考虑观测数据的似然函数
$$\mathcal{L}(\boldsymbol{\theta}) = \ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \ln \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}).$$
为了估计 $\mathcal{L}(\boldsymbol{\theta})$ 的一个下界, 引入 $\boldsymbol{Z}$ 的一个可能的分布函数 $q(\boldsymbol{Z})$, 并应用 Jensen 不等式得
$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= \ln \sum_{\boldsymbol{Z}} \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{q(\boldsymbol{Z})} \cdot q(\boldsymbol{Z}) \\
&\geq \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{q(\boldsymbol{Z})} \\
&= \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) - \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \ln q(\boldsymbol{Z}).
\end{aligned}$$

通常, 记

$$\mathcal{F}(q,\boldsymbol{\theta}) := \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta}) - \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \ln q(\boldsymbol{Z}),$$

则有 $\boldsymbol{\theta} \geq \mathcal{F}(q,\boldsymbol{\theta})$ 对任意的分布 q 成立. 称 $\mathcal{F}(q,\boldsymbol{\theta})$ 为**自由能**. EM 算法实际上:

- 在 $\mathbb{E}$ 步, 找到

$$q^{\text{new}} = \arg\max_{q} \mathcal{F}\left(q, \boldsymbol{\theta}^{\text{old}}\right);$$

- 在 $\mathbb{M}$ 步, 找到

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{F}\left(q^{\text{new}}, \boldsymbol{\theta}\right).$$

根据上面的过程, 我们知道一定有 $\mathcal{F}\left(q^{\text{new}},\boldsymbol{\theta}^{\text{new}}\right) \geq \mathcal{F}\left(q^{\text{old}},\boldsymbol{\theta}^{\text{old}}\right)$, 同时, 对于似然函数而言, 可以证明 $\mathcal{L}\left(\boldsymbol{\theta}^{\text{new}}\right) \geq \mathcal{L}\left(\boldsymbol{\theta}^{\text{old}}\right)$. 不断进行迭代, 即可最大化似然函数 $\mathcal{L}(\boldsymbol{\theta})$.

2 示例 1: 质量不同的硬币的参数估计

2.1 问题背景

假设桌子上有若干个两种不同的硬币 (分别记为硬币 A 和硬币 B), 抛掷硬币 A 和硬币 B 时, 正面朝上的概率不同. 现进行如下试验: 随机地从桌子上选取一枚硬币, 抛掷若干次, 记录正面朝上和反面朝上的次数. 共进行若干次试验. 使用 *EM 算法*估计:

(1) 从桌子上选取到硬币 A 的概率;

(2) 从桌子上选取到硬币 B 的概率;

(3) 硬币 A 正面朝上的概率;

(4) 硬币 B 正面朝上的概率.

使用模拟数据进行计算, 并分析结果.

2.2 理论基础

2.2.1 符号说明

设共进行 n 次试验, 每次试验抛掷硬币 m 次. 用 X 表示某次试验中硬币正面朝上的次数, Z 表示硬币的类型 (取值为 $\{A, B\}$), 符号说明如**表 1**所示.

表 1: 符号说明

符号	说明
π_A	从桌子上选取到硬币 A 的概率
π_B	从桌子上选取到硬币 B 的概率
θ_A	硬币 A 正面朝上的概率
θ_B	硬币 B 正面朝上的概率
γ	Z 的后验分布

2.2.2 EM 算法的迭代公式

设 $x_1, x_2, \cdots, x_n$ 为每次试验中硬币正面朝上的次数, 旧参数 $\boldsymbol{\theta}^{\text{old}} = (\theta_A^{\text{old}}, \theta_B^{\text{old}}, \pi_A^{\text{old}}, \pi_B^{\text{old}})$, 则在给定条件 $X = x_i$ 下, Z 的后验分布

$$\gamma_i\left(A, \boldsymbol{\theta}^{\text{old}}\right) = \frac{\left(\theta_A^{\text{old}}\right)^{x_i} \cdot \left(1-\theta_A^{\text{old}}\right)^{m-x_i} \cdot \pi_A^{\text{old}}}{\left(\theta_A^{\text{old}}\right)^{x_i} \cdot \left(1-\theta_A^{\text{old}}\right)^{m-x_i} \cdot \pi_A^{\text{old}} + \left(\theta_B^{\text{old}}\right)^{x_i} \cdot \left(1-\theta_B^{\text{old}}\right)^{m-x_i} \cdot \pi_B^{\text{old}}},$$

$$\gamma_i\left(B, \boldsymbol{\theta}^{\text{old}}\right) = \frac{\left(\theta_B^{\text{old}}\right)^{x_i} \cdot \left(1-\theta_B^{\text{old}}\right)^{m-x_i} \cdot \pi_B^{\text{old}}}{\left(\theta_A^{\text{old}}\right)^{x_i} \cdot \left(1-\theta_A^{\text{old}}\right)^{m-x_i} \cdot \pi_A^{\text{old}} + \left(\theta_B^{\text{old}}\right)^{x_i} \cdot \left(1-\theta_B^{\text{old}}\right)^{m-x_i} \cdot \pi_B^{\text{old}}},$$

且有参数的迭代公式

$$\theta_A^{\text{new}} = \frac{1}{n} \cdot \frac{\sum\limits_{1 \le i \le n} x_i \cdot \gamma_i(A, \boldsymbol{\theta}^{\text{old}})}{\sum\limits_{1 \le i \le n} \gamma_i(A, \boldsymbol{\theta}^{\text{old}})}, \quad \theta_B^{\text{new}} = \frac{1}{n} \cdot \frac{\sum\limits_{1 \le i \le n} x_i \cdot \gamma_i(B, \boldsymbol{\theta}^{\text{old}})}{\sum\limits_{1 \le i \le n} \gamma_i(B, \boldsymbol{\theta}^{\text{old}})},$$

$$\pi_A^{\text{new}} = \frac{1}{n} \cdot \sum_{1 \le i \le n} \gamma_i\left(A, \boldsymbol{\theta}^{\text{old}}\right), \quad \pi_B^{\text{new}} = \frac{1}{n} \cdot \sum_{1 \le i \le n} \gamma_i\left(B, \boldsymbol{\theta}^{\text{old}}\right).$$

2.3 数值结果

2.3.1 模拟数据的生成

生成模拟数据时, 我们考虑进行 $n = 10$ 次试验, 每次试验抛掷的硬币次数 $m = 10$, 并设硬币 A 正面朝上的概率是 0.3, 硬币 B 正面朝上的概率是 0.6, 选取到硬币 A 的概率是 0.6, 选取到硬币 B 的概率是 0.4. 生成的模拟数据如**表 2**所示, 存储在`data_1_1.mat` 中.

表 2: 模拟数据

试验	1	2	3	4	5	6	7	8	9	10
硬币	B	B	A	A	B	A	A	B	B	A
正面次数	7	6	3	3	7	2	2	7	6	3

另外, 再考虑进行 $n = 1000$ 次试验, 生成的模拟数据存储在`data_1_2.mat` 中.

2.3.2 EM 算法的结果

对**表 2**, 也即`data_1_1.mat` 中的数据, 应用 *EM 算法*, 计算得到 $\hat{\theta}_A = 0.2774, \hat{\theta}_B = 0.6392$, $\hat{\pi}_A = 0.4953$, $\hat{\pi}_B = 0.5047$. 再对`data_1_2.mat` 中的数据进行计算, 得到 $\hat{\theta}_A = 0.3082$, $\hat{\theta}_B = 0.6146, \hat{\pi}_A = 0.5959, \hat{\theta}_B = 0.4041$.

2.3.3 结果的简单分析

样本量的影响

在进行模拟时, 发现估计的参数 $\hat{\theta}_A, \hat{\theta}_B, \hat{\pi}_A, \hat{\pi}_B$ 和参数的真实值差距较大; 同时, 对于来自同一个总体的不同模拟数据, 不同的模拟所得到的结果之间的差别也较大. 同样选取总体的参数 $\theta_A = 0.3, \theta_B = 0.6, \pi_A = 0.6, \pi_B = 0.4$, 重复进行模拟, 每次模拟进行 10 次试验, 每次试验抛硬币 10 次, 得到的参数估计如表 3所示.

表 3: 每次模拟进行 10 次试验, 重复模拟的结果

模拟	1	2	3	4	5
$\hat{\theta}_A$	0.3342	0.4109	0.4319	0.4400	0.2644
$\hat{\theta}_B$	0.5372	0.6609	0.5782	0.4400	0.7058
$\hat{\pi}_A$	0.5280	0.7635	0.6030	0.5245	0.8286
$\hat{\pi}_B$	0.4720	0.2365	0.3970	0.4755	0.1714

在每次模拟中, 都有 $\hat{\theta}_A < \hat{\theta}_B, \hat{\pi}_A > \hat{\pi}_B$, 但是变化非常大. 猜测出现这样的结果, 是因为试验的次数太少. 若改成进行 1000 次试验, 同样只抛硬币 10 次, 结果如表 4所示.

表 4: 每次模拟进行 1000 次试验, 重复模拟的结果

模拟	1	2	3	4	5
$\hat{\theta}_A$	0.2962	0.2961	0.3303	0.2772	0.3087
$\hat{\theta}_B$	0.6151	0.6191	0.6228	0.5794	0.5984
$\hat{\pi}_A$	0.6209	0.6418	0.7045	0.5563	0.5709
$\hat{\pi}_B$	0.3791	0.3582	0.2955	0.4437	0.4291

注意到这次计算的结果更加合理, $\hat{\theta}_A \approx 0.3, \hat{\theta}_B \approx 0.6, \hat{\pi}_A \approx 0.6, \hat{\pi}_B \approx 0.4$, 这启发我们选取的样本量要充分大, 才能更容易看出结果的规律.

初值的影响

在迭代的开始, 我们设置初值接近总体的真实参数值. 考虑对`data_1_2.mat` 中的数据进行处理, 但是在这里修改初值, 以检查 *EM* 算法是否会收敛到上面的结果.

表 5: 修改参数的初值, 模拟的结果

$\boldsymbol{\theta}$ 的初值	$\boldsymbol{\pi}$ 的初值	$\hat{\boldsymbol{\theta}}$	$\hat{\boldsymbol{\pi}}$
$(0.3, 0.6)$	$(0.6, 0.4)$	$(0.3082, 0.6146)$	$(0.5959, 0.4041)$
$(0.3, 0.6)$	$(0.9, 0.1)$	$(0.3082, 0.6146)$	$(0.5959, 0.4041)$
$(0.3, 0.6)$	$(0.1, 0.9)$	$(0.3082, 0.6146)$	$(0.5959, 0.4041)$
$(0.1, 0.9)$	$(0.6, 0.4)$	$(0.3082, 0.6146)$	$(0.5959, 0.4041)$
$(0.9, 0.1)$	$(0.6, 0.4)$	$(0.6146, 0.3082)$	$(0.4041, 0.5959)$

根据本次模拟发现, 如果所给的初值满足 $\theta_A < \theta_B$, 那么计算的结果不会有太大的变化; 然而, 如果 $\theta_A > \theta_B$, 最后估计出来的参数是和原先的结果相反的. 这是因为 θ_A 决定了硬币 A, 而 θ_B 决定了硬币 B. 如果两个参数交换了位置, 那么可以认为程序“认为”的硬币 A 其实是硬币 B, 而硬币 B 其实是硬币 A, 这导致了估计出来的参数和理想的参数相反.

3 示例 2: 混合正态总体的参数估计

3.1 问题背景

假设一批数据可能来自三个不同的一维正态分布总体. 使用 *EM 算法*估计:

(1) 数据来自三个正态总体的概率;

(2) 三个正态总体的期望;

(3) 三个正态总体的方差.

使用模拟数据进行计算, 并分析结果. 最后, 选择真实数据, 估计其参数.

3.2 理论基础

3.2.1 符号说明

设样本量为 n, 用 X 表示数据值, Z 表示数据所来自的正态分布总体 (取值为 $\{1,2,3\}$), 符号说明如**表 6**所示.

表 6: 符号说明

符号	说明
π_i	数据来自第 i 个正态总体的概率
μ_i	第 i 个正态总体的均值
σ_i^2	第 i 个正态总体的方差
γ	Z 的后验分布

3.2.2 EM 算法的迭代公式

设 $x_1, x_2, \cdots, x_n$ 为样本, 旧参数 $\boldsymbol{\theta}^{\text{old}} = \left(\boldsymbol{\pi}^{\text{old}}, \boldsymbol{\mu}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}\right)$, 其中 $\boldsymbol{\pi}^{\text{old}} = \left(\pi_1^{\text{old}}, \pi_2^{\text{old}}, \pi_3^{\text{old}}\right)$, $\boldsymbol{\mu}^{\text{old}} = \left(\mu_1^{\text{old}}, \mu_2^{\text{old}}, \mu_3^{\text{old}}\right)$, $\boldsymbol{\sigma}^{\text{old}} = \left(\sigma_1^{\text{old}}, \sigma_2^{\text{old}}, \sigma_3^{\text{old}}\right)$, 并用 $p_i\left(\cdot \middle| \boldsymbol{\theta}^{\text{old}}\right)$ 表示第 i 个正态总体的概率密度, 则在给定条件 $X = x_i$ 下, Z 的后验分布

$$\gamma_i\left(k, \boldsymbol{\theta}^{\text{old}}\right) = \frac{\pi_k \cdot p_k\left(x_i \middle| \boldsymbol{\theta}^{\text{old}}\right)}{\sum\limits_{1 \le j \le 3} \pi_j \cdot p_j\left(x_i \middle| \boldsymbol{\theta}^{\text{old}}\right)}, \quad 1 \le k \le 3,$$

且有参数的迭代公式

$$\mu_k^{\text{new}} = \frac{\sum\limits_{1\le i\le n} x_i \cdot \gamma_i\left(k, \boldsymbol{\theta}^{\text{old}}\right)}{\sum\limits_{1\le i\le n} \gamma_i\left(k, \boldsymbol{\theta}^{\text{old}}\right)}, \quad \left(\sigma_k^{\text{new}}\right)^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - \left(\mu_k^{\text{new}}\right)^2,$$

以及

$$\pi_k^{\text{new}} = \frac{1}{n} \cdot \sum_{1\le i\le n} \gamma_i\left(k, \boldsymbol{\theta}^{\text{old}}\right).$$

3.3 数值结果

3.3.1 数据的生成与选取

模拟数据

生成模拟数据时, 我们考虑样本量 $n = 300$, 第一个正态总体为 $\mathcal{N}(-5, 1)$, 第二个正态总体为 $\mathcal{N}(0, 1)$, 第三个正态总体为 $\mathcal{N}(5, 1)$, 来自三个正态总体的概率分别为 0.2、0.3 和 0.5. 生成模拟数据如图 **1**所示, 其中有 59 个来自第一个正态总体, 有 93 个来自第二个正态总体, 有 148 个来自第三个正态总体. 数据存储在文件`data_2.mat` 中.

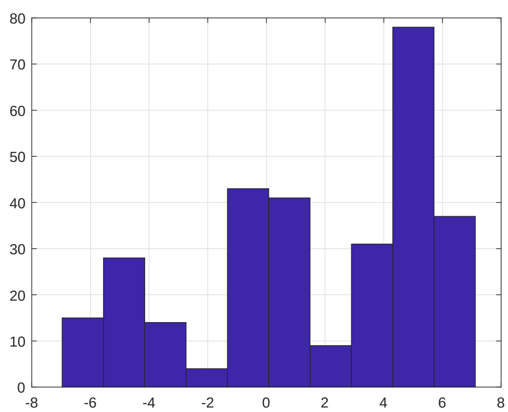

图 1: 模拟数据

真实数据

选取的真实数据来自经典的鸢尾花数据集, 选取的是第一个属性 (萼片长度), 通过计算得到, 三个总体的均值分别为 5.006, 5.936 和 6.588, 方差分别为 0.3489, 0.5110 和 0.6295. 画出图像如图图 **2**所示, 发现大约可以分成三个正态分布总体. 将数据存储在`iris.mat` 中.

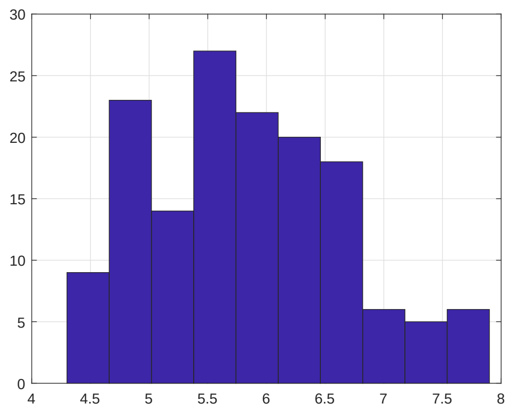

图 2: 150 朵不同类型的鸢尾花的萼片长度

3.3.2 EM 算法的结果

模拟数据

对`data_2.mat` 应用 *EM* 算法, 经过多次迭代, 计算得到 $\hat{\boldsymbol{\mu}} = (-4.9320, 0.1451, 4.9858)$, $\hat{\boldsymbol{\sigma}} = (0.9370, 0.9338, 0.9315)$, $\hat{\boldsymbol{\pi}} = (0.1903, 0.3161, 0.4936)$.

真实数据

对数据`iris.mat` 应用 *EM* 算法, 经过多次迭代, 计算得到 $\hat{\boldsymbol{\mu}} = (4.9073, 5.8534, 6.6184)$, $\hat{\boldsymbol{\sigma}} = (0.2836, 0.5198, 0.6523)$, $\hat{\boldsymbol{\pi}} = (0.2596, 0.4326, 0.3079)$.

3.3.3 结果的简单分析

迭代中出现 NaN 的原因

最初在计算正态分布的密度函数时, 使用的函数是`normpdf(x, mu, sigma)`. 然而, 在运行代码的时候发现, 对于某些模拟数据, 输出的参数结果是`NaN`. 通过对每次迭代的结果进行分析发现, 某个正态总体的方差可能极小, 在程序的计算中被当成 0, 而正态分布的方差不能为 0, 这将会导致`normpdf(x, mu, sigma) = NaN`.

为了解决该问题, 在程序中编写了函数`fixednormpdf(x, mu, sigma)`. 当正态总体的方差充分小 (例如, 在这里小于 0.0001 时), 使用参数为`mu` 的两点分布的概率密度函数代替正态分布的概率密度函数. 这样, 就可以修正`normpdf(x, mu, sigma)` 的问题.

```
%% 修正的正态分布密度函数

function p = fixednormpdf(x, mu, sigma)
```

```
        if sigma < 0.0001
            if x == mu
                p = 1;
            else
                p = 0;
            end
        else
            p = normpdf(x, mu, sigma);
        end
    end
```

另外, 若选取的初值不合适, 导致某些 γ 取到 0, 也会使结果出现NaN. 这就需要我们选取合适的初值.

重复模拟的结果分析

在这里考虑样本量 $n=300$, 三个正态总体分别为 $\mathcal{N}(-5,1), \mathcal{N}(0,1)$ 和 $\mathcal{N}(5,1)$, 来自三个正态总体的概率分别为 0.2, 0.3 和 0.5, 进行多次模拟, 得到的结果如**表 7**所示.

表 7: 重复模拟的结果

模拟	$\hat{\boldsymbol{\mu}}$	$\hat{\boldsymbol{\sigma}}$	$\hat{\boldsymbol{\pi}}$
1	$(-4.8651, 0.0160, 5.0255)$	$(1.0869, 0.8971, 1.0337)$	$(0.2628, 0.2604, 0.4768)$
2	$(-5.1248, -0.0030, 4.9971)$	$(0.9847, 0.9134, 1.0851)$	$(0.1773, 0.2872, 0.5355)$
3	$(-5.2781, -0.0432, 5.0064)$	$(1.0731, 0.7807, 1.0363)$	$(0.1667, 0.2934, 0.5399)$
4	$(-4.9777, 0.1144, 5.0673)$	$(0.9857, 1.0912, 0.9758)$	$(0.1787, 0.3272, 0.4941)$
5	$(-5.2118, -0.0812, 5.1286)$	$(0.8073, 1.0113, 0.8991)$	$(0.1572, 0.3265, 0.5163)$

根据结果发现, $\hat{\boldsymbol{\mu}} \approx (-0.5, 0, 0.5)$, $\hat{\boldsymbol{\sigma}} \approx (1,1,1)$, $\hat{\boldsymbol{\pi}} \approx (0.2, 0.3, 0.5)$, 这与总体的参数是接近的. 这也说明了算法的有效性.

3.3.4 基于 EM 算法的鸢尾花分类

在上面的基础上, 我们利用鸢尾花的萼片长度, 来对鸢尾花进行分类. 首先, 根据估计所得的参数 $\hat{\boldsymbol{\mu}}$ 和 $\hat{\boldsymbol{\sigma}}$, 可以确定三个正态总体及其密度函数, 见图 3.

据此, 用 c_i 表示 x_i 所属的类, 令 $c_i = \arg\max_k p_k\left(x_1 | \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}\right)$, 即可对鸢尾花进行分类. 在

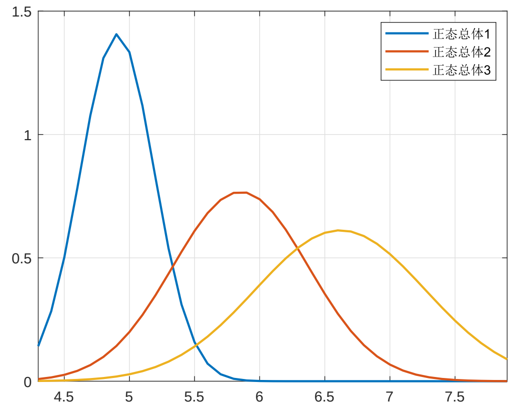

图 3: 三个正态总体的密度函数

这里考虑的数据维数较低, 为了得到更精确的分类, 可以考虑二维、三维甚至四维正态总体.

参考文献

[1] 周志华. 机器学习 [M]. 北京: 清华大学出版社, 2016.
[2] 李航. 统计学习方法 [M]. 北京: 清华大学出版社, 2012.

附录

A 示例 1 的代码

代码使用 MATLAB 编写, 存储在code_1.m 中, 与data_1_1.mat、data_1_2.mat 同目录.

```
%% 初始化

clc;
clear;
close;

%% 数据

% load data_1_1.mat; % 读取数据1
% load data_1_2.mat; % 读取数据2
n = 1000; % 试验次数
m = 10; % 每次试验抛硬币的次数
theta0 = [0.3; 0.6]; % 硬币正面朝上的概率
pi0 = [0.6; 0.4]; % 抽到两个硬币的概率
c = randsrc(1, n, [1 : 2; pi0']); % 1表示硬币A, 2表示硬币B
x = binornd(m, theta0(c))'; % 生成模拟数据

%% EM算法

theta = [0.3; 0.6];
pi = [0.6; 0.4];
gamma = zeros(2, n);
for i = 1 : 200
    for k = 1 : 2
```

```
        gamma(k, :) = binopdf(x, m, theta(k)) .* pi(k) ./ (binopdf(x,
            m, theta(1)) .* pi(1) + binopdf(x, m, theta(2)) .* pi(2));
    end
    theta = sum(x .* gamma, 2) ./ (m * sum(gamma, 2));
    pi = sum(gamma, 2) ./ sum(sum(gamma));
end

%% 输出结果

theta
pi
```

B　示例 2 的代码

代码使用 MATLAB 编写, 存储在code_2.m 中, 与data_2.mat、iris.mat 同目录.

```
%% 初始化

clc;
clear;
close;

%% 数据

% load data_2.mat; % 读取模拟数据
% load iris.mat; % 读取真实数据
n = 300; % 样本量
mu0 = [-5; 0; 5]; % 正态总体的均值
sigma0 = [1; 1; 1]; % 正态总体的方差
pi0 = [0.2; 0.3; 0.5]; % 来自不同正态总体的概率
c = randsrc(1, n, [1 : 3; pi0']); % 来自不同正态分布
x = normrnd(mu0(c), sigma0(c))'; % 生成模拟数据

%% EM算法

mu = [-5; 0; 5]; % [5.006; 5.936; 6.588];
```

```matlab
sigma = [1; 1; 1]; % [0.3489; 0.5110; 0.6295];
pi = [0.2; 0.3; 0.5]; % [1/3; 1/3; 1/3];
gamma = zeros(3, n);
for i = 1 : 100
    gamma = pi .* fixednormpdf(x, mu, sigma) ./ sum(pi .* fixednormpdf
        (x, mu, sigma));
    mu = sum(x .* gamma, 2) ./ sum(gamma, 2);
    sigma = sqrt(sum((x - mu).^2 .* gamma, 2) ./ sum(gamma, 2));
    pi = sum(gamma, 2) ./ sum(sum(gamma));
end

%% 输出结果

mu
sigma
pi

%% 绘制图像

xspace = min(x) : 0.1 : max(x);
plot(xspace, normpdf(xspace, mu(1), sigma(1)), xspace, normpdf(xspace,
    mu(2), sigma(2)), xspace, normpdf(xspace, mu(3), sigma(3)), '
   LineWidth', 1.5);
xlim([min(x) max(x)]);
legend('正态总体1', '正态总体2', '正态总体3');
grid on;
```