利用 EM 聚类对鸢尾花数据集进行分类

统计 91 董晟渤

2021 年 11 月 28 日

摘要

鸢尾花数据集是数据科学中经典的数据集. 本文利用正态分布来描述总体, 并使用 EM 聚类算法, 成功地对鸢尾花数据集进行了分类.

首先, 本文简单介绍了 EM 算法与 EM 聚类的原理, 给出了使用 EM 聚类对数据分类的基本思想, 这是本文的理论依据.

接下来, 通过对鸢尾花数据进行预处理, 选取鸢尾花的花瓣长度与宽度作为分类依据, 并使用二维正态分布来描述总体. 在此基础上, 设计分类算法, 使用 MATLAB 编写代码, 最终将所给的鸢尾花数据分为山鸢尾、变色鸢尾和维吉尼亚鸢尾三类, 绘制出散点图, 并得到了三个总体对应的参数. 分类的准确度高达 97.33%, 说明了结果是可靠的.

最后, 为了对比选取不同的属性进行分类时的准确度, 本文考虑了选取一个属性和两个属性的所有情形, 分别计算准确度, 验证了选取鸢尾花的花瓣长度与宽度进行分类时的准确度是最高的.

关键词: 极大似然估计, EM 算法, EM 聚类, 正态分布.

目录

1 概述

1.1 EM 算法与 EM 聚类

在提及 EM 算法之前, 首先需要说一说数理统计中一种常用的参数估计方法: 极大似然估计. 在生活中, 人们往往认为概率最大的事情最有可能发生. 基于这个朴素的观念, 极大似然估计方法认为, 使得样本 $x_1, x_2, \cdots, x_n$ 的联合概率密度函数 (可以认为是概率) 最大的参数 $\hat{\theta} = T(x_1, x_2, \cdots, x_n)$ 可以用来估计参数 θ. EM 算法基于极大似然估计的思想, 是一种非常有效的参数估计方法. 基于 EM 算法的 EM 聚类, 被广泛地应用于机器学习与统计学习当中. 以下, 首先叙述 EM 算法的基本思路.

设 Θ 为参数空间, $\theta \in \Theta$ 为参数, 给定参数 θ 的初值 $\theta_0 \in [0, 1]$. 并设独立样本 $x_i \sim p_i(x; \theta)$, 其中 $1 \leq i \leq n$. 首先根据极大似然估计的思想, 考虑使得极大似然函数

$$L(x_1, x_2, \cdots, x_n; \theta) = \prod_{i=1}^{n} p_i(x_i, \theta) \quad \text{或} \quad \ln L(x_1, x_2, \cdots, x_n; \theta) = \sum_{i=1}^{n} \ln p_i(x_i, \theta)$$

取极大值的参数 $\hat{\theta}$, 并设

$$\hat{\theta} = T(x_1, x_2, \cdots, x_n).$$

在此基础上, 为了得到参数 θ 的精确估计, 对于 $k \geq 0$, 我们重复进行以下两个步骤:

“期望”步 设 θ_k 是参数 θ 的估计, 并设独立样本 $x_i^{(k)} \sim p_i(x; \theta_k)$, 其中 $1 \leq i \leq n$. 令

$$\bar{x}_1^{(k)} = \mathbb{E}x_1^{(k)}, \quad \bar{x}_2^{(k)} = \mathbb{E}x_2^{(k)}, \quad \cdots, \quad \bar{x}_n^{(k)} = \mathbb{E}x_n^{(k)}.$$

“极大”步 根据极大似然估计的结果, 令

$$\theta_{k+1} = T\left(\bar{x}_1^{(k)}, \bar{x}_2^{(k)}, \cdots, \bar{x}_n^{(k)}\right).$$

对于以上过程, 注意到当 $x_i^{(k)} \sim p_i(x; \theta_k)$ 时, 有

$$\bar{x}_i^{(k)} = \mathbb{E}x_i^{(k)} = \int_{\mathbb{R}} p_i(x; \theta_k) \mathrm{d}x = e_i(\theta_k),$$

其中 $e_i(\theta)$ 是与 k 无关的函数; 又根据

$$\theta_{k+1} = T\left(\bar{x}_1^{(k)}, \bar{x}_2^{(k)}, \cdots, \bar{x}_n^{(k)}\right) = T(e_1(\theta_k), e_2(\theta_k), \cdots, e_n(\theta_k)) = f(\theta_k),$$

其中 $f:\Theta\to\Theta,\theta\mapsto T(e_1(\theta),e_2(\theta),\cdots,e_n(\theta))$ 也是与 k 无关的函数. 从而, 根据上面可以得到递推式

$$\theta_{k+1}=f(\theta_k),\quad k\geq 0.$$

代入初值 θ_0 进行迭代, 则 $\lim\limits_{k\to\infty}\theta_k$ 即为 $\hat{\theta}$ 的一个估计. 以上的步骤被称为 **EM 算法**.

现在考虑这样的一个问题: 我们有一些样本 $x_1,x_2,\cdots,x_n$, 它们来自 $m(m\leq n)$ 个不同的总体 $p_1(x;\theta_1),p_2(x;\theta_2),\cdots,p_m(x;\theta_m)$, 并设这些总体的参数的初值分别为 θ_{10}, θ_{20}, $\cdots$, θ_{m0}. 我们想要判断出这些样本 $x_1,x_2,\cdots,x_n$ 到底来自哪个总体. 对于 $1\leq l\leq m$, 设 $x_{l1},x_{l2},\cdots,x_{ln_l}$ 是来自总体 $p_l(x;\theta_l)$ 的样本, 考虑使得极大似然函数

$$L_l(x_{l1},x_{l2},\cdots,x_{ln_l})=\prod_{i=1}^{n}p_l(x_{li},\theta_l)\quad \text{或}\quad \ln L_l(x_{l1},x_{l2},\cdots,x_{ln_l})=\sum_{i=1}^{n}\ln p_l(x_{li},\theta_l)$$

取极大值的参数 $\hat{\theta}_l$, 并设

$$\hat{\theta}_l=T_l(x_{l1},x_{l2},\cdots,x_{ln_l}).$$

在极大似然估计与 EM 算法的基础上, 为了对样本进行分类, 对于 $k\geq 0$, 我们重复进行以下两个步骤:

第一步 对于样本 $x_i(1\leq i\leq n)$, 设

$$p_{l_0}(x_i;\theta_{l_0,k})=\max_{1\leq l\leq m}p_l(x_i;\theta_{l,k}),$$

则将 x_i 分到第 l_0 类. 设第 l 类最终的元素个数为 $n_l^{(k)}$, 且分别为 $x_{l1},x_{l2},\cdots,x_{ln_l^{(k)}}$.

第二步 根据极大似然估计的结果, 令

$$\theta_{l,k+1}=T_l\left(x_{l1},x_{l2},\cdots,x_{ln_l^{(k)}}\right).$$

从初值 $\theta_{10},\theta_{20},\cdots,\theta_{m0}$ 开始, 将上面的过程不断进行下去, 不但可以得到第 l 个总体的参数 θ_l 的估计值, 还可以将所给的样本进行分类. 以上的方法被称为 **EM 聚类**.

1.2 鸢尾花数据集简介

鸢尾花数据集 (Iris.csv) 是常用的分类实验数据集, 由 Fisher 在 1936 年收集整理, 共有 150 组数据. 每个数据包含 4 个属性, 分别为花萼长度 (Sepal.Length)、花萼宽度

(Sepal.Width)、花瓣长度 (Petal.Length) 和花瓣宽度 (Petal.Width). 根据这四个属性, 可以将鸢尾花分成山鸢尾 (Setosa)、变色鸢尾 (Versicolour) 和维吉尼亚鸢尾 (Virginica) 三种. 图1是这三种不同的鸢尾花的图片.

图 1: 山鸢尾、变色鸢尾和维吉尼亚鸢尾

鸢尾花数据集是非常经典的数据集, 结果也更加好看 [1], 因此在本文中选择该数据集进行分类. 并且通过查阅资料得知, 花瓣长度和花瓣宽度可以很好地区分不同的鸢尾花, 这便大大降低了分类的难度.

2 问题的解决

2.1 数据的预处理

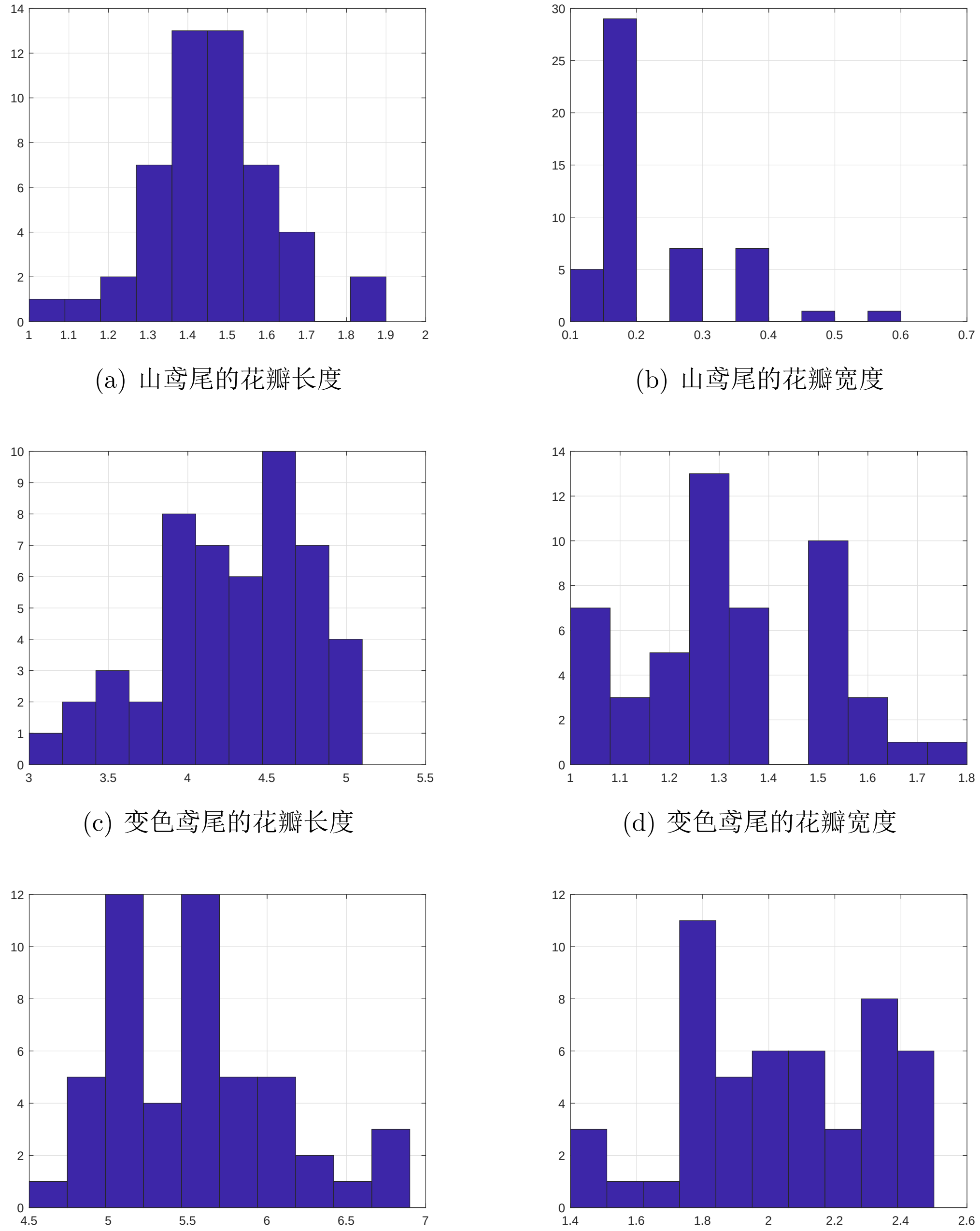
在开始实现 EM 聚类之前, 我们首先需要得到总体的分布. 为此, 我们在数据预处理的过程中, 绘制出三种不同的鸢尾花所对的花瓣长度和花瓣宽度的直方图如图2所示.

可以看出, 数据大多呈现“两边高、中间低”的趋势, 这一点在图2(a) 中尤其明显, 因此可以认为每组数据分别服从正态分布 [2]. 在这里, 我们同时考虑花瓣长度和花瓣宽度, 它们之间可能存在一定的相关性, 这可以用二维正态分布的相关系数 r 来描述. 从而, 我们假定总体服从二维正态分布 $\mathcal{N}(\mu_1, \mu_2; \sigma_1^2, \sigma_2^2; r)$.

在 EM 算法中, 为了得到三种分类的参数的初值, 我们还需要先计算出三种鸢尾花的数据所对的均值与标注差. 对于山鸢尾, 计算得花瓣长度的均值 $\mu_{11} = 1.462$, 标准差 $\sigma_{11} = 0.172$, 花瓣宽度的均值 $\mu_{12} = 0.246$, 标准差 $\sigma_{12} = 0.104$; 对于变色鸢尾, 计

[1]原本笔者选择的是上万个人的身高和体重的数据, 想要通过这些数据分辨出男性和女性, 但是这些数据十分密集, 最后分类的结果呈现“一刀切”的效果, 因此放弃该数据.

[2]事实上, 一般“默认”分布是正态分布



(a) 山鸢尾的花瓣长度

(b) 山鸢尾的花瓣宽度

(c) 变色鸢尾的花瓣长度

(d) 变色鸢尾的花瓣宽度

(e) 维吉尼亚鸢尾的花瓣长度

(f) 维吉尼亚鸢尾的花瓣宽度

图 2: 三种不同的鸢尾花的直方图

算得花瓣长度的均值 $\mu_{21}=4.26$, 标准差 $\sigma_{21}=0.465$, 花瓣宽度的均值 $\mu_{22}=1.326$, 标准差 $\sigma_{22}=0.196$; 对于维吉尼亚鸢尾, 计算得花瓣长度的均值 $\mu_{31}=5.552$, 标准差 $\sigma_{31}=0.546$, 花瓣宽度的均值 $\mu_{32}=2.026$, 标准差 $\sigma_{32}=0.272$.

查阅文献得到, 设 (x_1,y_1), (x_2,y_2), $\cdots$, (x_n,y_n) 是来自总体 $\mathcal{N}(\mu_1,\mu_2;\sigma_1,\sigma_2;r)$ 的样本, 则 μ_1,σ_1 的极大似然估计分别是样本 $x_1,x_2,\cdots,x_n$ 的均值和标准差, μ_2,σ_2 的极大似然估计分别是样本 $y_1,y_2,\cdots,y_n$ 的均值和标准差, 而 r 的极大似然估计是 (x_1,y_1), (x_2,y_2), $\cdots$, (x_n,y_n) 的相关系数.

2.2 算法的设计

根据 EM 聚类的原理, 我们设计算法1来对鸢尾花数据集进行分类.

算法 1: 利用花瓣长度和花瓣宽度对鸢尾花数据集进行分类

input 150 组鸢尾花的花瓣长度 $x(1,:)$ 和花瓣宽度 $x(2,:)$;

对 $j\in\{1,2,3\}$, 设置正态总体 $\mathcal{N}_j(\mu_{j1},\mu_{j2};\sigma_{j1}^2,\sigma_{j2}^2;r_j)$ 的初值;

$p_j(x(1,:),x(2,:))$ 表示第 j 个正态总体的概率密度;

$c(i)=j$ 表示第 i 组数据被分到第 j 个总体;

for 第 k 次计算 **do**

  for 第 i 组数据 **do**

    if $p_{j_0}(x(1,i),x(2,i))=\max\{p_j(x(1,i),x(2,i)),j\in\{1,2,3\}\}$ **then**

      $c(i)=j_0$;

    end

  end

  for 第 j 个分类 **do**

    利用第 j 个分类中新的样本, 计算 $\mu_{j1},\mu_{j2},\sigma_{j1}^2,\sigma_{j2}^2,r_j$;

    得到新的正态总体 $\mathcal{N}_j(\mu_{j1},\mu_{j2};\sigma_{j1}^2,\sigma_{j2}^2;r_j)$ 和概率密度 $p_j(x(1,:),x(2,:))$;

  end

end

计算分类的准确度 η;

output 三个正态总体的参数;

output 带有标签的数据的散点图;

output 分类的准确度 η;

2.3 分类的结果

我们所得的分类结果如图3所示. 结果显示, 山鸢尾的花瓣长度和花瓣宽度远小于变色鸢尾和维吉尼亚鸢尾. 变色鸢尾的花瓣长度和花瓣宽度都比维吉尼亚鸢尾略小一些.

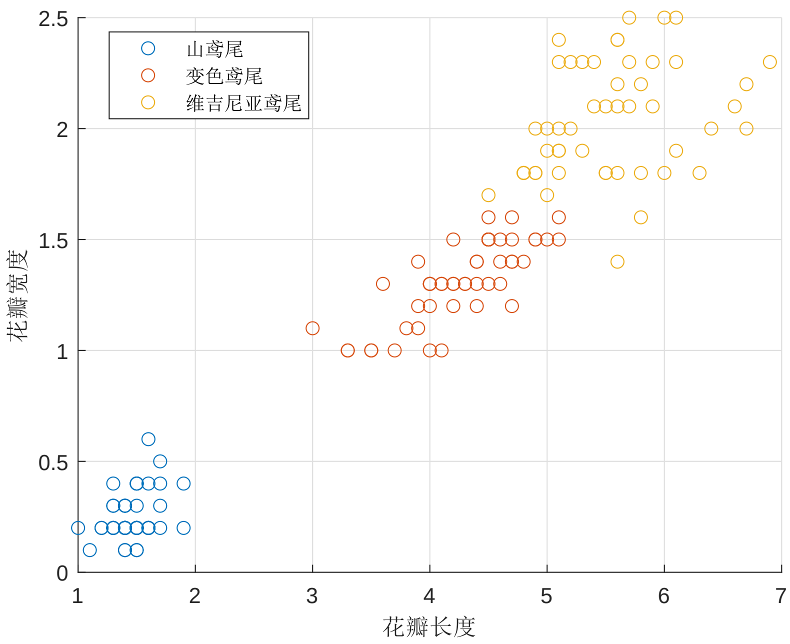


图 3: 鸢尾花分类的结果

同时, 三个正态总体的最终参数如表1所示. 参数的值与我们所设置的初值接近, 同时根据 $r > 0$ 得知, 花瓣的长度和花瓣的宽度有一定的正相关关系. 正相关的关系在变色鸢尾中体现最明显.

表 1: 鸢尾花总体的最终参数

总体	山鸢尾	变色鸢尾	维吉尼亚鸢尾
μ_1	1.4620	4.2660	5.5460
μ_2	0.2460	1.3160	2.0360
σ_1	0.1719	0.4740	0.5529
σ_2	0.1043	0.1793	0.2567
r	0.3316	0.7831	0.3124

最后, 为了检验分类的准确性, 我们将结果与数据中的标签进行对比, 发现准确度高达 97.3333%, 说明分类的结果是极可靠的.

2.4 准确度对比

本文所使用的鸢尾花数据集共有四个属性. 根据对该数据集处理的经验, 选取后两个属性所得到的准确率更高, 这也是上一节中我们解决问题的方法. 除了选取后两个属性以外, 我们也可以选取任意两个属性, 甚至仅选取一个属性. 为了进行对比, 本文考虑了选取一个属性和两个属性的所有情形.

首先, 在算法1的基础上, 我们设计使用单个属性进行分类的算法2.

算法 2: 利用单个属性对鸢尾花数据集进行分类

input 150 组鸢尾花的某个属性 $x(1,:)$;
对 $j \in \{1,2,3\}$, 设置正态总体 $\mathcal{N}_j(\mu_j, \sigma_j^2)$ 的初值;
$p_j(x(:))$ 表示第 j 个正态总体的概率密度;
$c(i) = j$ 表示第 i 组数据被分到第 j 个总体;
for 第 k 次计算 **do**
  for 第 i 组数据 **do**
    if $p_{j_0}(x(1,i), x(2,i)) = \max\{p_j(x(1,i), x(2,i)), j \in \{1,2,3\}\}$ **then**
      $c(i) = j_0$;
    end
  end
  for 第 j 个分类 **do**
    利用第 j 个分类中新的样本, 计算 μ_j, σ_j^2;
    得到新的正态总体 $\mathcal{N}(\mu_j, \sigma_j^2)$ 和概率密度 $p_j(x(:))$;
  end
end
计算分类的准确度 η;
output 分类的准确度 η;

接下来, 计算出三种不同的鸢尾花各属性的均值 μ 和标准差 σ, 如表2所示. 表2的数据可以作为分类时参数的初值. 在表2的基础上, 我们首先选取单个属性, 利用算法2对鸢尾花进行分类, 所得的准确率填写到表3的对角线上; 在此之后, 我们再选取两个属性, 仿照算法1进行分类, 所得的准确率填写到表3的对应位置上.

表 2: 三种不同的鸢尾花的均值和标准差

类别	参数	花萼长度	花萼宽度	花瓣长度	花瓣宽度
山鸢尾	μ	5.006	3.428	1.462	0.246
	σ	0.349	0.375	0.172	0.104
变色鸢尾	μ	5.936	2.770	4.260	1.326
	σ	0.511	0.311	0.465	0.196
维吉尼亚鸢尾	μ	6.588	2.974	5.552	2.026
	σ	0.629	0.319	0.546	0.272

表 3: 选取不同的属性进行分类所得的准确度

	花萼长度	花萼宽度	花瓣长度	花瓣宽度
花萼长度	63.33%	78.00%	94.00%	96.00%
花萼宽度	78.00%	56.00%	90.00%	96.00%
花瓣长度	94.00%	90.00%	58.67%	97.33%
花瓣宽度	96.00%	96.00%	97.33%	47.33%

根据表3可以看出, 选取后两个属性进行分类, 准确度确实是最高的. 同时注意到, 仅选取一个属性进行分类, 准确度通常会比选取两个属性的准确度更低. 直觉上, 考虑的属性越多, 分类的准确度也应该越高.

3 总结与推广

3.1 结果的总结

本文首先用简洁的语言, 介绍了 EM 算法、EM 聚类的基本思想. 接下来, 为了对鸢尾花数据进行初步分类, 选取鸢尾花的花瓣长度和花瓣宽度作为分类的依据, 使用三个不同的二维正态分布总体来描述三种不同的鸢尾花, 并通过 EM 聚类算法, 成功地对鸢尾花数据集分类. 分类的结果如图3所示, 最终总体的参数如表1所示. 分类的准确度高达 97.33%, 说明分类的结果是极可靠的. 最后, 为了对比选取不同的属性进行分类的准确度, 本文考虑了选取一个属性和两个属性的所有情形. 最终的结果如表3所示, 并且验证了选取花瓣长度和花瓣宽度进行分类的准确度是最高的.

3.2 问题的推广

在解决了问题之后, 还可以从以下的角度推广问题:

- 在使用 EM 聚类时, 除了假定总体是正态分布以外, 还可以假定总体是均匀分布等. 对于均匀分布 $\mathcal{U}[a,b]$ 及来自该样本的总体 $x_1, x_2, \cdots, x_n$, 我们知道其参数的极大似然估计为

$$a = x_{(1)} = \min\{x_1, x_2, \cdots, x_n\}, \quad b = x_{(n)} = \max\{x_1, x_2, \cdots, x_n\}.$$

  这样的形式更容易进行计算. 对于一维的数据而言, 这样的假设也许是合理的. 但是, 对于二维的数据而言, 如果它们之间有一定的相关性 (比如花瓣宽度更大的花, 花瓣的长度有可能更大), 那么总体的分布用二维均匀分布来描述就不太合适. 设它们的相关系数为 r, 此时用二维正态分布 $\mathcal{N}(\mu_1, \mu_2; \sigma_1, \sigma_2; r)$ 是最自然也是最合理的.

- 除了使用二维正态分布来描述总体以外, 还可以尝试三维正态分布或是四维正态分布. 但是, n 维正态分布涉及到的参数会有

$$n + n + \binom{n}{2} = \frac{n^2 + 3n}{2}$$

  个, 例如当 $n = 3$ 时, 我们在每个正态总体中都需要处理 9 个参数, 解决问题的复杂度就大大提高了.

参考文献

[1] 茆诗松, 程依明, 濮晓龙. 概率论与数理统计教程 [M]. 北京: 高等教育出版社, 2019.

[2] Dempster A P, Laird N M, Rubin D B. *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, 1977, 39(1):1-38.

附录

附录 A 所用软件

在本报告的撰写过程中, 使用到了如下软件.

- 基于 Visual Studio Code 的 LaTeX, 用于排版论文;
- MathWorks MATLAB R2019b, 用于实现 EM 算法及绘制图像.

附录 B 代码

这是算法1的实现.

```
%% 基本参数

data = xlsread('/Iris.csv');
x = [data(:, 4)'; data(:, 5)']; % 选取数据
n = size(x, 2); % 数据量
c = ones(1 ,n); % 分类，1表示setosa，2表示versicolor，3表示virginica
theta = [1.462 4.26 5.552 % mu1
     0.246 1.326 2.026 % mu2
     0.172 0.465 0.546% sigma1
     0.104 0.196 0.272% sigma2
     0.5 0.5 0.5];  % r
k = 100; % 迭代次数

%% 二维正态分布的密度函数
```

```
f = @ (x, y, mu1, mu2, sigma1, sigma2, r) ...
    1/(2*pi*sigma1*sigma2*sqrt(1-r^2)) .* ...
    exp(-1/(2-2*r^2).*((x-mu1).^2./sigma1.^2 - ...
        2*r.*(x-mu1).*(y-mu2)./(sigma1*sigma2) + (y-mu2).^2./sigma2^2));

%% EM聚类

for l = 1 : k

    % 第一步
    % 根据第k-1次计算时的参数theta，对x(:,i)进行分类

    for i = 1 : n
        if f(x(1,i), x(2,i), theta(1,1), theta(2,1), theta(3,1), ...
            theta(4,1), theta(5,1)) ...
                > f(x(1,i), x(2,i), theta(1,2), theta(2,2), ...
                    theta(3,2), theta(4,2), theta(5,2)) ...
                && f(x(1,i), x(2,i), theta(1,1), theta(2,1), ...
                    theta(3,1), theta(4,1), theta(5,1)) ...
                > f(x(1,i), x(2,i), theta(1,3), theta(2,3), ...
                    theta(3,3), theta(4,3), theta(5,3))
            c(i) = 1;
        elseif f(x(1,i), x(2,i), theta(1,2), theta(2,2), theta(3,2), ...
            theta(4,2), theta(5,2)) ...
                > f(x(1,i), x(2,i), theta(1,3), theta(2,3), ...
                    theta(3,3), theta(4,3), theta(5,3))
            c(i) = 2;
        else
            c(i) = 3;
        end
    end

    % 第二步
    % 根据第k次分类的结果计算参数theta

    for j = 1 : 3
```

```
            s1 = [0; 0]; % 第j类的样本的和
            s2 = [0; 0]; % 第j类的样本的平方和
            s3 = 0; % 第j类的样本的乘积和
            count = 0; % 第j类的样本的个数
            for i = 1 : n
                if c(i) == j
                    s1 = s1 + x(:,i);
                    s2 = s2 + x(:,i).^2;
                    s3 = s3 + x(1,i)*x(2,i);
                    count = count + 1;
                end
            end
            ave = s1./count; % 第j类的样本的均值
            var = 1/count.*(s2 - count.*ave.^2); % 第j类的样本的方差
            cov = 1/count.*(s3 - count.*ave(1,1)*ave(2,1)); % ...
                第j类的样本的协方差
            cor = cov/sqrt(var(1,1)*var(2,1)); % 第j类的样本的相关系数
            theta(:,j) = [ave(1,1); ave(2,1); sqrt(var(1,1)); ...
                sqrt(var(2,1)); cor]; % 正态分布的极大似然估计
        end
    end

%% 画出散点图

count1 = 1;
count2 = 1;
count3 = 1;
for i = 1 : n
    if c(i) == 1
        x1(:, count1) = x(:, i);
        count1 = count1 + 1;
    elseif c(i) == 2
        x2(:, count2) = x(:, i);
        count2 = count2 + 1;
    else
        x3(:, count3) = x(:, i);
        count3 = count3 + 1;
```

```
        end
end
scatter(x1(1, :), x1(2, :));
hold on;
scatter(x2(1, :), x2(2, :));
hold on;
scatter(x3(1, :), x3(2, :));
xlabel('花瓣长度');
ylabel('花瓣宽度');
legend('山鸢尾', '变色鸢尾', '维吉尼亚鸢尾');
grid on;

%% 检验准确度

right = 0;
for i = 1 : 50
        if c(i) == 1
                right = right + 1;
        end
end
for i = 51 : 100
        if c(i) == 2
                right = right + 1;
        end
end
for i = 101 : 150
        if c(i) == 3
                right = right + 1;
        end
end
right/150
```

这是算法2的实现.

```
%% 基本参数

```

```
data = xlsread('/Iris.csv');
x = data(:, 2)'; % 选取数据
n = size(x, 2); % 数据量
c = ones(1 ,n); % 分类，1表示setosa，2表示versicolor，3表示virginica
theta = [5.006 5.936 6.588 % mu
     0.349 0.511 0.629];  % sigma
k = 100; % 迭代次数

%% 正态分布的密度函数

f = @ (x, mu, sigma) 1/(sqrt(2*pi)*sigma)*exp(-(x-mu)^2/(2*sigma^2));

%% EM聚类

for l = 1 : k

    % 第一步
    % 根据第k-1次计算时的参数theta，对x(i)进行分类

    for i = 1 : n
        if f(x(i),theta(1,1),theta(2,1)) > ...
            f(x(i),theta(1,2),theta(2,2)) ...
                && f(x(i),theta(1,1),theta(2,1)) > ...
                    f(x(i),theta(1,3),theta(2,3))
            c(i) = 1;
        elseif f(x(i),theta(1,2),theta(2,2)) > ...
            f(x(i),theta(1,3),theta(2,3))
            c(i) = 2;
        else
            c(i) = 3;
        end
    end

    % 第二步
    % 根据第k次分类的结果计算参数theta

    for j = 1 : 3
```

```
            s1 = 0; % 第j类的样本的和
            s2 = 0; % 第j类的样本的平方和
            count = 1; % 第j类的样本的个数
            for i = 1 : n
                if c(i) == j
                    s1 = s1 + x(i);
                    s2 = s2 + x(i).^2;
                    count = count + 1;
                end
            end
            ave = s1/count; % 第j类的样本的均值
            var = 1/count*(s2 - count*ave^2); % 第j类的样本的方差
            theta(:,j) = [ave; var]; % 正态分布的极大似然估计
        end
end

%% 检验准确度

right = 0;
for i = 1 : 50
    if c(i) == 1
        right = right + 1;
    end
end
for i = 51 : 100
    if c(i) == 2
        right = right + 1;
    end
end
for i = 101 : 150
    if c(i) == 3
        right = right + 1;
    end
end
right/150
```