

改进自适应渐进 II 型删失下 Chen 分布的贝叶斯分析

张莉

西北师范大学数学与统计学院, 甘肃 兰州

收稿日期: 2023 年 12 月 20 日; 录用日期: 2023 年 12 月 29 日; 发布日期: 2024 年 1 月 31 日

摘要

本文基于改进的自适应渐进 II 型删失, 对 Chen 分布进行了贝叶斯分析。首先利用 EM 算法得到了参数的极大似然估计。针对共轭和非共轭的四种信息先验, 运用大方差和遗传算法确定了先验的超参数。进而根据 Metropolis-Hastings 算法实现了后验分布样本的抽取。最后, 通过真实数据集对不同先验下的贝叶斯估计性能进行比较并得出了相应的结论。

关键词

改进的自适应渐进 II 型删失, Chen 分布, EM 算法, 共轭和非共轭的信息先验, 大方差和遗传算法, Metropolis-Hastings 算法, 贝叶斯估计

Bayesian Analysis of Chen Distribution under Improved Adaptive Progressive Type-II Censoring

Li Zhang

College of Mathematics and Statistics, Northwest Normal University, Lanzhou Gansu

Received: Dec. 20th, 2023; accepted: Dec. 29th, 2023; published: Jan. 31st, 2024

Abstract

This paper presents a Bayesian analysis of the Chen distribution based on an improved adaptive progressive Type-II censoring. Initially, the EM algorithm is used to obtain the maximum likelihood estimation of the parameters. Four types of informative priors, including conjugate and non-conjugate, are then determined using large variance and genetic algorithms. Subsequently, the Metropolis-Hastings algorithm is employed to extract samples from the posterior distribution. Finally, a comparison of the Bayesian estimation performance under different priors is conducted using real datasets, leading to corresponding conclusions.

Keywords

Improved Adaptive Progressive Type-II Censoring, Chen Distribution, EM Algorithm, Conjugate and Non-Conjugate Informative Priors, Large Variance and Genetic Algorithm, Metropolis-Hastings Algorithm, Bayes Estimation

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在研究寿命数据时, 等待所有的试验单元达到寿命终点后再进行建模和相关分析往往是不现实的. 事实上, 常见的做法是在一定的时间段内或在特定事件发生之前停止观测, 从而产生删失寿命数据. 这种删失寿命数据可能会影响试验结果. 寿命数据可能受到不同类型删失的影响, 基于删失时间在观测寿命的右侧还是左侧, 可以将时间删失数据分为右删失或左删失. 在本文的讨论中仅关注右删失数据. 更详细的信息, 请参考 Klein & Moeschberger(1997) [1]. 展开对删失数据的研究对于研究人员来说非常重要. 在传统的 I 型 II 型删失下, Balakrishnan & Aggarwala(2000) [2] 引入了 II 型渐进删失方案. Ng et al.(2009) [3] 首次提出了自适应渐进 II 型 (AT-II PCS) 删失方案. Yan et al.(2021) [4] 扩展了自适应型 II 型渐进删失方案, 并提出了改进的自适应型 II 型渐进删失方案 (IAT-II PCS). 当然, 还存在许多其他改进的研究设计和方法. 在这些删失方案下, 许多学者对一些重要的寿命分布模型进行了研究, 可参考文献马金歌 (2021) [5], Ranjan et al.(2021) [6], Lv et

al.(2022) [7], Nassar & Elshahhat(2023) [8] 等.

近年来,许多学者讨论了在不同的删失方案下贝叶斯和非贝叶斯估计参数的方法. Balakrishnan & Mitra (2012) [9] 和 Shang & Ng(2020) [10], 主要关注一些非贝叶斯方法的发展, 特别是利用期望最大化 (EM) 算法进行最大似然估计. Okash et al.(2021) [11] 在自适应渐进 II 型删失方案下, 获得了 Burr-XII 分布参数和可靠性函数的 E-Bayesian 估计. Zhang et al.(2021) [12] 讨论了在广义 I 型混合删失下, 浴盆形寿命分布的贝叶斯和 E-Bayesian 估计. Algarni & Almarashi (2021) [13] 介绍了浴盆形寿命分布的 E-Bayesian 估计及其在可靠性特征上的应用. Shi & Gui(2023) [14] 利用经验贝叶斯和 E-Bayesian 估计, 在渐进 II 型删失下进行了浴盆形寿命分布的可靠性分析. Dey et al.(2023) [15] 利用渐进 II 型删失数据, 描述了伽马分布的贝叶斯参数估计方法.

Chen 分布由 Chen(2000) [16] 提出, 它的应用可以为产品设计、制造和维护提供重要参考, 对提高产品质量、可靠性和可维护性有积极影响. Chen 分布具有较高的灵活性、参数解释性和适用性, 特别适用于描述非负连续随机变量的分布, 在贝叶斯分析中具有一定的优势. 当需要进行贝叶斯分析时, 选择 Chen 分布可能是因为它能够更好地适应数据、提供直观的参数解释和适用于多种实际问题. Chen 分布的概率密度函数 (PDF) 和累积分布函数 (CDF) 如下:

$$f(x; \alpha, \beta) = \alpha\beta x^{\beta-1} \exp \left[\alpha(1 - e^{x^\beta}) + x^\beta \right] \quad x > 0, \alpha, \beta > 0, \quad (1)$$

$$F(x; \alpha, \beta) = 1 - \exp \left[\alpha(1 - e^{x^\beta}) \right] \quad x > 0, \alpha, \beta > 0. \quad (2)$$

可靠性函数和失效率可以表示为

$$S(x; \alpha, \beta) = \exp \left[\alpha(1 - e^{x^\beta}) \right] \quad x > 0, \alpha, \beta > 0,$$

$$h(x; \alpha, \beta) = \alpha\beta x^{\beta-1} e^{x^\beta} \quad x > 0, \alpha, \beta > 0,$$

特别地, 当 $\beta < 1$ 时, 失效率函数呈浴盆形状.

本文有两个主要目标. 首先考虑了 IAT-II PC 数据, 并考虑了在 IAT-II PC 下的 Chen 分布模型, 在解决参数的最大似然估计问题时, 选择了自适应步长 EM 算法. 其次对于贝叶斯分析, 本文提出了一个大方差和遗传算法进行四个先验分布的超参数的选取, 值得一提的是, 遗传算法能够广泛搜索超参数空间, 不局限于局部最优解, 并且对处理复杂的非线性问题具有良好的适应性和稳健性, 可以很容易地应用于更复杂的超参数优化问题.

2. 模型描述和相应的似然函数

基于 Yan et al.(2021) [4] 提出的 IAT-II 删失方案, 相应的似然函数可以表示为:

$$L = C \prod_{i=1}^{D_2} f(x_i) \prod_{i=1}^{D_1} [(1 - F(x_i))]^{R_i} (1 - F(x_B))^B,$$

这里 x_i 表示观测到的个体的寿命, R_i 是预先给定的删失方案, D_2, D_1, B, C 具体形式见下表 1

Table 1. Coefficient interpretation**表 1.** 系数解释

	D_2	D_1	B	C
Case1	m	m	0	$\prod_{i=1}^m \left(n - i + 1 - \sum_{s=1}^{i-1} R_s \right)$
Case2	m	k_1	$n - m - \sum_{i=1}^{k_1} R_i$	$\prod_{i=1}^m \left(n - i + 1 - \sum_{s=1}^{k_1} R_s \right)$
Case3	k_2	k_1	$n - k_2 - \sum_{i=1}^{k_1} R_i$	$\prod_{i=1}^m \left(n - i + 1 - \sum_{s=1}^{k_1} R_s \right)$

那么基于 Chen 分布的似然函数可以表示为:

$$L(\alpha, \beta) = C \left[\prod_{i=1}^{D_2} \alpha \beta x_i^{\beta-1} \exp \left(\alpha \left(1 - e^{x_i^\beta} \right) + x_i^\beta \right) \right] \times \left[\prod_{i=1}^{D_1} \left[\exp \left(\alpha \left(1 - e^{x_i^\beta} \right) \right) \right]^{R_i} \right] \left[\exp \left(\alpha \left(1 - e^{x_B^\beta} \right) \right) \right]^B. \quad (3)$$

使用 EM 算法得到参数 α 和 β 的最大似然估计 (MLE), 可以参考 Noori Asl(2017) [17] 文献中算法步骤. 本文引入了一种基于传统 EM 算法的自适应步长机制, 原理是根据参数更新状态动态调整步长, 旨在提高算法的收敛速度、稳定性和准确性, 使其更好地逼近最优解. 这使得能够在参数估计问题中获得更准确的结果. 具体的算法步骤如下所示.

Algorithm 1. EM 算法
Input:

观测数据 x
 隐变量 Z
 模型参数 Θ

Output:

步骤 1: 初始化参数. $\Theta^{(0)} = (\alpha^{(0)}, \beta^{(0)})$,
 步骤 2: E 步骤. 计算隐变量 Z 的后验概率期望, 即计算在给定观测数据 x 和当前参数 $\Theta^{(t)}$ 下, 隐变量 Z 的条件概率分布期望值, 表示为 $Q(\Theta, \Theta^{(t)})$.
 步骤 3: M 步骤. 利用 E 步骤计算得到的隐变量的后验概率期望, 对参数进行重新估计, 即调整参数使得似然函数最大化, 表示为 $\Theta^{(t+1)} = \operatorname{argmax}[Q(\Theta, \Theta^{(t)})]$.
 步骤 4: 自适应步长机制. 在第二次迭代之后, 根据前一次迭代的结果评估参数的更新状态. 如果观测到显著的更新步长减半; 如果观测到较小的更新步长加倍.
 步骤 5: 重复步骤 2、3、4. 直至参数收敛或达到预设的迭代次数.

接下来给出贝叶斯估计的先验和后验分布的形式.

3. 先验和后验分布

本节考虑了参数 α 和 β 的逆伽马、正态、对数正态和 Weibull 先验四种信息先验类型.

假设参数 α 和 β 是具有逆伽马信息先验的独立随机变量. 它们的先验分布可以分别表示为:

$$\pi_1(\alpha) = \frac{b_1^{a_1}}{\Gamma(a_1)} \alpha^{-(a_1+1)} \exp\left(-\frac{b_1}{\alpha}\right), \quad \alpha, a_1, b_1 > 0,$$

$$\pi_1(\beta) = \frac{b_2^{a_2}}{\Gamma(a_2)} \beta^{-(a_2+1)} \exp\left(-\frac{b_2}{\beta}\right), \quad \beta, a_2, b_2 > 0,$$

其中, a_1 和 a_2 是逆伽马先验的形状超参数, b_1 和 b_2 是尺度超参数.

因此, α 和 β 的联合先验分布可以表示为:

$$\pi_1(\alpha, \beta) \propto \alpha^{-(a_1+1)} \beta^{-(a_2+1)} \exp\left(-\frac{b_1}{\alpha}\right) \exp\left(-\frac{b_2}{\beta}\right).$$

同理假设参数 α 和 β 具有独立的正态先验分布, 它们各自的先验密度函数分别为

$$\pi_2(\alpha) = \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left(-\frac{(\alpha - u_1)^2}{2\sigma_1^2}\right), \quad \alpha, \sigma_1 > 0, -\infty < u_1 < +\infty,$$

$$\pi_2(\beta) = \frac{1}{\sigma_2 \sqrt{2\pi}} \exp\left(-\frac{(\beta - u_2)^2}{2\sigma_2^2}\right), \quad \beta, \sigma_2 > 0, -\infty < u_2 < +\infty.$$

得到的联合先验可以表示为:

$$\pi_2(\alpha, \beta) \propto \exp\left(-\frac{(\alpha - u_1)^2}{2\sigma_1^2}\right) \exp\left(-\frac{(\beta - u_2)^2}{2\sigma_2^2}\right). \quad (4)$$

类似地, 参数遵循对数正态先验的密度函数如下所示.

$$\pi_3(\alpha) = \frac{1}{\sigma_3 \sqrt{2\pi}} \frac{1}{\alpha} \exp\left(-\frac{1}{2} \frac{(\log \alpha - u_3)^2}{2\sigma_3^2}\right), \quad \alpha, \sigma_3 > 0, -\infty < u_3 < +\infty,$$

$$\pi_3(\beta) = \frac{1}{\sigma_4 \sqrt{2\pi}} \frac{1}{\beta} \exp\left(-\frac{1}{2} \frac{(\log \beta - u_4)^2}{2\sigma_4^2}\right), \quad \beta, \sigma_4 > 0, -\infty < u_4 < +\infty.$$

α 和 β 的联合先验由下式确定:

$$\pi_3(\alpha, \beta) \propto \frac{1}{\alpha} \frac{1}{\beta} \exp\left(-\frac{1}{2} \frac{(\log \alpha - u_3)^2}{2\sigma_3^2}\right) \exp\left(-\frac{1}{2} \frac{(\log \beta - u_4)^2}{2\sigma_4^2}\right). \quad (5)$$

最后, 参数遵循 Weibull 先验的密度函数的表达式如下所示. 其中, k_1 和 k_2 是 Weibull 先验的形状超参数, θ_1 和 θ_2 是尺度超参数.

$$\pi_4(\alpha) = k_1 \theta_1 \alpha^{k_1-1} \exp(-(\theta_1 \alpha)^{k_1}), \quad \alpha, k_1, \theta_1 > 0,$$

$$\pi_4(\beta) = k_2 \theta_2 \beta^{k_2-1} \exp(-(\theta_2 \beta)^{k_2}), \quad \beta, k_2, \theta_2 > 0.$$

联合先验如下:

$$\pi_4(\alpha, \beta) \propto \alpha^{k_1-1} \beta^{k_2-1} \exp(-(\theta_1 \alpha)^{k_1}) \exp(-(\theta_2 \beta)^{k_2}).$$

对于每种先验都有两个超参数. 本文通过设置较大的方差和遗传算法来获取超参数的最佳值. 首先使先验分布具有足够大的方差以适应模型参数. 然后使用遗传算法进一步优化参数结果. 实际上, 选择较大的方差有助于减少先验的影响, 从而使推断效率更高 (见 Natarajan & McCulloch(1998) [18]). 此外, 这种方法在缺乏明确信息时有优势. 遗传算法能够在超参数空间中进行全面的搜索, 可以更好地探索超参数空间中潜在的最优解, 从而有助于找到更好的超参数组合. 因此, 大方差和遗传算法的结合在确定超参数方面具有一定的优势, 特别是在处理复杂问题和全局优化时.

此外, 由于术语“足够大的值”似乎含糊不清, 人们更喜欢从专家那里获取一些指导来解决选择问题. 本文参考文献 Ranjan et al.(2021) [6], 设置形状和尺度参数的方差范围为 40000 和 50. 另一方面, (4)和(5)中给出的先验分布中, 超参数 $u_i, \sigma_i, (i = 1, 2, 3, 4)$ 来自正态分布和对数正态分布, 不满足上述形状和尺度参数的限定范围. 因此参数 u_i 设置在 (0, 10) 的范围内, σ_i^2 根据最大方差值 40000 进行取值. 大方差和遗传算法的步骤如下:

Algorithm 2. 大方差和遗传算法

Output:

步骤 1: 初始化超参数和遗传算法参数.

$$\eta(0) = (a_1(0), a_2(0), b_1(0), b_2(0), u_1(0), u_2(0), \sigma_1(0), \sigma_2(0), u_3(0), u_4(0), \sigma_3(0), \sigma_4(0), k_1(0), k_2(0), \theta_1(0), \theta_2(0)).$$

步骤 2: 定义评估度量函数 $G(z)$, 用于计算给定超参数组合的方差差异和变异系数差异.

- 基于 z 计算逆伽马分布、正态分布、对数正态分布和 Weibull 分布的参数.
- 从逆伽马分布、正态分布、对数正态分布和 Weibull 分布生成随机样本.
- 计算逆伽马分布、正态分布、对数正态分布和 Weibull 分布之间的方差差异.
- 将方差差异作为评估指标值返回.

步骤 3: 定义适应度函数 $H(z)$, 将方差差异转换为适应度值.

步骤 4: 定义超参数的范围, 包括逆伽马分布和 Weibull 分布的形状和尺度参数, 以及正态分布和对数正态分布的均值和标准差范围.

步骤 5: 定义遗传算法的参数 ϖ , 包括变量类型、适应度函数、超参数的上下界、种群大小、最大迭代次数、交叉和变异概率.

步骤 6: 运行遗传算法, 并输出最佳超参数、方差差异和变异系数差异结果.

现在, 定义 $\Theta = (\alpha, \beta)$. 相应的联合后验密度可以分别写为

$$\begin{aligned} \pi_1(\Theta | data) \propto & \alpha^{D_2-(a_1+1)} e^{-\frac{b_1}{\alpha}} \beta^{D_2-(a_2+1)} e^{-\frac{b_2}{\beta}} \exp \left[\alpha \left(\sum_{i=1}^{D_2} (1 - e^{x_i^\beta}) \right. \right. \\ & \left. \left. - \sum_{i=1}^{D_1} R_i (1 - e^{x_i^\beta}) - B (1 - e^{x_B^\beta}) \right) \right] \exp \left[\beta \left(\sum_{i=1}^{D_2} \log x_i \right) \right] \exp \left(\sum_{i=1}^{D_2} x_i^\beta \right), \end{aligned}$$

$$\pi_2(\Theta | data) \propto \alpha^{D_2} \exp\left(-\frac{(\alpha - u_1)^2}{2\sigma_1^2}\right) \beta^{D_2} \exp\left(-\frac{(\beta - u_2)^2}{2\sigma_2^2}\right) \exp\left[\alpha \left(\sum_{i=1}^{D_2} (1 - e^{x_i^\beta}) - \sum_{i=1}^{D_1} R_i (1 - e^{x_i^\beta}) - B(1 - e^{x_B^\beta})\right)\right] \exp\left[\beta \left(\sum_{i=1}^{D_2} \log x_i\right)\right] \exp\left(\sum_{i=1}^{D_2} x_i^\beta\right),$$

$$\begin{aligned} \pi_3(\Theta | data) &\propto \alpha^{D_2-1} \exp\left(-\frac{1}{2} \frac{(\log \alpha - u_3)^2}{2\sigma_3^2}\right) \beta^{D_2-1} \exp\left(-\frac{1}{2} \frac{(\log \beta - u_4)^2}{2\sigma_4^2}\right) \\ &\exp\left[\alpha \left(\sum_{i=1}^{D_2} (1 - e^{x_i^\beta}) - \sum_{i=1}^{D_1} R_i (1 - e^{x_i^\beta}) - B(1 - e^{x_B^\beta})\right)\right] \\ &\exp\left[\beta \left(\sum_{i=1}^{D_2} \log x_i\right)\right] \exp\left(\sum_{i=1}^{D_2} x_i^\beta\right), \end{aligned}$$

$$\begin{aligned} \pi_4(\Theta | data) &\propto \alpha^{D_2+k_1-1} \exp(-(\theta_1 \alpha)^{k_1}) \beta^{D_2+k_2-1} \exp(-(\theta_2 \beta)^{k_2}) \\ &\exp\left[\alpha \left(\sum_{i=1}^{D_2} (1 - e^{x_i^\beta}) - \sum_{i=1}^{D_1} R_i (1 - e^{x_i^\beta}) - B(1 - e^{x_B^\beta})\right)\right] \\ &\exp\left[\beta \left(\sum_{i=1}^{D_2} \log x_i\right)\right] \exp\left(\sum_{i=1}^{D_2} x_i^\beta\right). \end{aligned}$$

可以看到后验分布是复杂的, 无法直接获取样本. 因此, 采用 Metropolis-Hastings(M-H) 算法来解决这个问题. M-H 算法是一种用于从复杂概率分布中采样的马尔科夫链蒙特卡洛 (MCMC) 方法. 它可以用于估计参数的后验分布或计算模型的边际似然. Yan et al.(2021) [4] 将 M-H 算法与 Gibbs 采样结合使用. 然而, Gibbs 采样要求每个参数的条件分布可以很容易地进行采样, 这在某些情况下可能无法满足. 因此, M-H 算法更可取. 需要注意的是, $p(\Theta | \underline{x})$ 表示给定观测值 \underline{x} 下的向量参数 Θ 的联合后验分布. 基本算法可以定义如下.

Algorithm 3. Metropolis-Hastings 算法

Output:

步骤 1: 将迭代次数初始值设置为 $t = 1$, 初始化参数值 $\Theta(0)$, 并设置 $\Theta(t) = \Theta^{(0)}$.

步骤 2: 对于每个迭代 $t(t = t + 1)$, 重复以下步骤

a. 从提议分布 $Q(\Theta(t-1)|\Theta(t))$ 中生成一个提议参数值 Θ^* .

b. 计算接受概率 $\alpha(\Theta(t-1)|\Theta(t)) = \min\left(\frac{p(\Theta^*|\underline{x})}{p(\Theta(t-1)|\underline{x})}, 1\right)$.

c. 从均匀分布 $(0,1)$ 中生成一个随机数 u .

d. 如果 $u \leq \alpha(\Theta(t-1)|\Theta(t))$, 接受提议并设置 $\Theta^{(t)} = \Theta^*$. 否则, 拒绝提议并设置 $\Theta(t) = \Theta^{(t-1)}$.

步骤 3: 重复步骤 2 足够多次的迭代 ($t = N$), 以获得所需的样本.

3.1. 损失函数

在贝叶斯分析中, 损失函数的选择非常重要. 本文使用对称和非对称损失函数, 主要是均方误差损失函数 (SEL)、LINEX 损失函数 (LL) 和熵损失函数 (EL).

定义参数 Θ 的估计值为 $\hat{\Theta}$. 参数 Θ 在均方误差损失函数 (SEL)、非对称损失函数 LINEX 损失函数 (LL) 和熵损失函数 (EL) 下的定义如下:

$$L_{SEL}(\Theta, \hat{\Theta}) = (\Theta - \hat{\Theta})^2,$$

$$L_{LL}(\Theta, \hat{\Theta}) = e^{g(\hat{\Theta}-\Theta)} - g(\hat{\Theta} - \Theta) - 1, \quad g \neq 0,$$

$$L_{EL}(\Theta, \hat{\Theta}) = \left(\frac{\hat{\Theta}}{\Theta}\right)^q - q \ln \frac{\hat{\Theta}}{\Theta} - 1.$$

在三种损失函数下, 结合 M-H 算法通过计算后验均值可以获得参数 Θ 的贝叶斯估计值.

$$\hat{\Theta}_{SEL} = \frac{1}{N-M} \sum_{t=M+1}^N \Theta^{(t)},$$

$$\hat{\Theta}_{LL} = -\frac{1}{g} \ln \left(\frac{1}{N-M} \sum_{t=M+1}^N e^{g\Theta^{(t)}} \right), \quad g \neq 0,$$

$$\hat{\Theta}_{EL} = \left[\frac{1}{N-M} \sum_{t=M+1}^N (\Theta^{(t)})^{-q} \right]^{-\frac{1}{q}}, \quad q \neq 0,$$

其中 M 表示在燃烧阶段使用的迭代次数, g 和 q 是常数.

4. 实际数据分析

本节对 Hand & Ostrowski(1993) [19] 的真实数据集进行分析, 该数据集最初记录了 148 名肾移植患者的存活时间 (以月计). 完整的数据集如表 2 所示. Algarni et al.(2020) [20] 使用了相同的数据集, 采用了 I 型删失方案, 将伽马分布作为共轭先验分布, 在均方误差损失函数下获得了 E-Bayesian 估计量, 并提出该数据呈现出浴盆形的失效率.

数据集进行 KS 检验得到的拟合的结果见表 3, 拟合结果显示 Chen 分布的 K-S 距离较小, 而指数分布、Weibull 分布和 Kumaraswamy-指数分布的距离较大, 并且只有 Chen 分布的 P 值满足大于 0.5. 再次绘制数据集 2 的经验分布函数和 Q-Q 图, 如图 1 所示. 综上所述, Chen 分布适用于肾移植患者的数据集.

Table 2. Data of 148 kidney transplant patients
表 2. 148 名肾移植患者数据

0.035	0.068	0.100	0.101	0.167	0.168	0.190	0.213	0.233	0.234
0.508	0.508	0.533	0.633	0.767	0.768	0.770	1.066	1.267	1.300
1.600	1.639	1.803	1.867	2.180	2.667	2.967	3.328	3.393	3.700
3.803	4.311	4.867	5.180	6.233	6.367	6.600	6.600	7.180	7.667
7.733	7.800	7.933	7.967	8.016	8.300	8.410	8.607	8.667	8.800
9.100	9.233	10.541	10.607	10.633	10.667	10.869	11.067	11.180	11.443
12.213	12.508	12.533	13.467	13.800	14.267	14.475	14.500	15.213	15.333
15.525	15.533	15.541	15.934	16.200	16.300	16.344	16.600	16.700	16.933
17.033	17.067	17.475	17.667	17.700	17.967	18.115	18.115	18.933	18.934
19.508	19.574	19.733	20.148	20.180	20.900	21.167	21.233	21.600	22.100
22.148	22.180	22.180	22.267	22.300	22.500	22.533	22.867	23.738	24.082
24.180	24.705	25.213	25.705	29.705	30.443	31.667	31.934	32.180	32.367
32.672	32.705	33.148	33.567	33.770	33.869	34.836	34.869	34.934	35.738
36.180	36.213	39.410	39.433	39.672	40.001	41.733	41.734	42.311	42.869
43.180	43.279	43.902	44.267	44.475	44.900	45.148	46.451		

定义 K_1 为时间阈值 T_1 之前失效的单元数, 那么在 T_1 之后第一个失效的单元将是第 (K_1+1) 个失效的单元. K_2 ($K_1 \leq K_2 \leq m$) 则表示在时间阈值 T_2 之前观测到的失效单元数. 生成删失样本的删失方案如表 4 所示. 生成样本的算法可参考 Yan et al.(2021) [4]. 将 n 的值设置为 148, $\alpha = 0.043$ 和 $\beta = 0.386$ 作为参数初始值. 选择 m 的值为 100、74 和 37. 时间阈值 (T_1, T_2) 设置为 (4, 10) 和 (4, 30) 以生成具有 IAT-II PCS 的样本. 由大方差和遗传算法得到的先验超参数值见表 5.

根据上述描述, 图 2 和图 3 显示了参数 α 和 β 的 MLE 和贝叶斯估计的偏差 (Bias) 和均方误差 (MSE), 结果可以看出, 随着失效单元数 m 的减少, 均方误差增加; 而随着时间阈值 T_2 的增加, 均方误差减小. 此外, 四种先验下的贝叶斯估计性能都优于 MLE 估计; 在贝叶斯估计中, 对数正态先验下的估计性能相对更好; α 在平方误差损失函数下具有较小的均方误差, 而 β 在熵损失函数下具

有较小的偏差和均方误差; 当 $m = 74$ 和 $m = 37$ 时, 在删失方案 III 下表现优于删失方案 I 和 II. 图 4和图 5中的热图显示了不同时间阈值的均方误差对比, 可以对 MLE 和贝叶斯方法进行直观比较.

Table 3. K-S test results of the dataset

表 3. 数据集的 K-S 检验结果

分布	K-S 距离	P 值
Chen 分布	0.062083	0.6183
指数分布	0.59792	2.20E-16
Weibull 分布	0.84902	2.20E-16
Kumaraswamy-exponential 分布	0.63428	2.2E-16

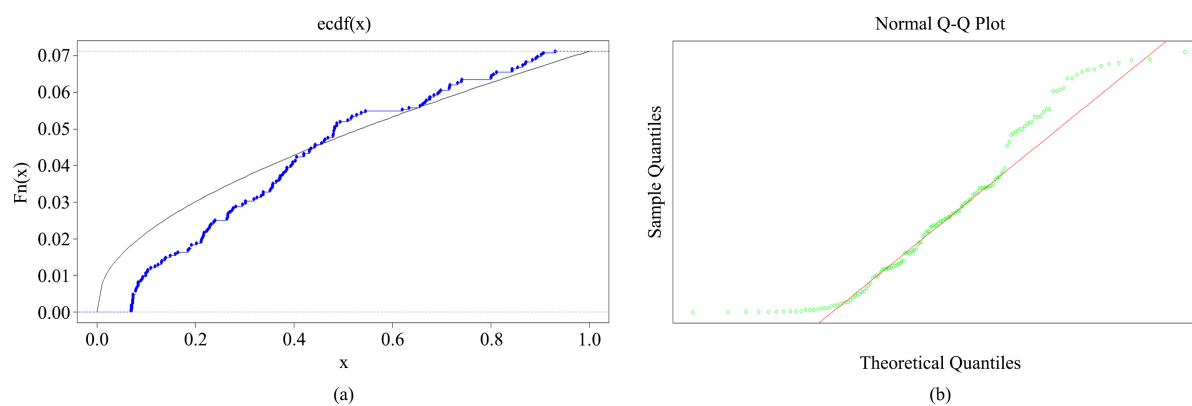


Figure 1. (a): Empirical distribution function of actual data, (b): Q-Q plot of actual data

图 1. (a) 实际数据经验分布函数, (b) 实际数据 Q-Q 图

Table 4. Deletion scheme**表 4.** 删失方案

m	删失方案	$(T_1, T_2)=(4,10)$ $(T_1, T_2)=(4,30)$	
		(K_1, K_2)	
100	I $R_m = n - m, R_i = 0$ for $i \neq m$	(31,52)	(31,100)
	II $R_1 = n - m, R_i = 0$ for $i \neq 1$	(2,10)	(2,67)
	III $R_{\frac{m+1}{2}} = n - m, R_i = 0$ for $i \neq \frac{m+1}{2}$; if m is odd, and $R_{\frac{m}{2}} = n - m, R_i = 0$ for $i \neq \frac{m}{2}$, if m is even	(31,52)	(31,74)
74	I $R_m = n - m, R_i = 0$ for $i \neq m$	(31,52)	(31,74)
	II $R_1 = n - m, R_i = 0$ for $i \neq 1$	(0,1)	(0,15)
	III $R_i = \frac{n-m}{m}, i = 1, 2, \dots, m$	(16,26)	(16,58)
37	I $R_m = n - m, R_i = 0$ for $i \neq m$	(31,37)	(31,37)
	II $R_1 = n - m, R_i = 0$ for $i \neq 1$	(0,1)	(0,4)
	III $R_i = \frac{n-m}{m}, i = 1, 2, \dots, m$	(8,13)	(8,29)

Table 5. Hyperparameter value**表 5.** 超参数值

先验分布	超参数
inverse Gamma	$a_1=1.00, b_1 = 1.02, a_2=1.00, b_2 = 1.01$
Normal	$u_1 = 2.79, \sigma_1 = 1.00, u_2 = 7.65, \sigma_2 = 1.00$
Log-normal	$u_3 = 3.35, \sigma_3 = 1.00, u_4 = 2.79, \sigma_4 = 1.00$
Weibull	$k_1 = 1.00, \theta_1 = 1.00, k_2 = 1.00, \theta_2 = 1.01$

(T_1, T_2)	m	α						β		
			I	II	III	I	II	III		
4, 10	100	MLE	Bias	0.11993	0.06340	0.06476	0.19100	0.13140	0.11408	
			MSE	0.01438	0.00402	0.00419	0.03648	0.01727	0.01301	
		IG_SE	Bias	0.05766	-0.02370	-0.02411	0.09456	0.05772	0.02660	
			MSE	0.00332	0.00056	0.00058	0.00894	0.00333	0.00342	
		IG_LL	Bias	0.05919	-0.02370	-0.02411	0.09596	0.05709	0.02050	
			MSE	0.00350	0.00056	0.00058	0.00921	0.00326	0.00042	
		IG_EL	Bias	0.05911	-0.02372	-0.02415	0.08646	0.05439	0.02013	
			MSE	0.00349	0.00056	0.00058	0.00747	0.00296	0.00041	
		N_SE	Bias	0.04472	0.05808	-0.02255	0.06746	0.04091	0.01803	
			MSE	0.00200	0.00337	0.00051	0.00455	0.00167	0.00033	
		N_LL	Bias	0.04645	0.05643	-0.02262	0.06900	0.04030	0.01801	
			MSE	0.00216	0.00318	0.00051	0.00476	0.00162	0.00032	
	N_EL	Bias	0.04653	-0.01952	-0.03670	0.06367	0.03783	0.01794		
		MSE	0.00217	0.00038	0.00135	0.00405	0.00143	0.00032		
	LN_SE	Bias	0.01455	-0.03477	-0.00876	-0.03646	-0.02989	-0.01891		
		MSE	0.00021	0.00121	0.00008	0.00133	0.00089	0.00036		
	LN_LL	Bias	0.01464	-0.03477	-0.00876	-0.03757	-0.02662	-0.01629		
		MSE	0.00021	0.00121	0.00008	0.00141	0.00071	0.00027		
	LN_EL	Bias	0.01464	-0.03537	-0.00912	-0.03621	-0.02607	-0.01586		
		MSE	0.00021	0.00125	0.00008	0.00131	0.00068	0.00025		
	W_SE	Bias	0.04774	-0.03621	-0.00894	0.06746	0.03252	0.03097		
		MSE	0.00228	0.00131	0.00008	0.00455	0.00106	0.00096		
	W_LL	Bias	0.04911	-0.03622	-0.00894	0.06646	0.03183	0.03077		
		MSE	0.00241	0.00131	0.00008	0.00442	0.00101	0.00095		
	W_EL	Bias	0.04917	-0.03681	-0.00925	0.06537	0.02883	0.03002		
		MSE	0.00242	0.00135	0.00009	0.00427	0.00083	0.00090		
	74	MLE	Bias	0.18661	0.16780	0.10828	0.21087	0.20094	0.20690	
			MSE	0.03482	0.02816	0.01172	0.04447	0.04037	0.04281	
		IG_SE	Bias	0.16936	0.15430	-0.05403	0.10646	0.09458	-0.08455	
			MSE	0.02868	0.02381	0.00292	0.01133	0.00895	0.00715	
		IG_LL	Bias	0.17243	0.15630	-0.05404	0.10642	0.09168	-0.05400	
			MSE	0.02973	0.02443	0.00292	0.01133	0.00840	0.00292	
		IG_EL	Bias	0.17280	0.15702	-0.05407	0.10631	0.08021	-0.04958	
			MSE	0.02986	0.02466	0.00292	0.01130	0.00643	0.00246	
		N_SE	Bias	-0.00614	-0.01956	0.01570	0.00646	0.00964	0.00378	
			MSE	0.00131	0.00028	0.00209	0.00748	0.00650	0.00407	
N_LL		Bias	0.05351	0.07187	0.04397	0.08644	0.07773	0.06240		
		MSE	0.00286	0.00517	0.00193	0.00747	0.00604	0.00389		
N_EL	Bias	0.05517	0.07404	-0.02211	0.08634	0.06746	0.05746			
	MSE	0.00304	0.00548	0.00049	0.00745	0.00455	0.00330			
LN_SE	Bias	-0.02295	-0.02656	-0.01548	0.04456	0.07097	0.03870			
	MSE	0.00053	0.00071	0.00024	0.00199	0.00504	0.00150			
LN_LL	Bias	0.02327	0.04474	-0.01549	0.04397	0.06913	0.03816			
	MSE	0.00054	0.00200	0.00024	0.00193	0.00478	0.00146			
LN_EL	Bias	0.02394	0.04513	-0.01587	0.04178	0.06249	0.03696			
	MSE	0.00057	0.00213	0.00025	0.00175	0.00390	0.00130			
W_SE	Bias	0.00526	0.10199	-0.01613	0.08832	0.09121	0.05769			
	MSE	0.00003	0.01040	0.00026	0.00967	0.00832	0.00333			
W_LL	Bias	0.15021	-0.01813	-0.01616	0.08201	0.07569	-0.07834			
	MSE	0.02256	0.00033	0.00026	0.00673	0.00573	0.00614			
W_EL	Bias	0.15673	0.10568	-0.01779	-0.23628	-0.02388	0.04624			
	MSE	0.02457	0.01117	0.00032	0.05583	0.00057	0.00214			
37	MLE	Bias	0.28750	0.27936	0.19257	0.27047	0.25025	0.21087		
		MSE	0.08266	0.07804	0.03708	0.07315	0.06263	0.04447		
	IG_SE	Bias	0.24940	-0.19347	-0.15704	0.15492	0.16814	0.11582		
		MSE	0.06220	0.03743	0.02466	0.02400	0.02827	0.01341		
	IG_LL	Bias	0.24956	-0.19357	-0.15706	0.15041	0.15380	0.10583		
		MSE	0.06228	0.03747	0.02467	0.02262	0.02365	0.01120		
	IG_EL	Bias	0.24958	-0.19360	-0.15837	0.13669	0.12097	0.08036		
		MSE	0.06229	0.03748	0.02508	0.01869	0.01463	0.00646		
	N_SE	Bias	0.01089	-0.18934	-0.15293	0.10373	-0.16786	-0.04582		
		MSE	0.00012	0.03585	0.02339	0.01076	0.02818	0.00210		
	N_LL	Bias	0.19996	-0.18935	-0.15294	-0.18070	-0.16280	-0.08154		
		MSE	0.03998	0.03585	0.02339	0.03265	0.02650	0.00665		
N_EL	Bias	0.21027	-0.18959	-0.15351	0.07746	-0.14672	-0.06220			
	MSE	0.04421	0.03594	0.02357	0.00600	0.02153	0.00387			
LN_SE	Bias	-0.05800	-0.16952	-0.14856	-0.20965	-0.35821	-0.05673			
	MSE	0.00336	0.02874	0.02207	0.08979	0.12832	0.00322			
LN_LL	Bias	0.08027	-0.16953	-0.14859	-0.11446	-0.11955	-0.05420			
	MSE	0.00644	0.02874	0.02208	0.01310	0.01429	0.00294			
LN_EL	Bias	0.08286	-0.17032	-0.14976	-0.10375	-0.09510	-0.05387			
	MSE	0.00687	0.02901	0.02243	0.01076	0.00904	0.00290			
W_SE	Bias	0.31003	-0.17146	-0.16252	-0.22986	-0.15756	-0.07811			
	MSE	0.09612	0.02940	0.02641	0.05284	0.02482	0.00610			
W_LL	Bias	0.31040	-0.17152	-0.16257	-0.11343	-0.15087	-0.06515			
	MSE	0.09640	0.02942	0.02643	0.01287	0.02276	0.00425			
W_EL	Bias	0.31058	-0.17671	-0.16580	-0.20433	-0.12142	-0.08327			
	MSE	0.09646	0.03122	0.02749	0.04175	0.01474	0.00693			

Figure 2. Bias and mean square error of Bayesian estimation using M-H algorithm at $T_1 = 4, T_2 = 10$

图 2. 利用 M-H 算法, 在 $T_1 = 4, T_2 = 10$ 下贝叶斯估计的偏差和均方误差

(T_1, T_2)	m		α			β				
			I	II	III	I	II	III		
100	MLE	Bias	0.04503	0.04796	0.05535	0.07180	0.05207	0.06217		
		MSE	0.00203	0.00230	0.00306	0.00516	0.00103	0.00047		
		IG_SE	Bias	0.03308	-0.01557	-0.01862	0.04335	0.04348	0.01027	
			MSE	0.00189	0.00024	0.00035	0.00206	0.00189	0.00011	
		IG_LL	Bias	0.03322	-0.01557	-0.01862	0.04563	0.04558	0.01026	
			MSE	0.00110	0.00024	0.00035	0.00208	0.00208	0.00011	
	IG_EL	Bias	0.03323	-0.01593	-0.01907	0.04399	0.04662	0.01023		
		MSE	0.00110	0.00025	0.00036	0.00193	0.00217	0.00010		
	N_SE	Bias	0.02258	-0.03951	-0.01810	0.03878	0.02579	0.01012		
		MSE	0.00051	0.00156	0.00033	0.00150	0.00067	0.00010		
	N_LL	Bias	0.02279	-0.03951	-0.01814	0.03850	0.02759	0.01011		
		MSE	0.00052	0.00156	0.00033	0.00148	0.00076	0.00010		
	N_EL	Bias	0.02280	-0.04028	-0.02784	0.03235	0.02677	0.01008		
		MSE	0.00052	0.00162	0.00078	0.00105	0.00072	0.00010		
	LN_SE	Bias	0.00539	-0.00382	-0.00499	0.01678	-0.01358	0.00064		
		MSE	0.00003	0.00001	0.00002	0.00028	0.00018	0.00000		
	LN_LL	Bias	0.00543	-0.00382	-0.00499	0.01739	-0.01678	0.00064		
		MSE	0.00003	0.00001	0.00002	0.00030	0.00028	0.00000		
	LN_EL	Bias	0.00543	-0.00383	-0.00507	0.01535	-0.01888	0.00061		
		MSE	0.00003	0.00001	0.00003	0.00024	0.00036	0.00000		
	W_SE	Bias	0.00567	-0.00175	0.00582	0.04535	0.02756	0.02171		
		MSE	0.00003	0.00000	0.00003	0.00206	0.00076	0.00047		
	W_LL	Bias	0.04386	0.02914	0.00582	0.04568	0.03208	0.02161		
		MSE	0.00192	0.00085	0.00063	0.00209	0.00103	0.00047		
	W_EL	Bias	0.04380	0.02836	0.00552	0.04463	0.03107	0.02125		
		MSE	0.00193	0.00080	0.00003	0.00199	0.00102	0.00045		
	4,30	MLE	Bias	0.12981	0.12828	0.09343	0.17113	0.18655	0.17113	
			MSE	0.01685	0.01646	0.00873	0.02929	0.03480	0.02929	
			IG_SE	Bias	0.12379	0.12436	0.02816	0.08963	0.08824	-0.04631
				MSE	0.01532	0.01547	0.00079	0.00803	0.00779	0.00214
			IG_LL	Bias	0.12559	0.12481	0.03021	0.08962	0.08668	-0.05075
				MSE	0.01577	0.01558	0.00091	0.00803	0.00751	0.00258
		IG_EL	Bias	0.12576	0.12483	0.03028	0.08959	0.08029	-0.08071	
			MSE	0.01581	0.01558	0.00092	0.00803	0.00645	0.00651	
		N_SE	Bias	-0.01592	-0.17205	-0.02147	0.03988	0.06291	0.04631	
			MSE	0.00025	0.02060	0.00046	0.00159	0.00396	0.00214	
N_LL		Bias	0.04858	0.04435	0.03120	0.03877	0.06155	0.04547		
		MSE	0.00236	0.00197	0.00097	0.00150	0.00379	0.00207		
N_EL		Bias	0.04883	0.04569	0.03213	0.03743	0.05672	0.04234		
		MSE	0.00238	0.00209	0.00103	0.00140	0.00322	0.00179		
LN_SE		Bias	-0.02901	0.01724	-0.01688	0.04085	0.06268	0.01713		
		MSE	0.00040	0.00030	0.00028	0.00167	0.00393	0.00029		
LN_LL		Bias	0.02226	0.03811	0.00978	0.04036	0.06163	0.01697		
		MSE	0.00050	0.00145	0.00010	0.00163	0.00580	0.00029		
LN_EL		Bias	0.02290	0.03908	0.01008	0.03853	0.05789	0.01633		
		MSE	0.00052	0.00153	0.00010	0.00148	0.00335	0.00027		
W_SE		Bias	0.04304	0.02916	0.01120	0.09312	0.05331	-0.04328		
		MSE	0.00185	0.00085	0.00013	0.00867	0.00284	0.00187		
W_LL		Bias	0.10539	0.07793	0.01254	0.07743	0.04137	-0.05282		
		MSE	0.01111	0.00607	0.00016	0.00600	0.00171	0.00279		
W_EL		Bias	0.10907	0.08044	0.01257	-0.04668	-0.06474	-0.16960		
		MSE	0.01190	0.00647	0.00016	0.00218	0.00419	0.02876		
37		MLE	Bias	0.21502	0.21288	0.16173	0.25060	0.23074	0.17113	
			MSE	0.04623	0.04532	0.02616	0.06280	0.05324	0.02929	
			IG_SE	Bias	0.20944	-0.18748	-0.14114	0.13127	0.11747	0.00641
				MSE	0.04387	0.03515	0.01992	0.01723	0.01380	0.00441
			IG_LL	Bias	0.20961	-0.18751	-0.14857	0.12760	0.10751	0.06019
				MSE	0.04394	0.03516	0.02207	0.01628	0.01156	0.00362
		IG_EL	Bias	0.20964	-0.19880	-0.14635	0.11555	0.08237	0.04289	
			MSE	0.04395	0.03952	0.02142	0.01335	0.00679	0.00184	
		N_SE	Bias	-0.04121	-0.02594	0.14627	0.09426	-0.15764	0.06541	
			MSE	0.00170	0.00067	0.02140	0.00889	0.02485	0.00428	
	N_LL	Bias	0.17763	-0.17207	0.14675	0.06831	-0.16247	0.06019		
		MSE	0.03155	0.02961	0.02154	0.00467	0.02640	0.00362		
	N_EL	Bias	0.18628	-0.17329	0.14656	-0.20304	-0.27625	0.04289		
		MSE	0.03470	0.03003	0.02148	0.04123	0.07631	0.00184		
	LN_SE	Bias	-0.01281	-0.15849	0.12582	-0.06659	-0.08505	-0.04060		
		MSE	0.00016	0.02512	0.01583	0.00443	0.00734	0.00165		
	LN_LL	Bias	0.07594	-0.15850	0.12679	-0.08121	-0.11059	-0.07192		
		MSE	0.00577	0.02512	0.01608	0.00659	0.01223	0.00517		
	LN_EL	Bias	0.07832	-0.15883	0.12725	-0.32605	-0.40501	-0.03713		
		MSE	0.00613	0.02523	0.01619	0.10631	0.16403	0.00138		
	W_SE	Bias	0.29132	-0.16264	0.15362	-0.10806	-0.12023	-0.06190		
		MSE	0.08487	0.02645	0.02360	0.01168	0.01446	0.00383		
	W_LL	Bias	0.29178	-0.16270	0.15473	-0.11238	-0.12043	-0.06395		
		MSE	0.08513	0.02647	0.02394	0.01263	0.01450	0.00409		
	W_EL	Bias	0.29186	-0.16593	0.15369	-0.10726	-0.11597	-0.06179		
		MSE	0.08518	0.02753	0.02362	0.01150	0.01345	0.00382		

Figure 3. Bias and mean square error of Bayesian estimation using M-H algorithm at $T_1 = 4, T_2 = 30$

图 3. 利用 M-H 算法, 在 $T_1 = 4, T_2 = 30$ 下贝叶斯估计的偏差和均方误差

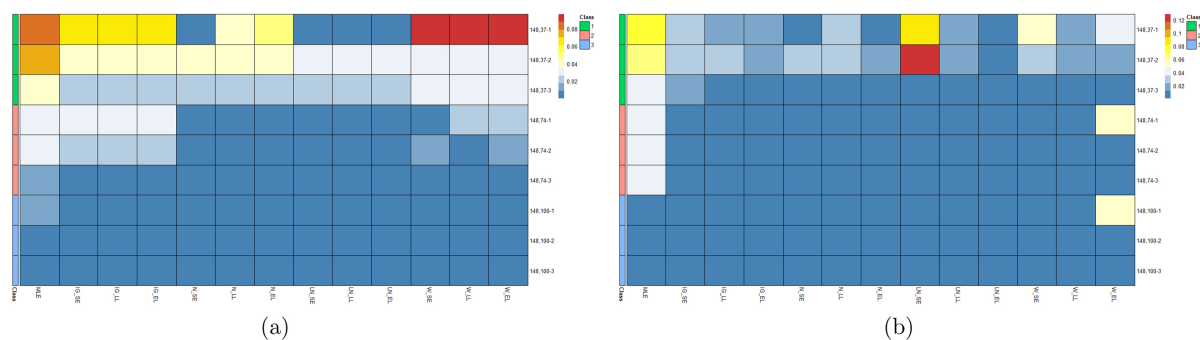


Figure 4. (a): Heat map of α , (b): Heat map of β at $(T_1, T_2) = (4, 10)$

图 4. (a) α , (b) β 在 $(T_1, T_2) = (4, 10)$ 下的热图

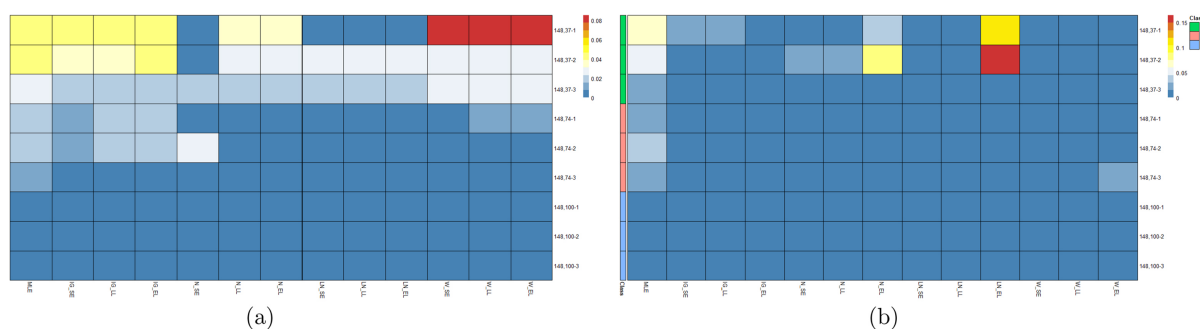


Figure 5. (a): Heat map of α , (b): Heat map of β at $(T_1, T_2) = (4, 30)$

图 5. (a) α , (b) β 在 $(T_1, T_2) = (4, 30)$ 下的热图

参考文献

- [1] Klein, J.P. and Moeschberger, M.L. (1997) *Survival Analysis: Techniques for Censored and Truncated Data*. Springer, New York.
- [2] Balakrishnan, N. and Aggarwala, R. (2000) *Progressive Censoring: Theory, Methods, and Applications*. Birkhäuser Boston, MA. <https://doi.org/10.1007/978-1-4612-1334-5>
- [3] Ng, H.K.T., Kundu, D. and Chan, P.S. (2009) Statistical Analysis of Exponential Lifetimes under an Adaptive Type-II Progressive Censoring Scheme. *Naval Research Logistics*, **56**, 687-698. <https://doi.org/10.1002/nav.20371>
- [4] Yan, W., Li, P. and Yu, Y. (2021) Statistical Inference for the Reliability of Burr-Xii Distribution under Improved Adaptive Type-II Progressive Censoring. *Applied Mathematical Modelling*, **95**, 38-52. <https://doi.org/10.1016/j.apm.2021.01.050>
- [5] 马金歌. 基于截尾样本的可靠性统计推断研究 [D]: [硕士学位论文]. 西安: 西安电子科技大学, 2021.

- [6] Ranjan, R., Sen, R. and Upadhyay, S.K. (2021) Bayes Analysis of Some Important Lifetime Models Using MCMC Based Approaches When the Observations Are Left Truncated and Right Censored. *Reliability Engineering and System Safety*, **214**, Article 107747. <https://doi.org/10.1016/j.ress.2021.107747>
- [7] Lv, Q., Tian, Y. and Gui, W. (2022) Statistical Inference for Gompertz Distribution under Adaptive Type-II Progressive Hybrid Censoring. *Journal of Applied Statistics*, 1-30. <https://doi.org/10.1080/02664763.2022.2136147>
- [8] Nassar, M. and Elshahhat, A. (2023) Estimation Procedures and Optimal Censoring Schemes for an Improved Adaptive Progressively Type-II Censored Weibull Distribution. *Journal of Applied Statistics*, 1-25. <https://doi.org/10.1080/02664763.2023.2230536>
- [9] Balakrishnan, N. and Mitra, D. (2012) Left Truncated and Right Censored Weibull Data and Likelihood Inference with an Illustration. *Computational Statistics and Data Analysis*, **56**, 4011-4025. <https://doi.org/10.1016/j.csda.2012.05.004>
- [10] Shang, X. and Ng, H.K.T. (2020) On Parameter Estimation for the Generalized Gamma Distribution Based on Left-Truncated and Right-Censored Data. *Computational and Mathematical Methods*, **3**, e1091. <https://doi.org/10.1002/cmm4.1091>
- [11] Okasha, H., Nassar, M. and Dobbah, S.A. (2021) E-Bayesian Estimation of Burr Type-XII Model Based on Adaptive Type-II Progressive Hybrid Censored Data. *AIMS Mathematics*, **6**, 4173-4196. <https://doi.org/10.3934/math.2021247>
- [12] Zhang, Y., Liu, K. and Gui, W. (2021) Bayesian and E-Bayesian Estimations of Bathtub-Shaped Distribution under Generalized Type-I Hybrid Censoring. *Entropy*, **23**, Article 934. <https://doi.org/10.3390/e23080934>
- [13] Algarni, A. and Almarashi, A.M. (2021) E-Bayesian Estimation of Reliability Characteristics of Two-Parameter Bathtub-Shaped Lifetime Distribution with Application. *Quality and Reliability Engineering International*, **37**, 1635-1649. <https://doi.org/10.1002/qre.2817>
- [14] Shi, W. and Gui, W. (2023) Reliability Analysis of Bathtub-Shaped Distribution Using Empirical Bayesian and E-Bayesian Estimations under Progressive Type-II Censoring. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*.
- [15] Dey, S., Elshahhat, A. and Nassar, M. (2023) Analysis of Progressive Type-II Censored Gamma Distribution. *Computational Statistics*, **38**, 481-508. <https://doi.org/10.1007/s00180-022-01239-y>
- [16] Chen, Z. (2000) A New Two-Parameter Lifetime Distribution with Bathtub Shape or Increasing Failure Rate Function. *Statistics and Probability Letters*, **49**, 155-161. [https://doi.org/10.1016/S0167-7152\(00\)00044-4](https://doi.org/10.1016/S0167-7152(00)00044-4)
- [17] Noori Asl, M., Belaghi, R.A. and Bevrani, H. (2017) On Burr-Xii Distribution Analysis under Progressive Type-II Hybrid Censored Data. *Methodology and Computing in Applied Probability*, **19**, 665-683. <https://doi.org/10.1007/s11009-016-9514-7>

-
- [18] Natarajan, R. and McCulloch, C.E. (1998) Gibbs Sampling with Diffuse Proper Priors: A Valid Approach to Data-Driven Inference? *Journal of Computational and Graphical Statistics*, **7**, 267-277. <https://doi.org/10.1080/10618600.1998.10474776>
- [19] Hand, D., Daly, F., McConway, K., Lunn, D. and Ostrowski, E. (1993) *A Handbook of Small Data Sets*. Chapman and Hall/CRC, London.
- [20] Algarni, A., Almarashi, A. M., Okasha, H. and Ng, H.K.T. (2020) E-Bayesian Estimation of Chen Distribution Based on Type-I Censoring Scheme. *Entropy*, **22**, Article 636. <https://doi.org/10.3390/e22060636>