软件可靠性工程基础

——软件可靠性评估

北京航空航天大学可靠性与系统工程学院

资料整理
火龙果软件
荐

目的

- 理解什么是软件可靠性评估，基本原理
- 掌握和应用典型的软件可靠性模型
- 了解比较软件可靠性模型的主要方法
- 理解稳定使用阶段的软件可靠性评估
- 理解增长测试中的软件可靠性评估

资料整理
火龙果软件
荐

内容

1. 软件可靠性评估概述
2. 软件可靠性评估模型
3. 模型预计的有效性
4. 稳定使用阶段的软件可靠性评估
5. 增长测试中的软件可靠性评估

资料整理
火龙果软件
荐

1 软件可靠性评估概述

资料整理
火龙果软件
荐

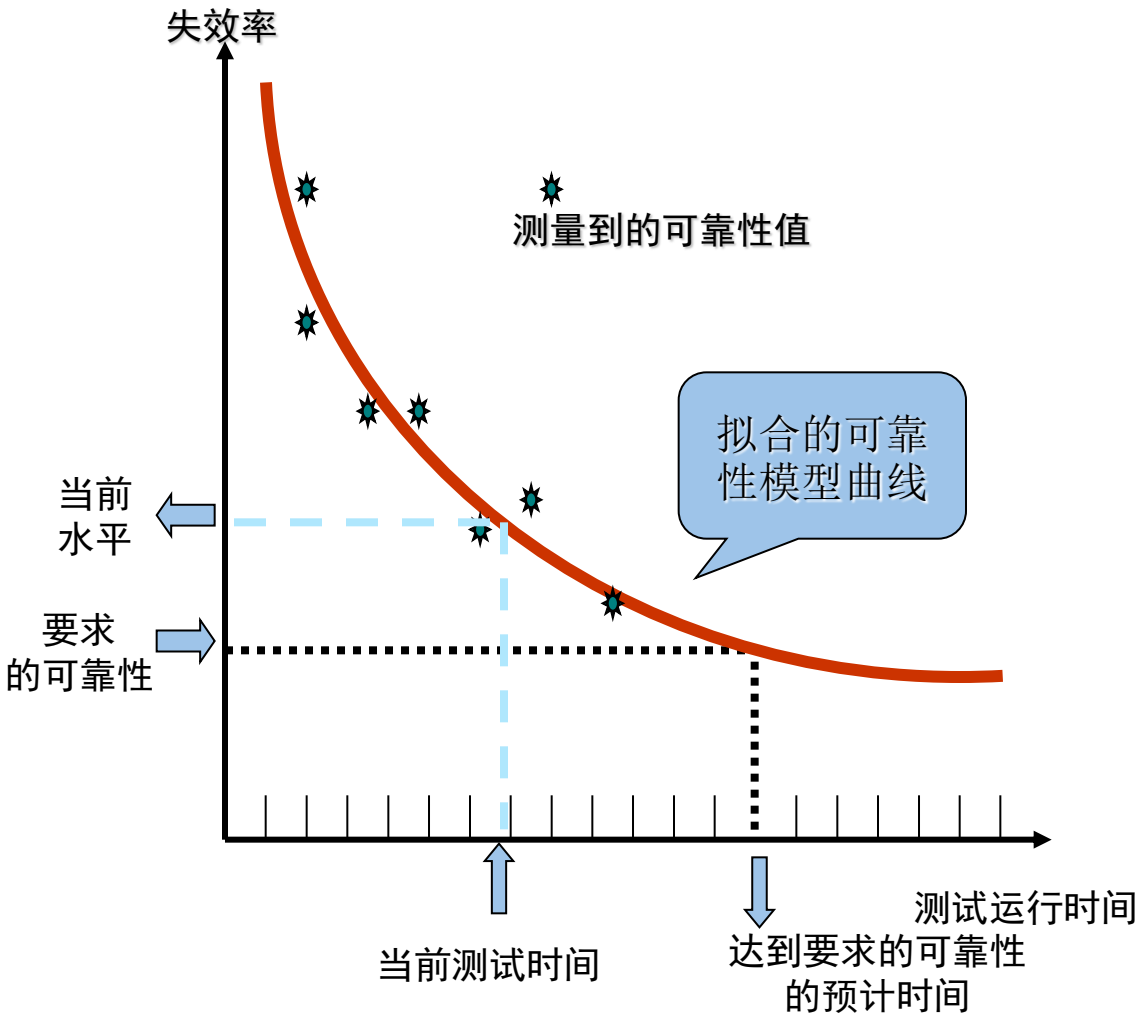
何为软件可靠性评估

- 国家标准GB11457 （软件工程术语）
  - 软件可靠性评估（software reliability assessment）或软件可靠性评价（software reliability evaluation）是指“确定现有系统或系统部件可靠性所达到的水平的过程”
- 国际标准IEEE Std.1633 （IEEE Recommended Practice on Software Reliability）
  - 软件可靠性评估（software reliability evaluation）被定义为“统计学技术在系统测试和运行期间收集的可观测失效数据上的应用，用于评价软件的可靠性”

因此，二者的定义均认为，软件可靠性评估是在获得了软件的失效数据之后对软件可靠性水平的定量估计和评价。

资料整理
火龙果软件
荐

评估的原理



评估的准确与否取决于什么？

资料整理 火龙果软件 荐

2 软件可靠性评估模型

资料整理
火龙果软件
荐

模型概述

- 基于模型的软件可靠性评估基本原理是：对软件可靠性测试中或实际使用中收集的失效数据，利用统计知识分析其规律，建立一个参数模型，在软件可靠性数据的基础上对该统计分布的参数进行估计，从而在此模型基础上对软件的可靠性进行定量估计或评价。

资料整理
火龙果软件
荐

呈指数分布的NHPP模型

———Jelinski-Moranda模型

资料整理
火龙果软件
荐

Jelinski-Moranda模型——概述

- 模型概述
  - 该模型是由Jelinski、Moranda于1972年开发的可靠性模型，是最早建立的软件可靠性模型之一，曾用于麦克唐奈道格拉斯海军工程中。
  - 该模型以一种简便和合乎直觉的方式表明如何根据软件缺陷的显露历程来预计未来软件可靠性的行为，它包含软件可靠性建模中若干典型和最主要的假设，因而引发出后来的许多种变形。
  - 事实上，现有大多数软件可靠性模型要么可认为是其变形或扩展，要么与其密切相关。该模型对软件可靠性定量分析技术的建立和发展做出了重要的贡献，是软件可靠性研究领域的第一个里程碑。

资料整理
火龙果软件
荐

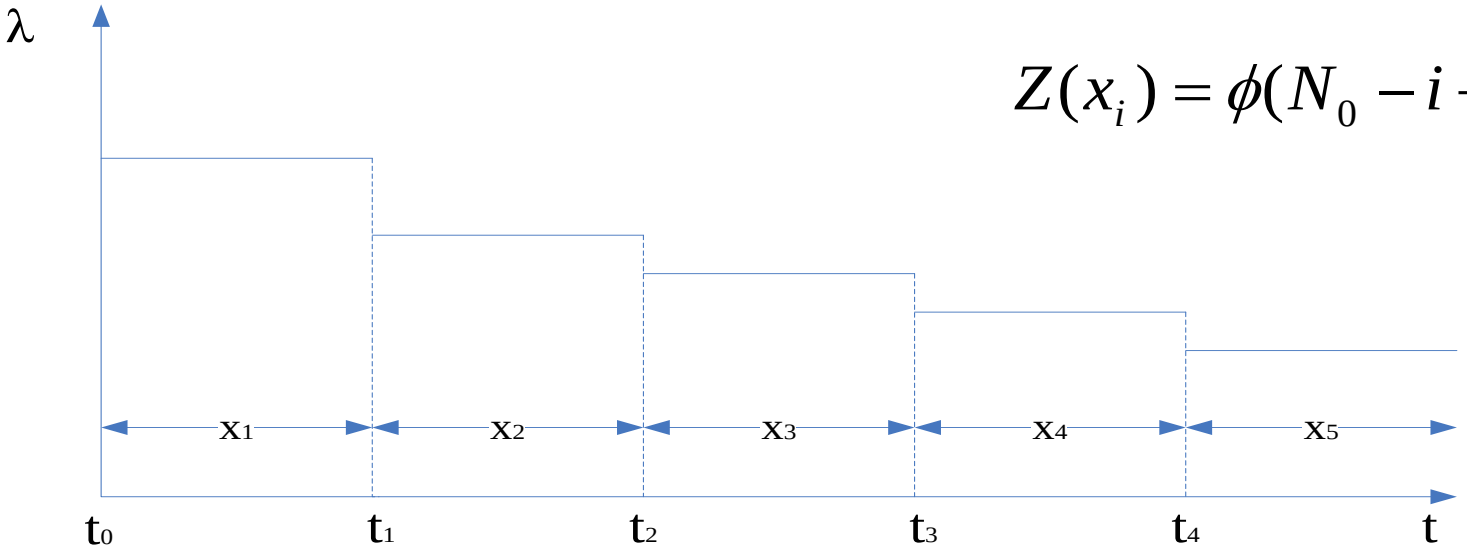
Jelinski-Moranda模型-假设与数据要求

- 假设与数据要求
  - 程序中的固有错误数 N_0 是一个未知的常数；
  - 程序中的各个错误是相互独立的，每个错误导致系统发生失效的可能性大致相同，各次失效间隔时间也相互独立；
  - 测试过程中检测到的错误，都被排除，每次排错只排除一个错误，排错时间可以忽略不计，在排错过程中不引入新的错误；
  - 程序的失效率在每个失效间隔时间内是常数，其数值正比于程序中残留的错误数，在第i个测试区间，其失效率函数为

$$Z(x_i) = \phi(N_0 - i + 1)$$

资料整理
火龙果软件
荐

Jelinski-Moranda模型-假设与数据要求



$$Z(x_i) = \phi(N_0 - i + 1)$$

- 图 7-1 失效率变化曲线

- 错误以相等的可能发生，且相互独立，错误检测率正比于当前程序中的错误数；
- 软件的运行方式与预期的运行方式相似。

资料整理 火龙果软件 荐

Jelinski-Moranda模型-构造与参数估计

- 模型的构造与参数估计
  - 在假设的基础上，运用可靠性工程学的基本理论，以第i—1次失效为起点的第i次失效发生的时间X_i是一个随机变量，它服从以$Z(x_i)$为参数的指数分布，其密度函数为

$$f(x_i)=\phi(N_0-i+1)\exp\{-\phi(N_0-i+1)x_i\}$$

  - 其分布函数为

$$F(x_i)=\int_0^{x_i} f(x_i)dx_i=1-\exp\{-\phi(N_0-i+1)x_i\}$$

  - 其可靠性函数为

$$R(x_i)=1-F(x_i)=\exp\{-\phi(N_0-i+1)x_i\}$$

资料整理
火龙果软件
荐

Jelinski-Moranda模型-构造与参数估计

- 假设总共发生n个失效，似然函数为

$$L(x_1,\cdots,x_n)=\prod_{i=1}^{n} f(x_i)=\prod_{i=1}^{n}\phi(N_0-i+1)\exp\{-\phi(N_0-i+1)x_i\}$$

- 对上式两边取对数，得

$$\ln L(x_1,\cdots,x_n)=\sum_{i=1}^{n}\ln f(x_i)=\sum_{i=1}^{n}(\ln\phi(N_0-i+1)-\phi(N_0-i+1)x_i)$$

- 则模型参数的极大似然法估计值是以下方程组的解

$$\begin{cases}\hat{\phi}=\dfrac{n}{\hat{N}(\sum_{i=1}^{n}x_i)-\sum_{i=1}^{n}(i-1)x_i}\\ \sum_{i=1}^{n}\dfrac{1}{\hat{N}-(i-1)}=\dfrac{n}{\hat{N}-(1/\sum_{i=1}^{n}x_i)(\sum_{i=1}^{n}(i-1)x_i)}\end{cases}$$

资料整理
火龙果软件
荐

Jelinski-Moranda模型

- 可靠性预计

$$R_{n+1}(x)=R(x\mid t_n)=e^{-(\hat{N}-n)\hat{\phi}x}$$

$$F_{n+1}(x)=1-R_{n+1}(x)=1-e^{-(\hat{N}-n)\hat{\phi}x}$$

$$f_{n+1}(x)=-\frac{dR_{n+1}(x)}{dx}=(\hat{N}-n)\hat{\phi}e^{-(\hat{N}-n)\hat{\phi}x}$$

$$\hat{T}_{TF_{n+1}}=1/\hat{\phi}(\hat{N}-n)$$

$$T_{TF_{n+1}}=E\{X_{n+1}\mid x_1,\cdots,x_n\}=\int_0^{\infty}R_{n+1}(x)dx=\frac{1}{\Phi(N_0-n)}$$

资料整理
火龙果软件
荐

应用案例

- 1972年，Jelinski和Moranda分析U.S. Navy Fleet Computer Programming Center的数据。这些数据是从美国海军战术数据系统（Naval Tactical Data System-NTDS)保存的一个实时多处理机系统软件开发过程的资料中提取的，包括了38个不同的模块，每个程序模块都经过了开发阶段、测试阶段和使用阶段。

资料整理
火龙果软件
荐

应用案例

- 每当操作者或用户发现了系统的错误征兆，就用软件异常报告的形式记载和反映，再由熟悉整个程序的人员跟踪这些问题，查找原因。 这些模块中的一个大型模块的失效间隔时间（用日表示)示于下表。
- 这个模块在开发阶段，发现了26个错误，在测试阶段又发现了5个错误，在使用阶段发现了1个错误后，对模块再次进行了测试，结果又发现了2个错误。

资料整理
火龙果软件
荐

失效数据

错误数（k）	实际失效时间（日）	错误数（k）	实际失效时间（日）	错误数（k）	实际失效时间（日）
1	9	18	3	28	47
2	12	19	6	29	12
3	11	20	1	30	9
4	4	21	11	31	135
5	7	22	33	32	258
6	2	23	7	33	16
……	……	24	91	34	35
15	4	25	2		
16	1	26	1		
17	3	27	87		

测试发现

使用发现

再测发现

开发发现

资料整理 火龙果软件 荐

利用J-M模型评估案例软件的可靠性

- 目的：
  - 评估开发结束时软件的可靠性水平MTBF或失效率
- 过程：
  - 根据失效数据评估模型参数 N_0 和 Φ
  - 根据评估公式计算开发结束时软件的可靠性水平

资料整理
火龙果软件
荐

基本公式

- 失效率函数为： $Z(x_i)=\Phi(N_0-i+1)$
- 其分布函数为： $F(x_i)=1-\exp\{-\Phi(N_0-i+1)x_i\}$
- 其可靠性函数为： $R(x_i)=\exp\{-\Phi(N_0-i+1)x_i\}$
- 失效时间间隔为： $MTBF_i=\dfrac{1}{\Phi(N_0-i+1)}$

资料整理
火龙果软件
荐

模型参数的求解

$$\begin{cases} \sum_{i=1}^{n} \dfrac{1}{N_0 - i + 1} = \dfrac{n}{N_0 - \dfrac{1}{t}\sum_{i=1}^{n}(i-1)x_i} \\ \Phi = \dfrac{n}{N_0 t - \sum_{i=1}^{n}(i-1)x_i} \end{cases}$$

其中 $$t = \sum_{i=1}^{n} x_i$$

利用数值计算法可以得到 N_0 和 Φ 的点估计值。

资料整理
火龙果软件
荐

模型参数和可靠性的估计值

- Jelinski和Moranda得到的参数估计值为

$$\hat{\Phi} = 0.00685 \qquad \hat{N}_0 = 31.2$$

- 失效间隔时间的估计值则可由下式算出：

$$MTB\hat{F}_i = \frac{1}{0.00685(32 - i + 1)}$$

- i=27时，MTBF=24.3309(日)

资料整理
火龙果软件
荐

应用案例结果分析

- 程序在使用和再测试中发现的错误数共有34个，假定程序的排错工作已经很彻底，可认为程序的固有错误总数为34个，估计值32略低于实际值。

,

资料整理
火龙果软件
荐

非指数分布的NHPP模型

———Musa-Okumoto（M-O）对数泊松执行时间模型

资料整理
火龙果软件
荐

Musa-Okumoto（M-O）模型-概述

- 模型概述
  - 对数泊松模型是另一个被广泛使用的模型，它是由Musa和Okumoto提出的。
  - 该模型是失效强度函数随失效发生而指数递减的非均匀泊松过程。
  - 指数率递减反映了以下观点：早期发现的缺陷比晚期发现的缺陷对失效强度函数的减小作用大。
  - 之所以称之为对数泊松模型是因为期望的失效数是时间的对数函数。

资料整理
火龙果软件
荐

Musa-Okumoto（M-O）模型-假设与数据要求

- 假设与数据要求
  - 到时刻t的累积失效数M(t)符合泊松过程；
  - 失效强度随着失效期望数的增加而呈现指数递减，即 $\lambda(t)=\lambda_0 e^{-\theta\mu(t)}$ 其中 $\mu(t)$ 为均值函数，$\theta>0$ 是失效强度递减参量，且 $\lambda_0>0$是初始失效强度；
  - 软件的运行方式与预期的运用方式相似；
  - 每个错误发生的机会相同，且严重等级相同；
  - 失效之间相互独立；
  - 数据要求为：完全失效数据，即失效间隔时间或累计失效时间。

资料整理
火龙果软件
荐

Musa-Okumoto（M-O）模型-模型构造与参数估计

- 模型的构造与参数估计

$$\mu(t)=[\ln(\lambda_0\theta\cdot t+1)]/\theta \qquad \lambda(t)=\mu'(t)=\lambda_0/(\lambda_0\theta\cdot t+1)$$

- M-O模型还可以用另一个表达式来给出：
- 令 $\beta_0=\theta^{-1}$ 且 $\beta_1=\lambda_0\theta$

$$\mu(t)=\beta_0\ln(\beta_1 t+1) \qquad \lambda(t)=\beta_0\beta_1/(\beta_1 t+1)$$

- 用极大似然估计法：

$$\hat{\beta}_0=\frac{n}{\ln(1+\hat{\beta}_1 t_n)} \qquad \frac{1}{\hat{\beta}_1}\sum_{i=1}^{n}\frac{1}{1+\hat{\beta}_1 t_i}=\frac{nt_n}{(1+\hat{\beta}_1 t_n)\ln(1+\hat{\beta}_1 t_n)}$$

资料整理
火龙果软件
荐

Musa-Okumoto（M-O）模型-可靠性预计

- 可靠性预计

$$\lambda_n(x) = \hat{\beta}_0 \hat{\beta}_1 / (\hat{\beta}_1 (t_n + x) + 1) \qquad \mu_n(x) = \hat{\beta}_0 \ln(\hat{\beta}_1 (t_n + x) + 1)$$

$$R_{n+1}(x) = R(x \mid t_n) = \left[\frac{\hat{\beta}_1 t_n + 1}{\hat{\beta}_1 (t_n + x) + 1}\right]^{\hat{\beta}_0}$$

$$F_{n+1}(x) = 1 - R(x \mid t_n) = 1 - \left[\frac{\hat{\beta}_1 t_n + 1}{\hat{\beta}_1 (t_n + x) + 1}\right]^{\hat{\beta}_0}$$

资料整理
火龙果软件
荐

Musa-Okumoto（M-O）模型-可靠性预计

- 可靠性预计

$$f_{n+1}(x) = -\frac{dR_{n+1}(x)}{dx}$$

$$= \hat{\beta}_0 \hat{\beta}_1 (\hat{\beta}_1 t_n + 1)^{\hat{\beta}_0} / [\hat{\beta}_1 (t_n + x) + 1]^{\hat{\beta}_0 + 1}$$

$$T_{TF_{n+1}} = E\{X_{n+1} \mid x_1, \cdots, x_n\} = \int_0^{\infty} R_{n+1}(x) dx$$

$$= \frac{\hat{\beta}_1 t_n + 1}{\hat{\beta}_1 (\hat{\beta}_0 - 1)}$$

资料整理
火龙果软件
荐

软件可靠性模型

- 软件可靠性评估模型是随机过程的一种表示形式，将软件可靠性或与软件可靠性直接相关的量，例如：平均无失效时间、失效率等，表示成时间与软件产品的特性、或者开发过程的函数。
- 软件可靠性评估模型是将软件失效过程的一般形式规定为各因素，例如缺陷引入、缺陷移除和运行环境等的函数或数学表达式，是描述软件失效与软件缺陷之间关系的数学方程。

资料整理
火龙果软件
荐

软件可靠性模型的组成

- 模型假设
- 模型输出
- 参数估计方法
- 数据要求

资料整理
火龙果软件
荐

模型假设

- **模型假设：**模型是实际情况的理想或简单化，总是包含若干假设。
- **标准假设：**
  - 软件运行方式和进行软件可靠性预计时的软件运行方式相同—确保预计结果的有效性
  - 当检测到缺陷时，各个失效是独立的—简化推导

资料整理
火龙果软件
荐

模型输出

- 模型输出

软件可靠性模型常见的输出量有：可靠度、不可靠度、失效密度函数、失效强度、平均故障时间、残留缺陷数等。它们通常以数学表达式给出。

资料整理
火龙果软件
荐

参数估计方法

- 参数估计方法：模型输出中的某些值是无法直接得到的，最常见的比如残留缺陷数，这就要求模型给出一定的方法估计参数的值。
  - 极大似然法
  - 最小二乘法
  - 矩法
  - 贝叶斯估计

资料整理
火龙果软件
荐

数据要求

- 一个软件可靠性模型要求一定的输入数据，即软件可靠性数据。软件可靠性数据是在软件可靠性测试或实际使用中观测到的软件失效数据。
- 在实际采集过程中，通常可以得到两类数据：失效间隔时间数据（完全失效数据）、失效计数数据（不完全失效数据）。

资料整理
火龙果软件
荐

- 软件可靠性模型没有普适性，一个模型可能仅对一个或几个软件做出较为准确的评估和预计，因此，如何选择、评价模型就成为一个很有必要研究的课题。

资料整理
火龙果软件
荐

3 如何判断模型的“好坏”？

资料整理
火龙果软件
荐

模型的比较和选择

- 各种模型所表现出来的可靠性度量的准确性差异很大，目前还无法给出一个普遍适用的模型。
- Iannino、Musa、Okumoto和Littlewood等专家为可靠性模型设计了一套模型评价和比较的标准，主要包括预计的有效性、模型的能力、假设条件的质量、模型的适用性、模型的简洁性和对噪声的不敏感性等内容。
- 利用这些标准，可对模型进行比较和选择。此处重点介绍预计的有效性。

资料整理
火龙果软件
荐

预计的有效性

- 最小相关误差
- 拟合优度检验
- 序列似然比（PLR）法
- U图法
- Y图法

资料整理
火龙果软件
荐

最小相关误差——看谁预测的准

- 利用观测数据进行模型参数估计，将历史失效数据和基于观测数据的预计结果进行对比，预计结果和实际数据之间偏差最小的模型被视为最适合的模型。
- 这意味着在给定的失效数据集上对模型集进行比较，最佳拟合性可以通过最小相关误差来度量，例如：

$$ESS = \sqrt{\sum_{j=m}^{i} \frac{(actual_j - predicted_j)^2}{i-m}}$$

资料整理
火龙果软件
荐

拟合优度检验——检验总体是否服从某种分布

- χ^2 检验
- K-S距离检验

资料整理
火龙果软件
荐

序列似然比

- Prequential Likelihood Ratio（PLR）
- 序列似然比是用来对两个模型预计质量的相对优劣进行评定的一个指标。通过计算这个指标，可以比较不同模型对某一分布所作预计的相对准确程度。

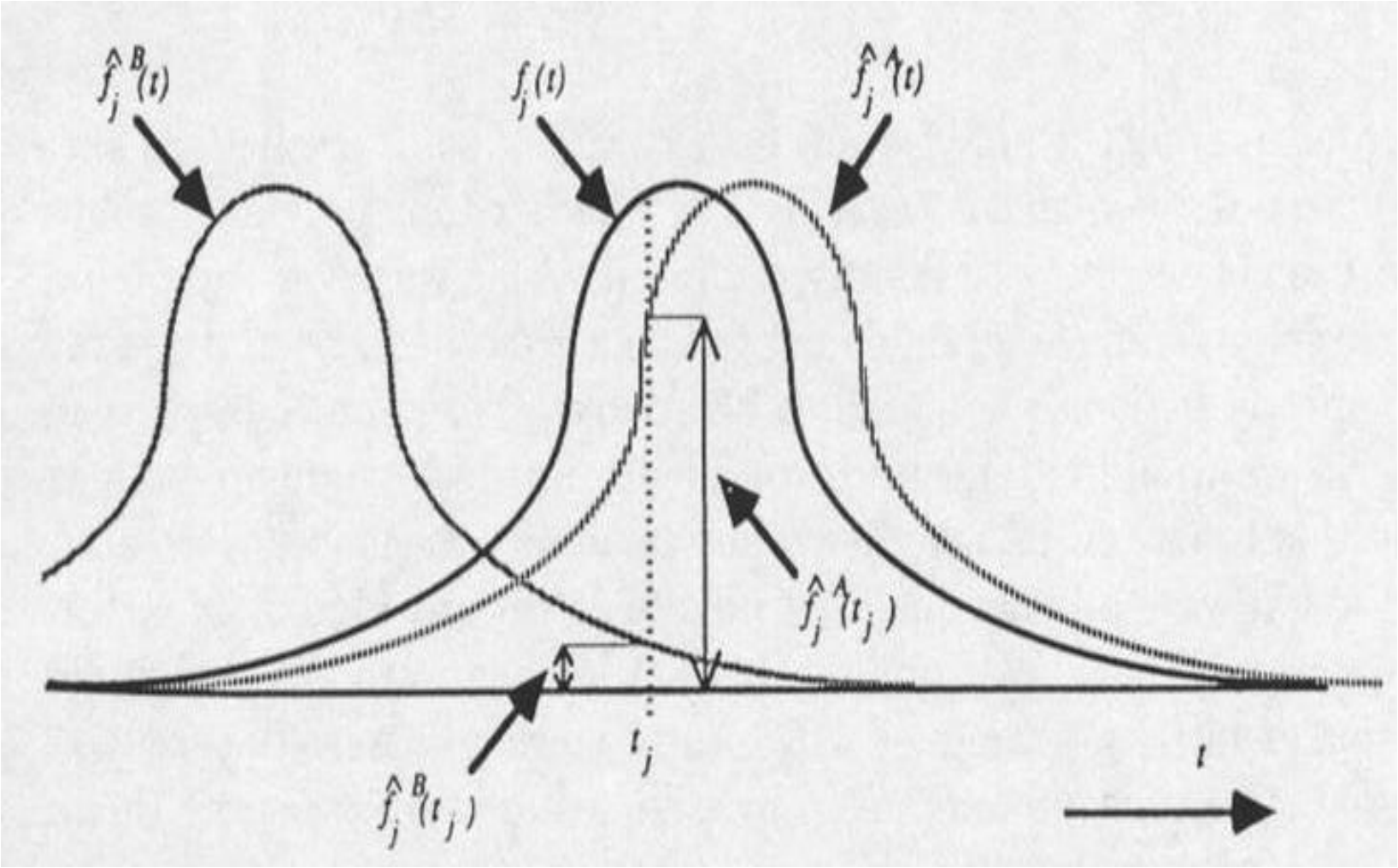
资料整理
火龙果软件
荐

序列似然比（PLR）法

- PLR法的总体思路是认为软件失效发生在软件失效时间的分布函数密度大的地方的可能性较大。
- 预计软件失效时间的分布函数密度大的预计相比较而言更接近于真实。
- 如比较预计模型A和预计模型B的准确性，PLR的定义为：

$$PLR_{s,i}^{AB} = \prod_{j=s}^{j=i} \frac{\tilde{f}_j^A(t_j)}{\tilde{f}_j^B(t_j)}$$

资料整理
火龙果软件
荐



资料整理
火龙果软件
荐

序列似然比（PLR）法

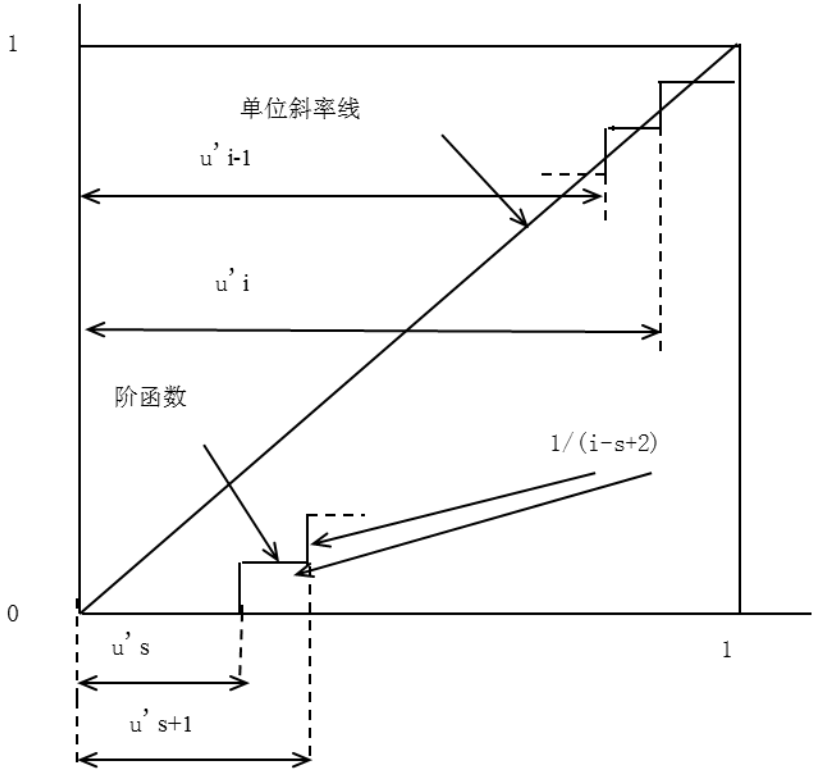
- $\tilde{f}_j^A$ 表示预计模型A的预计概率密度，$\tilde{f}_j^B$ 表示预计模型B的预计概率密度。
- 如果 $i \to \infty$ ，$PLR_{s,i}^{AB} \to \infty$，可以认为预计模型A优于预计模型B。
- 反之，可以认为预计模型B优于预计模型A。
- 如果比值既不趋于无穷又不趋于0，而是趋近于某一个常数c，则说明预计模型A和预计模型B提供不相上下的预计，显然，PLR法只能用于比较两个预计模型的相对预计性。

资料整理
火龙果软件
荐

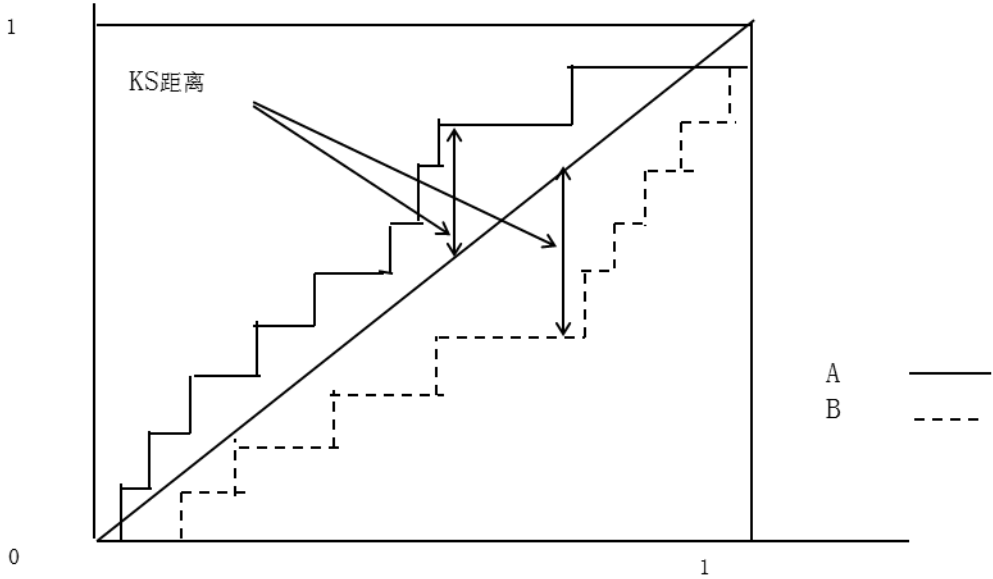
U图法

- U图是用来检测预计和观测的失效行为之间系统而客观的差别的。
- U图的目的是用来判断预测预计分布函数 $\hat{F}_j(t)$ 是否均匀地接近于实际分布 $F_j(t)$ 。
- U图就是序列{ u_j }的抽样分布函数图，用来表示{ u_j}的抽样累积分布函数接近U（0，1）的累积分布函数的程度。

资料整理
火龙果软件
荐



U图



KS距离

资料整理
火龙果软件
荐

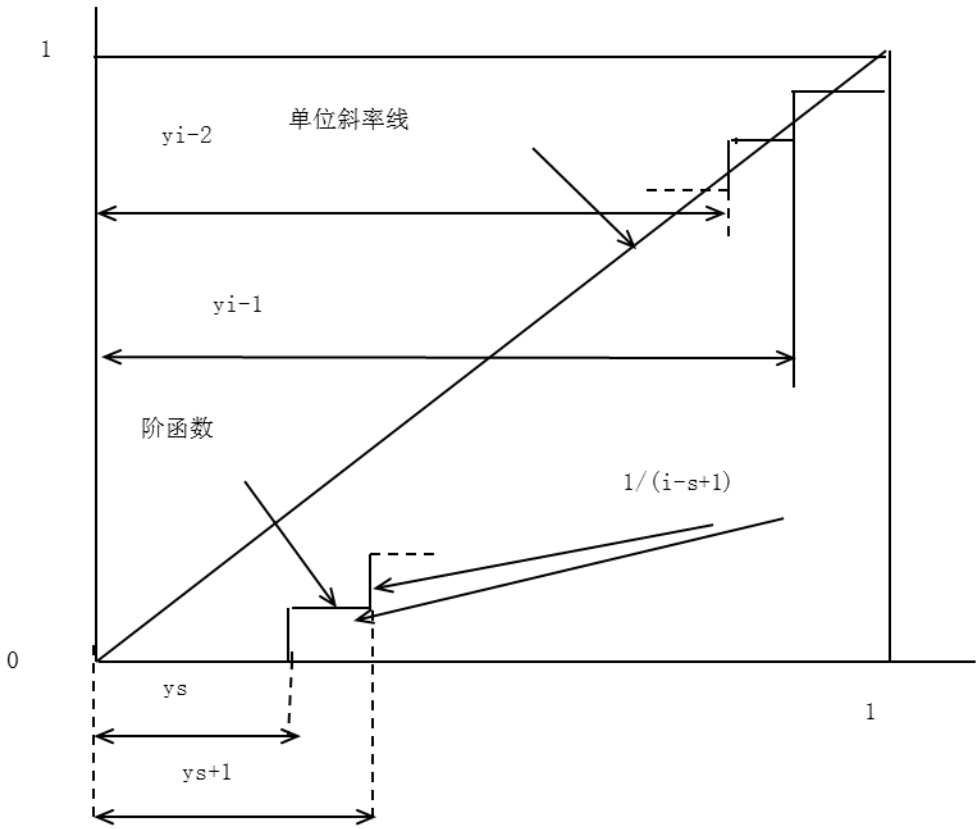
Y图法

- U图能够检测出预计与现实的严重偏差。但KS距离很小的U图有时也会掩盖了偏差，需要求助于Y图。Y图是通过对序列 u_j 进行变换后绘制的。
- Y图对{ u_j }作如下的进一步变换：

$$x_j = -\ln(1-u_j) \qquad y_k = \frac{\sum_{j=s}^{k} x_j}{\sum_{j=s}^{i} x_j}$$

- Y-图表示{ u_j }的样本累积分布函数与45度线的接近程度。

资料整理
火龙果软件
荐



Y图

资料整理
火龙果软件
荐

4 稳定使用阶段的软件可靠性评估

资料整理
火龙果软件
荐

稳定使用阶段的软件可靠性评估

研究表明：

软件在稳定使用阶段，且在发现缺陷不更改的情况下，软件的寿命服从指数分布。

资料整理
火龙果软件
荐

评估/预计结果

- 失效率=失效强度

$$Z(x)=\hat{\lambda}$$

- 不可靠度

$$F(x)=1-e^{-\hat{\lambda}x}$$

- MTTF

$$T_{TF}=\frac{1}{\hat{\lambda}}$$

资料整理
火龙果软件
荐

注意事项

- 稳定使用过程中，**发现失效后不对软件进行修改**，从而保证失效时间服从指数分布。
- 在评估前必须由测试方、开发方以及使用方（可能的话）明确失效的判据，并明确哪些失效需要统计在内。**对于不同的可靠性要求其失效判据可能不同，例如，如果只对影响任务的失效提出了可靠性定量要求，则失效应定义为影响软件系统任务完成的软件行为。**
- 在可靠性参数评估中，**应计入每一次失效**，无论其是否由相同的原因导致。

资料整理
火龙果软件
荐

某软件失效数据

失效序号	失效间隔时间（秒）	失效序号	失效间隔时间（秒）
1	897.8	31	242.7
2	79.6	32	622.5
3	442.5	33	132.7
4	119.1	34	524.2
5	76.8	35	98.4
6	95.3	36	268.5
7	212.2	37	92.6
8	73.7	38	382.8
9	73.3	39	101.1
10	637.9	40	81.8
11	359.7	41	166.3
12	392.6	42	598.5
13	617.7	43	56.7
14	393.6	44	238.8
15	1224	45	110.2

资料整理
火龙果软件
荐

某软件失效数据

失效序号	失效间隔时间（秒）	失效序号	失效间隔时间（秒）
16	112	46	93
17	313	47	74. 3
18	324	48	119. 8
19	259. 3	49	526. 4
20	175. 6	50	1202. 9
21	122. 8	51	801. 5
22	1686. 2	52	109. 2
23	259	53	235. 2
24	187. 2	54	1389. 7
25	552. 4	55	154. 1
26	541. 9	56	605. 4
27	725. 3	57	132
28	129. 5	58	72. 6
29	303. 5	59	335. 2
30	699. 2	60	127. 9

资料整理
火龙果软件
荐

示例

- 按照理论分析的结果，该软件寿命应该服从指数分布。因此选用指数模型对失效数据进行拟合，并按照指数模型的参数计算方法得到：
- 失效率为

$$Z(t)=\hat{\lambda}=N/\sum_{i=1}^{N}t_i=60/\sum_{i=1}^{60}t_i=2.75\times10^{-3}$$

- 可靠度函数：

$$R(t)=\exp(-2.75\times10^{-3}t)$$

资料整理
火龙果软件
荐

5 增长测试中的软件可靠性评估

资料整理
火龙果软件
荐

评估相关工作

- 定义失效
- 构造操作剖面和选取测试数据进行测试
- 收集软件失效数据
- 对失效数据进行趋势分析
- 选择软件可靠性模型
- 参数估计
- 模型验证
- 评估/预计分析

资料整理
火龙果软件
荐

收集软件失效数据

- 收集软件失效数据
  - 收集软件失效数据是采用模型进行软件可靠性评估的基础和必要条件。
  - 针对不同的可靠性要求，应该制定不同的数据收集程序和数据收集要求。
  - 常用的模型使用两类失效数据：失效计数数据和失效时间数据。

资料整理
火龙果软件
荐

失效时间数据

失效序号	失效间隔时间（小时）	累计失效时间（小时）
1	0.5	0.5
2	1.2	1.7
3	2.8	4.5
4	2.7	7.2
5	2.8	10.1
6	3.1	13.1
7	1.8	14.8
8	0.9	15.7
9	1.4	17.1
10	3.5	20.6
11	3.4	24.1
12	1.2	25.2

资料整理
火龙果软件
荐

失效时间数据

失效序号	失效间隔时间（小时）	累计失效时间（小时）
13	0.9	26.1
14	1.7	27.8
15	1.4	29.2
16	2.7	31.9
17	3.2	35.1
18	2.5	37.6
19	2.1	39.6
20	4.5	44.1
21	3.5	47.6
22	5.2	52.8
23	7.2	60.1
24	10.7	70.1

资料整理
火龙果软件
荐

失效计数数据

时间（小时）	失效数	累计失效数
8	4	4
16	4	8
24	3	11
32	5	16
40	3	19
48	2	21
56	1	22
64	1	23
72	1	24

资料整理
火龙果软件
荐

对失效数据进行趋势分析——判断可靠性趋势

- 图形法
- 拉普拉斯法

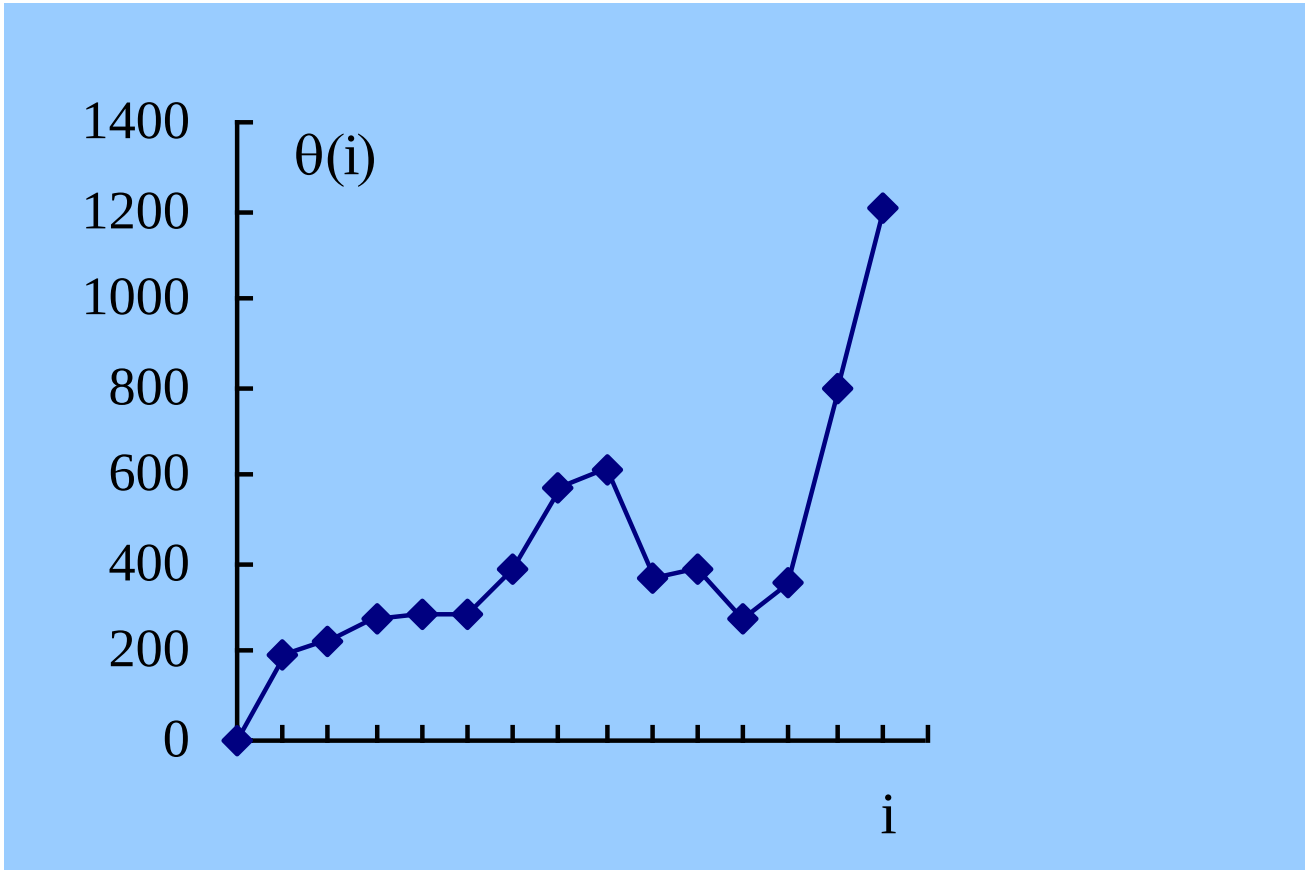
资料整理
火龙果软件
荐

图形法

- 图形法是在工程中粗略判断数据趋势的常用方法，被检验的数据可以是失效间隔时间，也可以是累积失效数或失效强度（单位时间内的失效数）。

资料整理
火龙果软件
荐

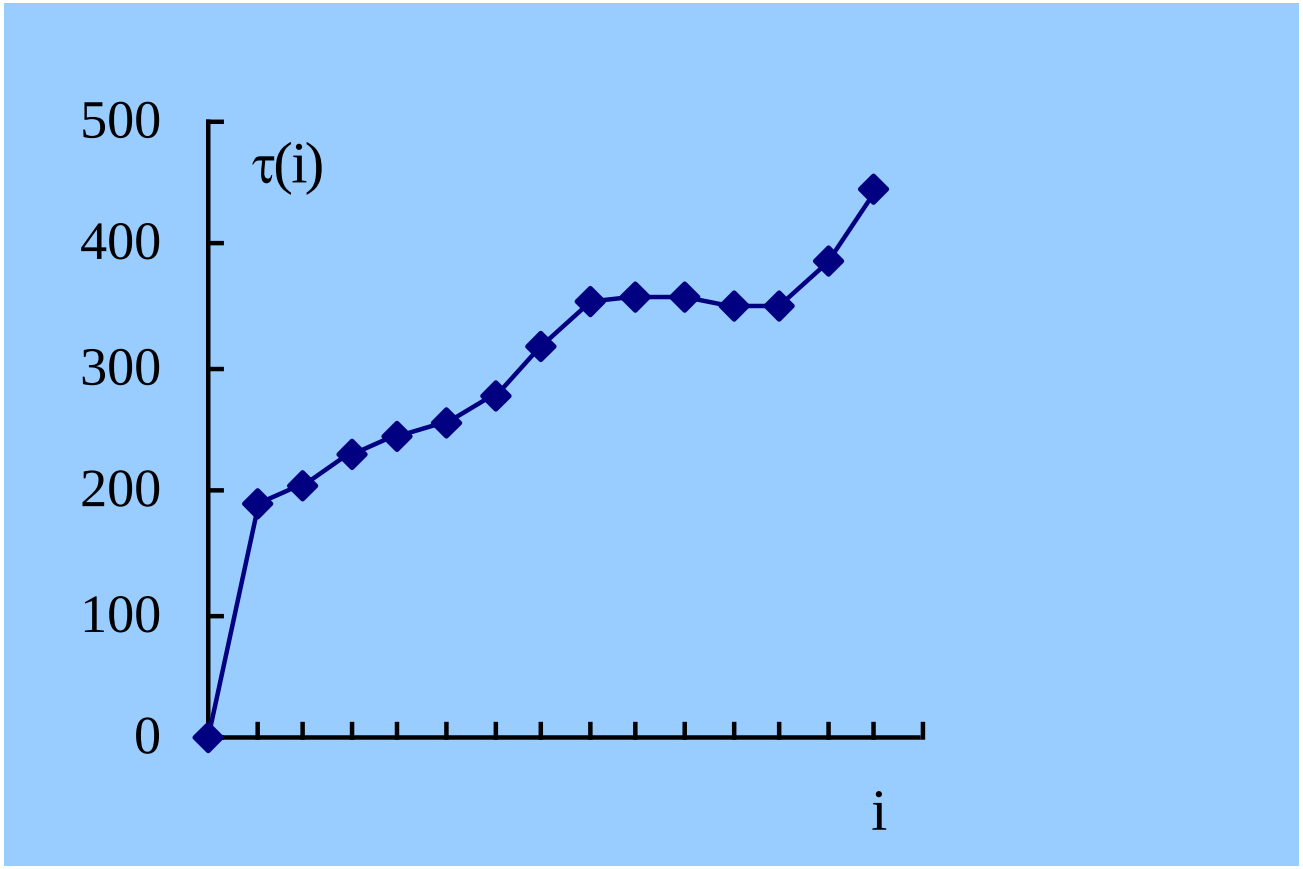
软件可靠性趋势分析——图形法



每次失效的失效间隔时间

资料整理
火龙果软件
荐

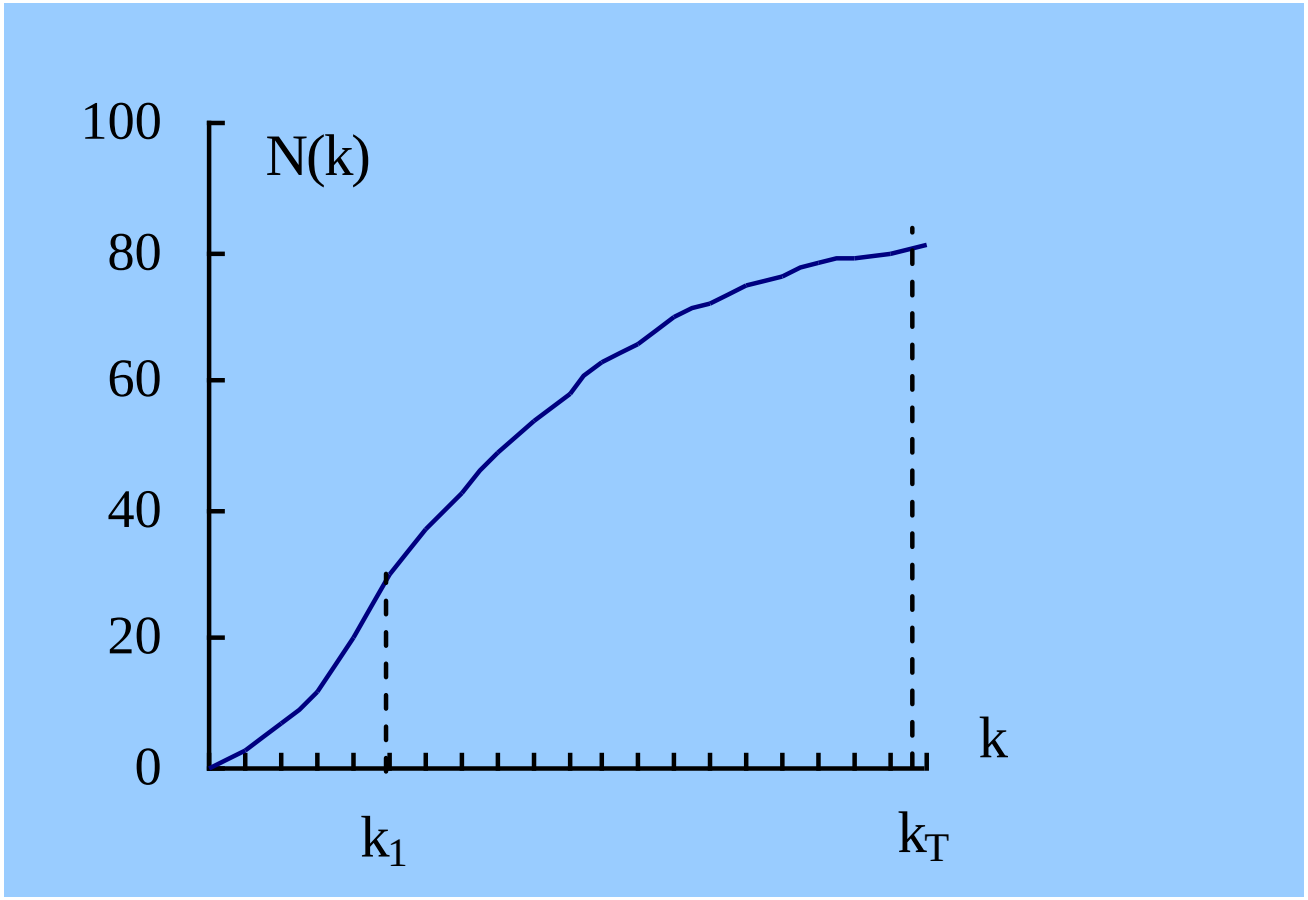
软件可靠性趋势分析——图形法



第i个失效发生后的平均失效时间

资料整理
火龙果软件
荐

软件可靠性趋势分析——图形法



给定时间的累计失效数

资料整理
火龙果软件
荐

拉普拉斯法

- 拉普拉斯法是用分析法检验失效数据趋势的一种最常用和最重要的方法。被检验的数据可以是失效间隔时间也可以是失效强度（单位时间内的失效数）或累积失效数。

资料整理
火龙果软件
荐

拉普拉斯法

- 计算拉氏因子

$$u(i)=\frac{\frac{1}{i-1}\sum_{n=1}^{i-1}\sum_{j=1}^{n}\theta_j-\frac{1}{2}\sum_{j=1}^{i}\theta_j}{\sum_{j=1}^{i}\theta_j\sqrt{\frac{1}{12(i-1)}}}\qquad u(k)=\frac{\sum_{i=1}^{k}(i-1)n(i)-\frac{k-1}{2}\sum_{i=1}^{k}n(i)}{\sqrt{\frac{k^2-1}{12}\sum_{i}^{k}n(i)}}\qquad u(k)=-\frac{\sum_{i=1}^{k}N(i)-\frac{k+1}{2}N(k)}{\sqrt{\frac{k^2-1}{12}N(k)}}$$

判断趋势

1）|μ(k)|<1.96时认为趋势稳定；

2）若1）不满足，则当u(k)<-1.96时认为可靠性增长；

3）若1）不满足，则当u(k)>1.96时认为可靠性下降。

资料整理
火龙果软件
荐

对失效数据进行趋势分析、选择模型

- 趋势分析结果可以分为以下类型：
  - 可靠性增长：选择可靠性增长模型；
  - 可靠性下降：一般这种情况下没必要进行可靠性估计；
  - 可靠性先降后升：使用S型模型；
  - 可靠性稳定趋势：用失效率为常值的模型

资料整理
火龙果软件
荐

参数估计

- 参数估计
  - 矩估计法，最小二乘法及最大似然法
  - 最大似然估计方法为最常用的经典统计方法。
  - 母体X具有密度函数 $f(x;\theta)$ ， $x=(x_1,x_2,\cdots,x_n)$ 是来自X的一个给定的子样，则似然函数：

$$L(\theta,x)=\prod_{i=1}^{n} f(x_i;\theta)$$

  - 假如 $\hat{\theta}(x)$存在，则有：$L(\hat{\theta}(x);x)=\sup_{\theta\in\Theta} L(\theta;x)$ ，此时 $\hat{\theta}(x)$称为参数 θ 的最大似然估计。似然方程：

$$\frac{\partial}{\partial\theta_i} LnL(\theta;x)\,|_{\theta_i=\hat{\theta}_i}=0$$

资料整理
火龙果软件
荐

模型验证

- 模型验证
  - 首先要考虑模型假设是否适合，其次，要选择适当的失效数据集。模型验证时的失效数据集应当按照如下推荐的优先级进行选取：
    - 该软件的已有失效数据集（如果该软件的运行时间足够长，且产生了足够多的失效数据）；
    - 该软件的快速原型项目中积累的失效数据集；
    - 该软件的先前版本积累的失效数据集。
  - 选用模型后，了解模型的预计有效性。
  - 在选择模型和应用所选模型之后，不断地对模型进行重新验证。

资料整理
火龙果软件
荐

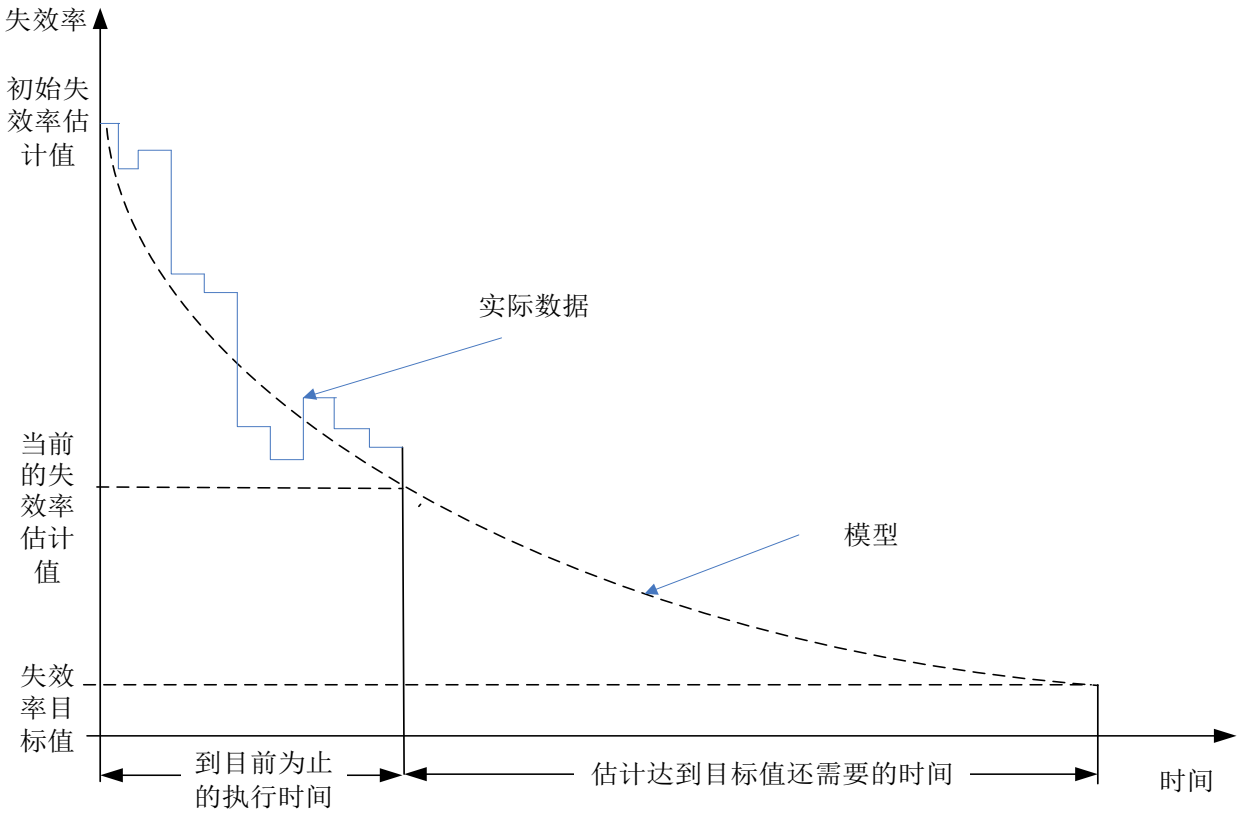
预计的有效性

- 最小相关误差
- 拟合优度检验
- 序列似然比（PLR）法
- U图法
- Y图法

资料整理
火龙果软件
荐

评估/预计分析

- 预计可靠性目标的实现



软件可靠性评估模型的应用

资料整理 火龙果软件 荐

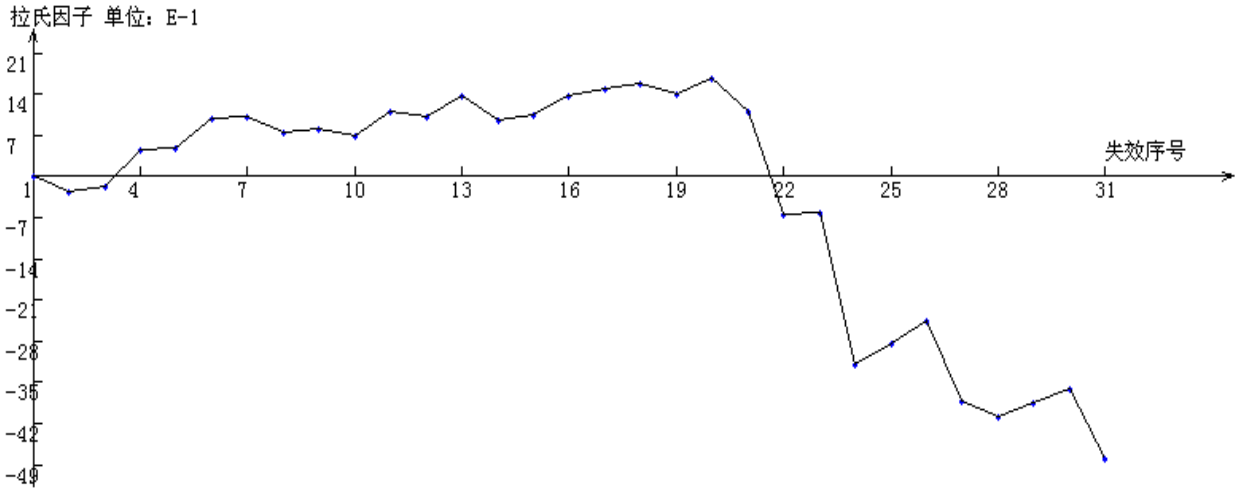
示例

失效序号	失效间隔时间（秒）	失效序号	失效间隔时间（秒）
1	9	17	3
2	12	18	3
3	11	19	6
4	4	20	1
5	7	21	11
6	2	22	33
7	5	23	7
8	8	24	91
9	5	25	2
10	7	26	1
11	1	27	87
12	6	28	47
13	1	29	12
14	9	30	9
15	4	31	135
16	1		

资料整理
火龙果软件
荐

示例

- 对失效数据进行趋势分析
  - 得到如图所示的趋势分析图，并能够得到数据趋势为可靠性增长的结论。



NTDS的趋势分析图

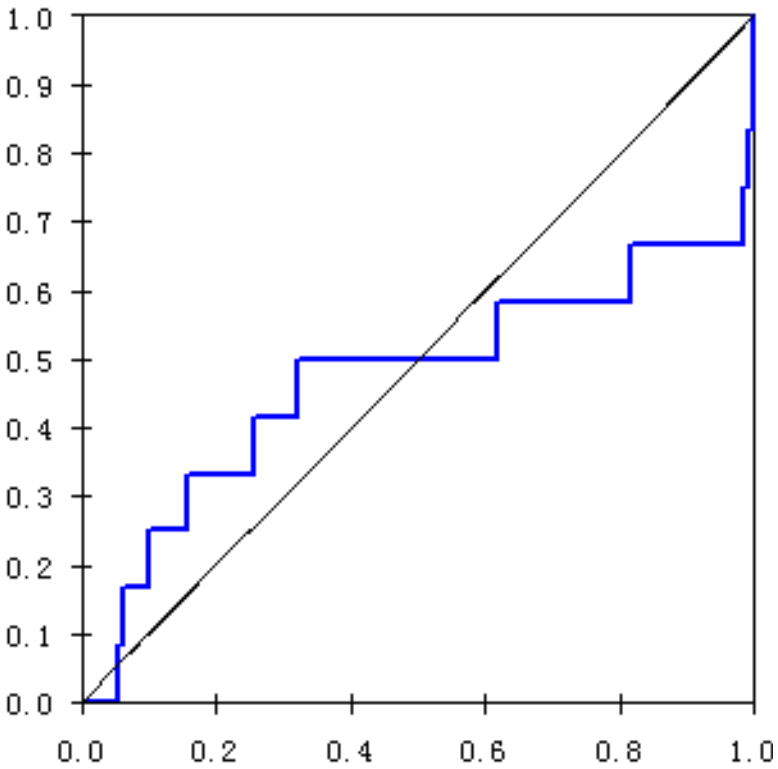
资料整理
火龙果软件
荐

示例

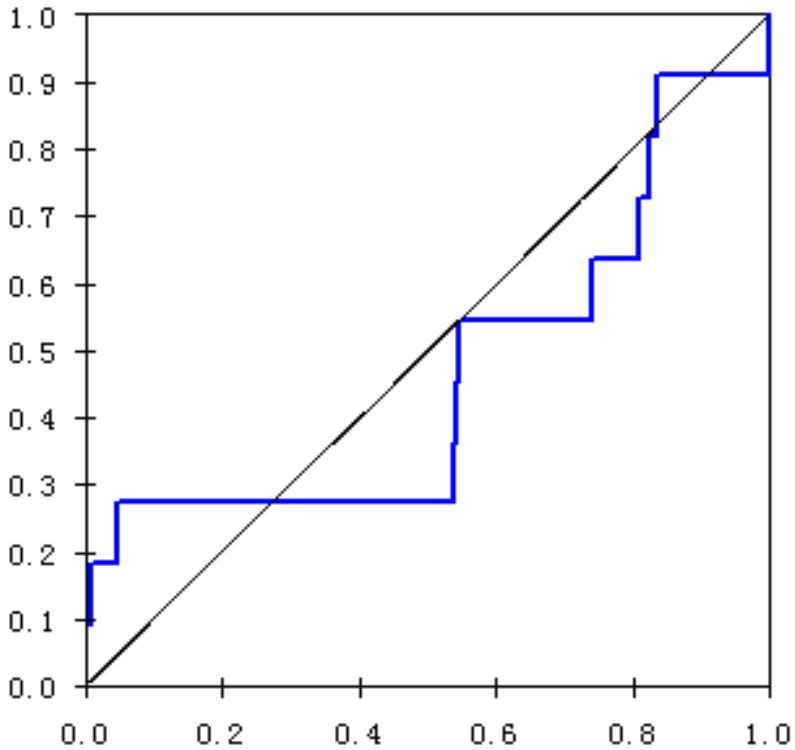
- 选择软件可靠性模型
  - 选定M-O模型和Schneidewind模型作为被选中的模型。
- 确定模型参数的估计方法
  - 假定根据失效数据的特性，此处选用最大似然法。
- 模型验证
  - 此处利用U图和Y图对模型的预计有效性进行验证。
  - 利用M-O模型得到的U图和Y图，其中U图KS距离=0.232，Y图KS距离=0.224。

资料整理
火龙果软件
荐

示例



利用M-O模型得到的U图

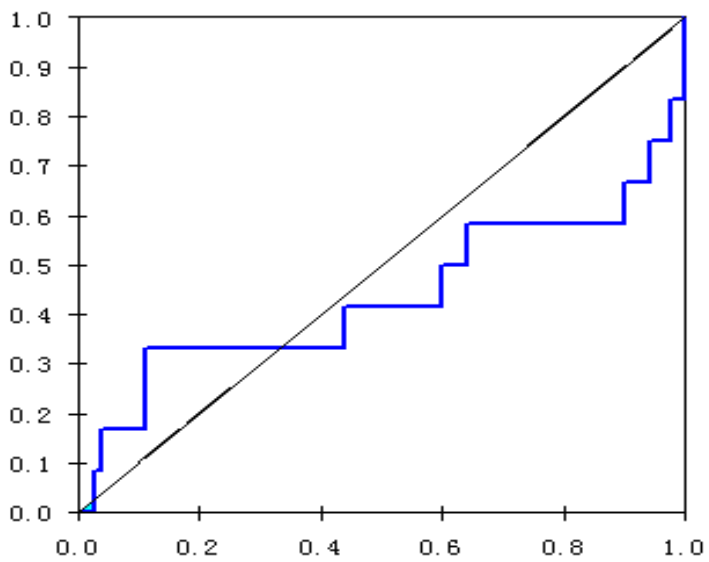


利用M-O模型得到的Y图

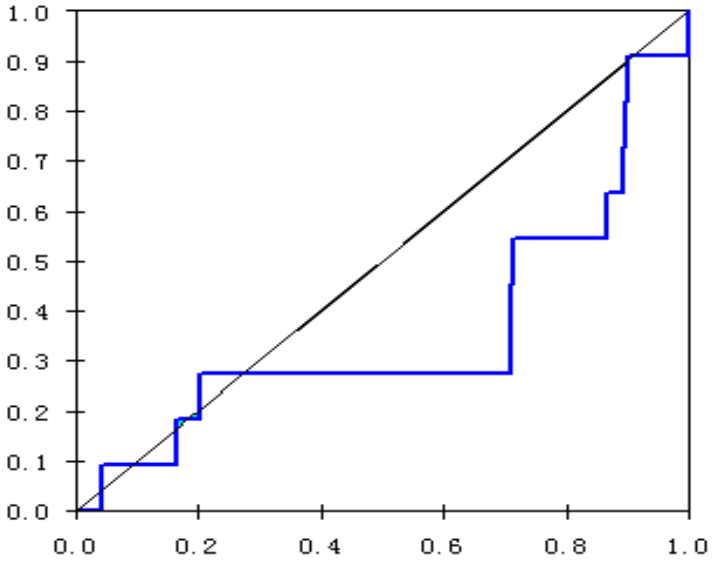
资料整理
火龙果软件
荐

示例

- 利用Schneidewind模型得到的U图和Y图，其中U图KS距离=0.235，Y图KS距离=0.345。



利用Schneidewind模型得到的U图



利用Schneidewind模型得到的Y图

资料整理
火龙果软件
荐

示例

- 通过对U图KS距离、Y图KS距离与K-S距离检验的临界值 的对比，可以看出，M-O模型能很好地与实测数据拟合，因此选用M-O模型。

- 基于拟合的模型做出判定或进行可靠性预计
  - 依据第1~31个失效数据，进行模型参数的估计，得到模型参数估计值：

$$\beta_0=13.29,\quad \beta_1=0.01722,\quad \theta=0.07524,\quad \lambda_0=0.22885。$$

资料整理
火龙果软件
荐

示例

- 第31个失效之后的时间x内的失效强度如下。当x=0时，可以得到当前的失效强度的估计值是0.02222。

$$\lambda_n(x)=\hat{\beta}_0\hat{\beta}_1/(\hat{\beta}_1(t_n+x)+1)=\frac{0.22885}{0.01722(540+x)+1}$$

- 第31个失效之后的时间x内的失效数均值函数：

$$u_n(x)=\hat{\beta}_0\ln(\hat{\beta}_1(t_n+x)+1)=13.29\ln(0.01722(540+x)+1)$$

- 第31个失效之后的时间x内的可靠度、不可靠度：

$$R_{n+1}(x)=R(x\mid t_n)=\left[\frac{\hat{\beta}_1 t_n+1}{\hat{\beta}_1(t_n+x)+1}\right]^{\hat{\beta}_0}=\left[\frac{10.2988}{0.01722(540+x)+1}\right]^{13.29}$$

资料整理
火龙果软件
荐

示例

$$F_{n+1}(x)=1-R(x\mid t_n)=1-\left[\frac{10.2988}{0.01722(540+x)+1}\right]^{13.29}$$

- 如当预计时间为第31个失效后的10秒时，预计的可靠度值为0.802。
- 第31个失效之后的时间x内的失效密度为

$$f_{n+1}(x)=-\frac{dR_{n+1}(x)}{dx}=\frac{0.22885\times 10.2988^{13.29}}{[0.01722(540+x)+1]^{14.29}}$$

- 第32个失效的MTTF：

$$\theta=\frac{1}{\beta_0}=0.07524 \qquad T_{TF_{n+1}}=\frac{\hat{\beta}_1 t_n+1}{\hat{\beta}_1(\hat{\beta}_0-1)}=\frac{0.01722\times 540+1}{0.01722\times(13.29-1)}=48.65$$

- 即根据1~31个失效数据，可以预计第32个失效的MTTF是48.65秒。

资料整理
火龙果软件
荐

示例

- 利用上述计算获得的结果就可以指导实际的测试进程，例如：
  - 如果用户给出失效强度要求时，根据当前的失效强度的估计值，就可以判断是否满足用户的要求，而决定是否可以停止测试；
  - 根据失效强度的预计函数，可以知道任何时刻的失效强度，如果用户给出失效强度要求时，就可以确定还需进行多长时间的测试能够达到用户给出的失效强度要求。

资料整理
火龙果软件
荐

结束

Thank You!

资料整理
火龙果软件
荐