Imprecise Software Reliability

Lev Utkin

Munich, September 2009

Standard definitions of software reliability

- A **fault** in a software is an incorrect step, process, or data definition.
- A fault may cause a **failure**, or "the inability of a system or component to perform its required functions within specified performance requirements"; that is, a deviation from the stated or implied requirements.
- An **error** is a "discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

Two types of software models

There are two main software reliability models:

1. **Reliability growth model**
   - *statistical data are obtained during debugging process under condition that a detected error is removed or corrected.*
2. **Reliability growth model**
   - *statistical data are obtained during testing under condition that a detected error is not removed or corrected.*

Standard initial information concerning software reliability

Three main kinds of reliability statistical data.

1. Calendar times (reals) between software failures $\mathbf{T} = (t_1, ..., t_n)$.
2. Numbers of successful runs (a run is minimum execution unit of software, integers) between software failures $\mathbf{K} = (k_1, ..., k_n)$.
3. Numbers of failures in certain periods of time $(k_1, t_1), ..., (k_n, t_n)$.

Main goals of modelling

To predict the reliability measures or indices of a analyzed software after:

1. **Debugging process (**growth models**)**
2. **Testing (**testing models**)**

In other words, we have to compute the probabilistic measures of the r.v. X_{n+1} (time or number of runs to failure, number of failures after n observations).

Three types of models

1. Calendar times: continuous models (continuous random variables)
2. Numbers of successful runs: discrete models (discrete random variables).
3. Numbers of failures in certain periods of time: non-homogeneous Poisson process models.

Assumptions in most well-known reliability growth models

1. The operational profile of the software remains constant, i.e., the software where the data comes from is operated in a similar manner as that in which reliability predictions are to be made.
2. Every model assume some precise probability distribution of random variables under consideration.
3. The failures, when the faults are detected, are independent, i.e., random variables under consideration are statistically independent.
4. Every fault has an equal chance of being encountered within a severity class as any other fault in that class.
5. After a failure, the fault causing it is corrected immediately and no new faults result from the correction.

Assumptions in most well-known reliability testing models

1. The software doesn't change during testing and usage, except that faults are fixed.
2. The operational profile of the software remains constant, i.e., the software where the data comes from is operated in a similar manner as that in which reliability predictions are to be made.
3. Every model assume some precise probability distribution of random variables under consideration.
4. The failures, when the faults are detected, are independent, i.e., random variables under consideration are statistically independent.

Assumptions in most well-known reliability models

The assumptions are usually not fulfilled. As a result, the reliability may be too unreliable and risky.

A standard way for dealing the models

1. Let X_i be a random time interval between the $(i-1)$-st and i-th software failures. The variable X_i is governed by a probability density function $p_i(x|\theta)$ with a vector of parameters θ_i. It is assumed that there holds $\theta_i = f(i, \theta)$.

A standard way for dealing the models

1. Let X_i be a random time interval between the $(i-1)$-st and i-th software failures. The variable X_i is governed by a probability density function $p_i(x|\theta)$ with a vector of parameters θ_i. It is assumed that there holds $\theta_i = f(i, \theta)$.
2. Let $\mathbf{X} = (x_1, ..., x_n)$ be the successive intervals between failures. The likelihood function is

$$L(\mathbf{X}|\theta) = \Pr\{X_1 = x_1, ..., X_n = x_n|\theta\} = \prod_{i=1}^{n} p_i(x_i|\theta).$$

A standard way for dealing the models

1. Let X_i be a random time interval between the $(i-1)$-st and i-th software failures. The variable X_i is governed by a probability density function $p_i(x|\theta)$ with a vector of parameters θ_i. It is assumed that there holds $\theta_i = f(i, \theta)$.
2. Let $\mathbf{X} = (x_1, ..., x_n)$ be the successive intervals between failures. The likelihood function is

$$L(\mathbf{X}|\theta) = \Pr\{X_1 = x_1, ..., X_n = x_n|\theta\} = \prod_{i=1}^{n} p_i(x_i|\theta).$$

3. $\theta_{opt} = \arg\max_\theta L(\mathbf{X}|\theta)$

A standard way for dealing the models

1. Let X_i be a random time interval between the $(i-1)$-st and i-th software failures. The variable X_i is governed by a probability density function $p_i(x|\theta)$ with a vector of parameters θ_i. It is assumed that there holds $\theta_i = f(i, \theta)$.
2. Let $\mathbf{X} = (x_1, ..., x_n)$ be the successive intervals between failures. The likelihood function is

$$L(\mathbf{X}|\theta) = \Pr\{X_1 = x_1, ..., X_n = x_n|\theta\} = \prod_{i=1}^{n} p_i(x_i|\theta).$$

3. $\theta_{opt} = \arg\max_\theta L(\mathbf{X}|\theta)$
4. The software failure function after the n-th software failure is computed as follows:

$$F_{n+1}(t) = \int_0^t p_{n+1}(x|\theta_{opt})dx.$$

Jelinski-Moranda model

1. The initial number of faults in the software is N.
2. The time between failures follows an exponential distribution with a parameter that is proportional to the number of remaining faults in the software.
3. The mean time between the $(i-1)$-st and i-th failures is $1/\lambda(N-(i-1))$.
4. $$L(\mathbf{X}|\lambda, N) = \prod_{i=1}^{n} \lambda(N-(i-1)) \exp\left(-\lambda(N-(i-1))t\right) \to \max_{\lambda,N} .$$
5. $$F_{n+1}(t) = \exp\left(-\lambda(N-n)t\right).$$

Schick-Wolverton and Littliwood-Verrall models

1. **Schick-Wolverton model:** the time between failures is governed by the Rayleigh distribution with the pdf

$$f_i(t) = t\lambda \cdot \exp\left(-\lambda_i t^2/2\right).$$

2. **Littliwood-Verrall model:** the time between failures is governed by the Pareto distributions with the pdf

$$f_i(t) = \frac{\sigma\left(\psi(i)\right)^{\sigma}}{\left[t + \psi(i)\right]^{\sigma+1}},$$

$$\psi(i) = \beta_0 + \beta_1 i, \quad \psi(i) = \beta_0 + \beta_1 i^2.$$

Non-homogeneous Poisson process (NHPP) models

For any time points $0 < t_1 < t_2 < ...$, the probability that the number of failures between t_{i-1} and t_i is k can be written as

$$\begin{aligned} &\Pr\left\{N(t_i) - N(t_{i-1}) = k\right\} \\ &= \frac{\left\{m(t_i) - m(t_{i-1})\right\}^k}{k!} \exp\left\{-\left(m(t_i) - m(t_{i-1})\right)\right\}. \end{aligned}$$

Here $m(t)$ is the mean number of failures occuring up to time t. The NHPP models differ by a mean value function $m(t)$.

Specific NHPP models

- $m(t) = at^b$ (Duan model),
- $m(t) = a(1 - (1 - b)^t)$ (Yamada-Ohba-Osaki model),
- $m(t) = a(1 - \exp(-bt))$ (Goel-Okumoto model),
- $m(t) = a\ln(1 + bt)$ (Musa-Okumoto model).

General class of NHPP models

It has been shown by Pham *et al* (Pham, Nordmann, Zhang 1999) that a general class of NHPP models can be obtained by solving the differential equation

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t} = b(t)\left[a(t) - m(t)\right]$$

with suitably chosen $a(t)$ and $b(t)$.
The parameters can be estimated using the maximum likelihood method based on the number of failures per interval of testing and debugging.

Fuzzy software reliability models (Cai's models)

- The time between the $(i-1)$-st and i-th failures X_i is a fuzzy variable governed by a membership function $\mu_i(x)$, for example $\mu_i(x) = \exp\left(-(x-a_i)^2\right)$ (*Kai-Yuan Cai et al 1991,1993*).
- $a_i = f(i, \theta)$, for example, $f(i) = (A + Bi)^{-\alpha} + C$, $A, B, C, \alpha \in \theta$.
- The possibilistic likelihood function is

$$L(x_1, .., x_n|\theta) = \mathrm{Pos}\left\{X_1 = x_1, .., X_n = x_n\right\} = \min\left\{\mu_1(x_1), ., \mu_n(x_n)\right\}.$$

- The reliability after n-th software failure is

$$R(t) = \sup_{x \geq t} \mu_{n+1}(x).$$

- The main difficulty is how to interpret $R(t)$.

The first idea: maximum of the likelihood function over the set of CDFs (discrete case)

1. Every X_i is governed by an unknown CDF belonging to a set $\mathcal{M}_i(\mathbf{d})$ depending on a vector of parameters $\mathbf{d}$ and defined by **lower and upper CDFs**:

$$\underline{F}_i(k \mid \mathbf{d}) = \inf_{F(k)\in\mathcal{M}_i(\mathbf{d})} F(k), \ \overline{F}_i(k \mid \mathbf{d}) = \sup_{F(k)\in\mathcal{M}_i(\mathbf{d})} F(k).$$

2. The likelihood function $L(\mathbf{K} \mid \mathbf{d}, F)$ is maximized over all distributions F from $\mathcal{M}_i(\mathbf{d})$ and the resulting "modified" likelihood function depends on $\mathbf{d}$:

$$L(\mathbf{K} \mid \mathbf{d}) = \max_{F\in\mathcal{M}_1(\mathbf{d}),\dots,F\in\mathcal{M}_n(\mathbf{d})} L(\mathbf{K} \mid \mathbf{d}, F).$$

The maximized likelihood function

Proposition

If random variables $X_1, ..., X_n$ are independent and discrete, then there holds

$$\max_{\mathcal{M}} \Pr\{X_1 = k_1, ..., X_n = k_n\} = \prod_{i=1}^{n} \left\{ \overline{F}_i(k_i) - \underline{F}_i(k_i - 1) \right\}.$$

“Precise” case

Corollary

If $\overline{F}_i(k) = \underline{F}_i(k) = F_i(k)$, *then*

$$\max_{\mathcal{M}} \Pr\left\{X_1 = k_1, ..., X_n = k_n\right\} = \prod_{i=1}^{n} p_i(k_i) = L(\mathbf{K} \mid \mathbf{d}).$$

Here $p_i(k)$ *is the probability mass function corresponding to the distribution function* $F_i(k)$.

We have the standard likehood function.

The maximized likelihood function (the lack of independence)

Proposition

If there is no information about independence of random variables $X_1, ..., X_n$, then there holds

$$\max_{\mathcal{M}} \Pr\left\{X_1 = k_1, ..., X_n = k_n\right\} = \min_{i=1,...,n} \left\{\overline{F}_i(k_i) - \underline{F}_i(k_i - 1)\right\}.$$

"Precise" case

Corollary

If $\overline{F}_i(k) = \underline{F}_i(k)$, *then*

$$\max_{\mathcal{M}} \Pr\left\{X_1 = k_1, ..., X_n = k_n\right\} = \min_{i=1,...,n} p_i(k_i).$$

We have the possibilistic likehood function.

Justification

The likelihood function is the probability

$$L(\mathbf{K} \mid \mathbf{d}) = \Pr\left\{X_1 = k_1, ..., X_n = k_n \mid \mathbf{d}\right\}.$$

Our final goal is to maximize this probability over a set of parameters **d**. But we have a set of probabilities. Therefore, we choose the largest probability in the set, i.e., we maximize the likelihood function over the set of probabilities depending on **d**.

The second idea: imprecise Bayesian inference

- Every set $\mathcal{M}_i(\mathbf{d})$ is defined by boundary **lower and upper CDFs** $\underline{F}_i(k \mid \mathbf{d})$ and $\overline{F}_i(k \mid \mathbf{d})$ whichcan be determined by using imprecise Bayesian inference.
- Before debugging process: we do not have information about $\underline{F}_i(k \mid \mathbf{d})$ and $\overline{F}_i(k \mid \mathbf{d})$. They are 0 and 1.
- After debugging process: we have n observations and by taking one of the imprecise Bayesian models (corresponding to a specific probability distribution), we detemine the lower and upper CDFs depending on $\mathbf{d}$.

Now we can construct SRGMs by taking the corresponding probability distributions for imprecise Bayesian inference.

The imprecise beta-binomial model (1)

Assumption 1: The i-th run lifetime of software X_j is governed by the geometric distribution with parameter p_j, $j = 1, ..., n$.
The prior Beta distribution of the random variable p is:

$$\pi(p) = \mathrm{BetaDen}(a, b) = \frac{1}{\mathrm{Beta}(a, b)} p^{a-1}(1-p)^{b-1},\ 0 \leq p \leq 1.$$

Here $a > 0$, $b > 0$ are parameters.
The posterior beta distribution $\pi(p|k)$ after k events under consideration from the total number of n event:

$$\pi(p|k) = \mathrm{BetaDen}(a + k, b + n - k).$$

The imprecise beta-binomial model (2)

Assumption 2: Introduce the growth function $\psi(j) = (j-1)\varphi$ (with parameter φ) such that parameters of the posterior beta distribution are

$$a^* = a + j - 1, \quad b^* = b + D_j,$$

$$D_j = K_j + \psi(j), \quad K_j = \sum_{i=1}^{j-1}(k_i - 1), \; \psi(j) = (j-1)\varphi.$$

The predictive CDF $F_j(m)$ for the j-th step of the debugging is

$$F_j(m|\varphi) = 1 - \frac{\Gamma(\alpha^* + \beta^*)}{\Gamma(\beta^*)} \frac{\Gamma(\beta^* + m)}{\Gamma(\alpha^* + \beta^* + m)}.$$

The imprecise beta-binomial model (3)

Introduce new parameters $s > 0$ and $\gamma \in [0, 1]$ such that

$$a = s\gamma, \;\; b = s - s\gamma.$$

Then

$$\underline{F}_j^{(s)}(m|\varphi) = 1 - \frac{\mathrm{Beta}(s + j - 1 + D_j, \;\; m)}{\mathrm{Beta}(s + D_j, \;\; m)},$$

$$\overline{F}_j^{(s)}(m|\varphi) = 1 - \frac{\mathrm{Beta}(s + j - 1 + D_j, \;\; m)}{\mathrm{Beta}(D_j, \;\; m)}.$$

The imprecise beta-binomial model (4)

$$\varphi_0 = \arg\max_{\varphi} L^{(s)}(\mathbf{K}|\varphi)$$
$$= \prod_{j=1}^{n} \left(\frac{\mathrm{Beta}(s+j-1+D_j,\ k_j-1)}{\mathrm{Beta}(s+D_j,\ k_j-1)} - \frac{\mathrm{Beta}(s+j-1+D_j,\ k_j)}{\mathrm{Beta}(D_j,\ k_j)} \right).$$

$$\underline{F}_{n+1}^{(s)}(m) = 1 - \frac{\mathrm{Beta}(s+n+K_{n+1}+n\varphi_0,\ m)}{\mathrm{Beta}(s+K_{n+1}+n\varphi_0,\ m)},$$

$$\overline{F}_{n+1}^{(s)}(m) = 1 - \frac{\mathrm{Beta}(s+n+K_{n+1}+n\varphi_0,\ m)}{\mathrm{Beta}(K_{n+1}+n\varphi_0,\ m)}.$$

The imprecise negative binomial model

The number of failures has a Poisson distribution with the parameter λ. If we observed K failures during a period of time T, then the predictive probability of k failures during time t under condition that K failures were observed during time T is

$$
\begin{aligned}
P(k,t) &= \int_0^\infty \frac{(\lambda t)^k e^{-\lambda t}}{k!} \mathrm{Gamma}(\alpha^*, \beta^*) \mathrm{d}\lambda \\
&= \frac{\Gamma(\alpha^* + k)}{\Gamma(\alpha^*) k!} \left(\frac{\beta^*}{\beta^* + t} \right)^{\alpha^*} \left(\frac{t}{\beta^* + t} \right)^k .
\end{aligned}
$$

Here $\alpha^* = \alpha + K$, $\beta^* = \beta + T$.

The imprecise negative binomial growth model (1)

- **Assumption:** $m(t; a, b) = a \cdot \tau(t, b)$ (the parameter a can be written separately).
- The parameter λ and the argument t of the Poisson distribution are replaced by the parameter a and the discrete time $\tau(t_i, b) - \tau(t_{i-1}, b)$, respectively.

The imprecise negative binomial growth model (2)

The predictive CDF of the number of failures in the interval between t_i and t ($t \in [t_i, t_{i+1}]$) after n periods is

$$F_i(k,t|\mathbf{c},b) = 1 - \frac{\mathrm{B}_{q(i,t)}(k+1,\alpha+K_n)}{\mathrm{B}(k+1,\alpha+K_n)} = 1 - I\left(q(i,t),k+1,\alpha+K_n\right).$$

$$q(i,t) = \frac{T_i(t,b)}{\beta+\tau(t_n,b)+T_i(t,b)},$$
$$T_i(t,b) = \tau(t,b) - \tau(t_i,b),\ K_n = \textstyle\sum_{j=1}^{n} k_j,$$

$\mathrm{B}_q(k+1,r)$ is the incomplete Beta-function with $I\left(q,k,r\right)$ the regularized incomplete Beta-function.

The imprecise negative binomial growth model (3)

- Choose all vectors (α, β) within the triangle $(0,0)$, $(s_1, 0)$, $(0, s_2)$. All possible prior rates of occurrence of failures are represented, as the prior allows interpretation of $\alpha/\beta = \gamma$ as this rate, hence this would include all such rates in $(0, \infty)$.
-

$$\underline{F}_i(k, t \mid s_1, s_2, b) = 1 - I\left(\frac{T_i(t,b)}{\tau(t_n, b) + T_i(t,b)}, k+1, s_1 + K_n\right),$$

$$\overline{F}_i(k, t \mid s_1, s_2, b) = 1 - I\left(\frac{T_i(t,b)}{s_2 + \tau(t_n, b) + T_i(t,b)}, k+1, K_n\right).$$

- The likelihood function is

$$L(\mathbf{K}|b, s) = \prod_{i=1}^{n} \left(\overline{F}_i(k_i, t_i \mid s_1, s_2, b) - \underline{F}_i(k_i - 1, t_i \mid s_1, s_2, b)\right).$$

About two caution parameters

1. The lower bound $\underline{\mathbb{E}}_i^{(s)} X$ is

$$\underline{\mathbb{E}}_i^{(s)} X = K \frac{t}{s+T}$$

In fact, the parameter s here increases the time of testing on the value s (hidden time).

2. The upper bound $\overline{\mathbb{E}}_i^{(s)} X$ is

$$\overline{\mathbb{E}}_i^{(s)} X = (s+K) \frac{t}{T}$$

In fact, the parameter s here increases the number of failures on the value s (hidden number of failure).

The imprecise negative binomial growth model (3)

The cumulative probability distribution of the number of failures in time interval $[t_n, t]$ after n periods of debugging

$$\underline{F}_{n+1}(k, t|s_1, s_2) = 1 - I\left(\frac{T_n(t, b)}{\tau(t_n, b) + T_n(t, b)}, k + 1, s_1, +K_n\right),$$

$$\overline{F}_{n+1}(k, t|s_1, s_2) = 1 - I\left(\frac{T_n(t, b)}{s_2 + \tau(t_n, b) + T_n(t, b)}, k + 1, K_n\right).$$

Imprecise modifications of NHPP models

1. **Imprecise Bayesian modifications of Musa-Okumoto model:** $m(t) = a \ln(1 + bt)$.
   For the model:

$$\tau(t, b) = \ln(1 + bt), \;\; T_j(t, b) = \frac{\ln(1 + bt)}{\ln(1 + bt_j)}.$$

Imprecise modifications of NHPP models

1. **Imprecise Bayesian modifications of Musa-Okumoto model:** $m(t) = a\ln(1+bt)$.
   For the model:
   $$\tau(t,b) = \ln(1+bt),\ \ T_j(t,b) = \frac{\ln(1+bt)}{\ln(1+bt_j)}.$$
2. **Imprecise Bayesian modification of Goal-Okumoto model:** $m(t) = a(1-\exp(-bt))$.
   $$\tau(t,b) = 1-\exp(-bt).$$

Validation of models

Algorithm: We predict the $(i+1)$-st mean time to failure $\underline{\mathbb{E}}^{(s)}_{i+1}X_{i+1}$, $\overline{\mathbb{E}}^{(s)}_{i+1}X_{i+1}$, $\mathbb{E}^{(s)}_{i+1}X_{i+1}$, starting from $i=3$. Then we compare these values with the actual times to failure k_{i+1}.

Measures of model quality:

$$R_1 = M^{-1}\cdot\left|\underline{\mathbb{E}}^{(s)}_{i+1}X_{i+1} - k_{i+1}\right|,\ R_2 = M^{-1}\cdot\left|\overline{\mathbb{E}}^{(s)}_{i+1}X_{i+1} - k_{i+1}\right|,$$

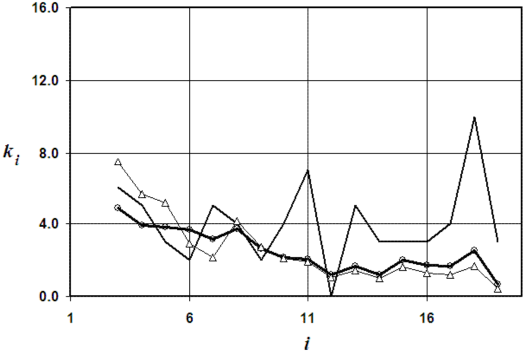
$$R_3 = M^{-1}\cdot\left|\mathbb{E}^{(s)}_{i+1}X_{i+1} - k_{i+1}\right|,\ R_4 = M^{-1}\cdot\left|\mathbb{E}_{i+1}X_{i+1} - k_{i+1}\right|,$$

where M is the number of predicted times to failure,

$$\mathbb{E}^{(s)}_{i+1}X_{i+1} = \gamma\underline{\mathbb{E}}^{(s)}_{i+1}X_{i+1} + (1-\gamma)\overline{\mathbb{E}}^{(s)}_{i+1}X_{i+1},\ \gamma = 0.5.$$

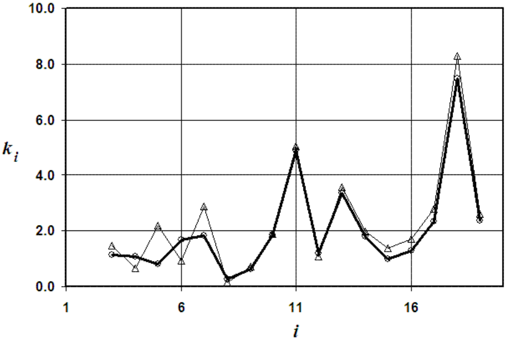
Validation of the imprecise Goal-Okumoto model (1)

Predicted expected values of numbers of failures for different models based on the Goal-Okumoto model



Validation of the imprecise Goal-Okumoto model (2)

Deviations of the predicted expected values of numbers of failures from the given values in data sets for the Goal-Okumoto models



Validation of the imprecise Goal-Okumoto model (3)

- The measures of quality by $s=1$ after predicting 17 numbers of failures (from the 3-rd test till 19-th test) by means of the imprecise Bayesian Goel-Okumoto model and the standard Goel-Okumoto model

$$R_1 = 7.491,\ \ R_2 = 2.054,\ \ R_3 = 1.786,$$

$$R_1^* = 8.311,\ \ R_2^* = 2.312,\ \ R_3^* = 1.959.$$

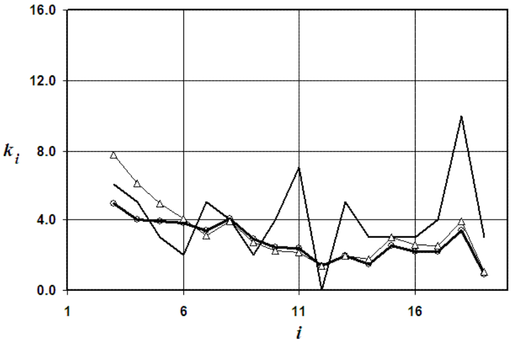
- After predicting 6 numbers of failures (from the 3-rd test till 8-th test)

$$R_1 = 1.827,\ \ R_2 = 1.133,\ \ R_3 = 0.568,$$

$$R_1^* = 2.854,\ \ R_2^* = 1.373,\ \ R_3^* = 1.007.$$

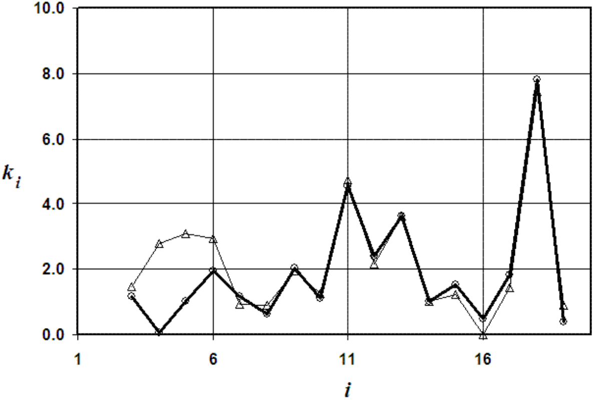
Validation of the imprecise Musa-Okumoto model (1)

Predicted expected values of numbers of failures for different models based on the Musa-Okumoto model



Validation of the imprecise Musa-Okumoto model (2)

Deviations of the predicted expected values of numbers of failures from the given values in data sets for the Musa-Okumoto models



Validation of the imprecise Musa-Okumoto model (3)

- The measures of quality by $s = 1$ after predicting 17 numbers of failures (from the 3-rd test till 19-th test) by means of the imprecise Bayesian Musa-Okumoto model and the standard Musa-Okumoto model

$$R_1 = 7.817,\ \ R_2 = 1.924,\ \ R_3 = 1.905,$$

$$R_1^* = 7.445,\ \ R_2^* = 2.222,\ \ R_3^* = 1.800.$$

- After predicting 6 numbers of failures (from the 3-rd test till 8-th test)

$$R_1 = 1.932,\ \ R_2 = 0.990,\ \ R_3 = 0.620,$$

$$R_1^* = 3.060,\ \ R_2^* = 2.005,\ \ R_3^* = 1.020.$$

Open problems and ideas (1)

1. The lack of independence of times to failures:

$$\max_{\mathcal{M}} \Pr\left\{X_1 = k_1, ..., X_n = k_n\right\} = \min_{i=1,...,n} \left\{\overline{F}_i(k_i) - \underline{F}_i(k_i - 1)\right\}.$$

How to realize Bayesian approach in this case?

Open problems and ideas (1)

1. The lack of independence of times to failures:

$$\max_{\mathcal{M}} \Pr\left\{X_1 = k_1, ..., X_n = k_n\right\} = \min_{i=1,...,n} \left\{\overline{F}_i(k_i) - \underline{F}_i(k_i - 1)\right\}.$$

How to realize Bayesian approach in this case?

2. **The metod of generalized moments in place of the imprecise Bayesian approach.** The probability $\Pr\left\{X_1 = k_1, ..., X_n = k_n\right\}$ can be found by using the natural extension with sample moments (or sample generalized moments) as the initial previsions. The number of moments defines the imprecision like the parameter s.

Open problems and ideas (2)

3 **Hypothesis**: k moments produce a set $\mathcal{M}$ of probability mass functions such that

$$\underline{F}_i(k \mid \mathbf{d}) = \inf_{F(k)\in\mathcal{M}_i(\mathbf{d})} F(k),\ \overline{F}_i(k \mid \mathbf{d}) = \sup_{F(k)\in\mathcal{M}_i(\mathbf{d})} F(k).$$

Suppose that $\pi^*(k) = \max_{\mathcal{M}} \Pr\{X = k\}$. Is it true that

$$\pi^*(k) = \overline{F}(k) - \underline{F}(k-1)?$$

It is true for two first moments!

Questions

?