불완전 디버깅 환경에서 Input Domain에 기초한 소프트웨어 신뢰성 성장 모델

박 중 양[†] · 김 영 순[††] · 황 양 숙[††]

요 약

Park, Seo and Kim[12]은 소프트웨어의 시험단계와 유지보수단계에 모두 적용할 수 있는 입력 영역 기반 소프트웨어 신뢰성 성장 모델을 개발하였다. 이들의 모형은 완전디버깅의 가정 하에서 개발되어졌다. 입력 영역 기반 소프트웨어 신뢰성 성장 모델이 현실적이기 위해서는 이러한 가정은 개선되어야 한다. 본 논문에서는 불완전 디버깅 하에서 사용할 수 있는 입력 영역 기반 소프트웨어 신뢰성 성장 모델을 제안하고 그 통계적 특성을 조사한다.

An Input Domain-Based Software Reliability Growth Model In Imperfect Debugging Environment

Joong-Yang Park[†] · Young-Soon Kim[††] · Yang-Sook Hwang[††]

ABSTRACT

Park, Seo and Kim [12] developed the input domain-based SRGM, which was able to quantitatively assess the reliability of a software system during the testing and operational phases. They assumed perfect debugging during testing and debugging phase. To make this input domain-based SRGM more realistic, this assumption should be relaxed. In this paper we generalize the input domain-based SRGM under imperfect debugging. Then its statistical characteristics are investigated.

키워드 : 입력영역(Input Domain), 소프트웨어 신뢰성 성장 모델(Software Reliability Growth Model), 불완전 디버깅(Imperfect Debugging) 다항 분포(Multinomial Distribution), 최대 우도법(Maximum Likelihood Method), 수학적 귀납법(Mathematical Indunction)

1. Introductions

In order to evaluate a software system, there are many attributes of software quality. Software reliability, however, is generally accepted as the key factor in software quality since it quantifies software failures. Software reliability has been defined as the probability that no failure occurs in a specified environment during a specified exposure period. The time unit of exposure period depends on the type of reliability model used. For the evaluation of the reliability of a software system during the testing and operational phases, many software reliability growth models (SRGMs) have been proposed in the literature. See the review papers such as Brown and Lipow [1], Duran and Ntafos [6], Goel[7], MacWilliams [9], Nelson [11], Ramamoorthy and Bastani [16] and Weiss and Weyuker [17], etc. The SRGMs are usually used to estimate the number of remaining faults, software reliability and other software quality assessment measures. Park, Lee and Park [13, 14] recently suggested several neural network models for predicting software reliability. Most of software reliability growth models (SRGMs) have been developed for evaluating software reliability growth behavior by analyzing the failure data obtained during testing software systems. Several researches [2, 4, 5, 8] indicate drawbacks of SRGMs : The usual assumptions made for SRGMs are still questionable ; SRGMs do not sufficiently account for the characteristics of the software systems under testing ; SRGMs do not work well for the software systems which rarely fail during testing. This problem can be overcome by considering the input domain-based models which do not require assumptions on the software failure and cor-

† 정 회 원 : 경상대학교 통계학과 교수
†† 준 회 원 : 경상대학교 대학원 정보처리학과
논문접수 : 2002년 3월 13일, 심사완료 : 2002년 5월 24일

rection processes. The input domain-based models usually classify into the class of reliability models for the validation phase. However, the input domain-based models can also be applied in the testing and debugging phase by treating the software system after each fault correction as a new software system.

Input domain-based models generally require large number of test inputs for obtaining accurate reliability estimates. Several studies have been proposed a method of reduction the number of required test inputs by taking advantage of information obtained previously. For example, Podgurski and Weyuker [15] and Dasu and Weyuker [3] proposed economical approaches for estimating reliabilities of successive software versions resulting from the maintenance of software. They proposed a heuristic algorithm for updating the previous estimate of reliability. Providing the heuristic of Podgurski and Weyuker [15] with statistical justification, Park, Seo and Kim [12] suggested an input domain-based SRGM, which did not require assumptions about software development and usage environment. And it described the reliability growth behavior of a software system, is based on a multistage testing procedure, in which fault corrections occur after each stage and two software systems, the software system before fault correction and the software system after fault corrections, are tested at each stage. But Park, et al [12] developed input domain-based SRGM under perfect debugging. It is necessary to develop input domain-based SRGM which assume imperfect debugging because the faults detected by testing are not always corrected/removed. Section 2 presents assumptions and testing procedure. The input domain-based SRGM under perfect debugging is briefly described in Section 3. In Section 4, we generalize the input domain-based SRGM underlying imperfect debugging and investigate its statistical characteristics.

2. Assumptions and Testing Procedure

Park, Seo and Kim [12] developed the input domain based SRGM under the following assumptions :

(1) Input domain of the software system is defined and will not change.
(2) Test inputs are randomly selected from operational profile, which will not change.
(3) Debugging is perfect.
(4) The cost of executing test inputs is small relative to the cost of checking whether an execution conforms to requirements.
(5) Reliability is a function of the number of remaining faults.

We will adopt these assumptions except for assumption (3) and change it into imperfect debugging. Next we consider the testing procedure. A test run is an execution of the software system by applying an input selected according to the given testing profile. The term "test stage" refers to some predetermined number of consecutive test runs. The testing is performed stage by stage. If failures occur during a testing stage, testing will not be interrupted for fault corrections. Fault corrections occur at the end of each testing stage. We denote by P_i the software system after fault corrections of $(i-1)$ st testing stage (equivalently at the beginning of i th testing stage). The failure probability of P_i is denoted by θ_i. Moreover we denote the number of test inputs for ith testing stage by n_i. The existing input domain-based reliability models treat the software system after each test stage as a new software system. Thus we estimate θ_i from the failure data obtained during ith testing stage. Two software systems P_{i-1} and P_i are identical except for the part debugged after $(i-1)$ st testing stage. The debugged part is likely to be a small portion of the software system. This implies that the failure data obtained before i th testing stage contains information on the undebugged part of P_i. It is thus desirable to develop a testing procedure through which relationship among θ_i's can be derived. We thus propose the testing procedure in which n_i test inputs are applied to both P_{i-1} and P_i for $i \geq 2$. One exception is that only P_1 is executed at 1st testing stage.

3. Review Of Input Domain-Based SRGM

By considering testing procedure suggested in Section 2, Park, et al [12] proposed the following formulation of the problem. Execution of each input in 1st testing stage results in one of the two outcomes, success (S) and failure (F). The number of failures occurred in 1st testing stage be denoted by x_{1F}. In case of i th testing stage, both P_{i-1} an P_i be exercised, for $i \geq 2$, each test run results in one of the five outcomes, *SS*, *SF*, *FS*, *FFD* and *FFS* of which brief descriptions are presented in <Table 1>.

〈Table 1〉 Brief descriptions of outcomes of each test run

outcome	description
SS	both P_{i-1} and P_i succeed
SF	P_{i-1} succeeds and P_i fails
FS	P_{i-1} fails and P_i succeeds
FFD	P_{i-1} and P_i both fail and produce different outputs
FFS	P_{i-1} and P_i both fail and produce same outputs

They denoted the number of test runs and the occurrence probability corresponding to each outcome by $x_{i\,\text{outcome}}$ and $p_{i\,\text{outcome}}$. Then x_{1F} follows a binomial distribution with parameters n_1 and θ_1 and the joint distribution of x_{iSS}, x_{iSF}, x_{iFS}, x_{iFFD} and x_{iFFS} for $i \geq 2$ be multinomial distribution. But under the perfect debugging, only three outcomes, *SS*, *FS* and *FFS*, can occur. Thus the joint distribution of x_{iSS}, x_{iFS} and x_{iFFS} are given by

$$f(x_{iSS}, x_{iFS}, x_{iFFS}) = \binom{n_i}{x_{iSS}, x_{iFS}, x_{iFFS}} p_{iSS}^{x_{iSS}} p_{iFS}^{x_{iFS}} p_{iFFS}^{x_{iFFS}}, \quad (1)$$

where $x_{iSS}+x_{iFS}+x_{iFFS}=n_i$ and $p_{iSS}+p_{iFS}+p_{iFFS}=1$. In order to represent Expression (1) in terms of θ_i's, Park, et al [12] defined θ_i's to be

$$\begin{aligned}\theta_{i-1} &= \Pr(P_{i-1} \text{ fails}) \\ &= \Pr(P_{i-1} \text{ fails and } P_i \text{ succeeds}) \\ &\quad + \Pr(P_{i-1} \text{ and } P_i \text{ both fail}) \\ &= \Pr(P_{i-1} \text{ fails and } P_i \text{ succeeds}) \\ &\quad + \Pr(P_{i-1} \text{ and } P_i \text{ both fail and produce same outputs}) \\ &\quad + \Pr(P_{i-1} \text{ and } P_i \text{ both fail and produce different outputs}) \\ &= p_{iFS} + p_{iFFS} + p_{iFFD} \qquad (2)\end{aligned}$$

and

$$\begin{aligned}\theta_i &= \Pr(P_i \text{ fails}) \\ &= \Pr(P_{i-1} \text{ succeeds and } P_i \text{ fails}) \\ &\quad + \Pr(P_{i-1} \text{ and } P_i \text{ both fail}) \\ &= \Pr(P_{i-1} \text{ succeeds and } P_i \text{ fails}) \\ &\quad + \Pr(P_{i-1} \text{ and } P_i \text{ both fail and produce same outputs}) \\ &\quad + \Pr(P_{i-1} \text{ and } P_i \text{ bloth fail and produce different outputs}) \\ &= p_{iFS} + p_{iFFS} + p_{iFFD}. \qquad (3)\end{aligned}$$

Since $p_{iSF} = p_{iFFD} = 0$ under the perfect debugging assumption, Expressions (2) and (3) are simplified to $\theta_{i-1} = p_{iFS} + p_{iFFS}$ and $\theta_i = p_{iFFS}$. The probabilities p_{iFS} and p_{iFFS} are then expressed as $p_{iFS} = \theta_{i-1} - \theta_i$ and $p_{iFFS} = 1 - p_{iSS} - \theta_{i-1} + \theta_i$. Substituting these into Expression (1), the joint distribution of x_{iSS}, x_{iFS} and x_{iFFS} can be rewritten as

$$f(x_{iSS}, x_{iFS}, x_{iFFS}) = \binom{n_i}{x_{iSS}, x_{iFS}, x_{iFFS}} p_{iSS}^{x_{iSS}} (\theta_{i-1} - \theta_i)^{x_{iFS}} (1 - p_{iSS} - \theta_{i-1} + \theta_i)^{x_{iFFS}}. \quad (4)$$

By maximum likelihood method, $\hat{\theta}_1$ be x_{1F}/n_1, for $i=1$. If $i \geq 2$, then

$$\hat{\theta}_i = \widehat{\theta_{i-1}} - \frac{x_{iFS}}{n_i} = \frac{x_{1F}}{n_1} - \sum_{k=2}^{i} \frac{x_{kFS}}{n_k}. \quad (5)$$

$\hat{\theta}_i$ is an unbiased for θ_i and the variance of this estimator is given by :

$$\begin{aligned}Var(\hat{\theta}_i) &= Var(\hat{\theta}_{i-1}) + \frac{(\theta_{i-1} - \theta_i)(1 - \theta_{i-1} + \theta_i)}{n_i} \\ &= \frac{1}{n_1}\theta_1(1-\theta_1) + \sum_{j=2}^{i} \frac{(\theta_{j-1} - \theta_j)(1 - \theta_{j-1} + \theta_j)}{n_j}.\end{aligned}$$

Thus $\hat{\theta}_i$ is a consistent estimator of θ_i and it is clear that failure probability after fault corrections has a larger variance than previous estimator. In next section, we propose an estimator under imperfect debugging.

4. Generalization of Input Domain-Based SRGM

We propose a more realistic and general estimator for θ_i within the same framework and constraints except for the perfect debugging discussed in the previous section. The detected faults by debugging are not always corrected/removed. And once a failure occurs, the corresponding fault corrections may introduce new faults. Thus we assume the imperfect debugging. Under this assumption, the joint distribution of x_{iSS}, x_{iSF}, x_{iFS}, x_{iFFD} and x_{iFFS} is given by the following multinomial distribution :

$$\begin{aligned}f(x_{iSS}, x_{iSF}, x_{iFS}, x_{iFFD}, x_{iFFS}) &= \binom{n_i}{x_{iSS}, x_{iSF}, x_{iFS}, x_{iFFD}, x_{iFFS}} \quad (6) \\ &\quad p_{iSS}^{x_{iSS}} p_{iSF}^{x_{iSF}} p_{iFS}^{x_{iFS}} p_{iFFD}^{x_{iFFD}} p_{iFFS}^{x_{iFFS}},\end{aligned}$$

where

$$\binom{n_i}{x_{iSS},\ x_{iSF},\ x_{iFS},\ x_{iFFD},\ x_{iFFS}} = \frac{n_i!}{x_{iSS}!x_{iSF}!x_{iFS}!\ x_{iFFD}!\ x_{iFFS}!},$$

$x_{iSS}+x_{iSF}+x_{iFS}+x_{iFFD}+x_{iFFS}=n_i$ and $p_{iSS}+p_{iSF}+p_{iFS}+p_{iFFD}+p_{iFFS}=1$. In expression (2) and (3), the probabilities p_{iFS} and p_{iFFS} are then expressed as $p_{iFS}=\theta_{i-1}-\theta_i+p_{iSF}$ and $p_{iFFS}=1-p_{iSS}-\theta_{i-1}+\theta_i-2p_{2SF}-p_{2FFD}$. For representing Expression (6) in terms of θ_i's, substituting these into Expression (6). Then the joint distribution of x_{iSS}, x_{iSF}, x_{iFS}, x_{iFFD} and x_{iFFS} can be rewritten as

$$\begin{aligned} f(x_{iSS}, x_{iSF}, x_{iFS}, x_{iFFD}, x_{iFFS}) &= \binom{n_i}{x_{iSS},\ x_{iSF},\ x_{iFS},\ x_{iFFD},\ x_{iFFS}} \\ &\quad p_{iSS}^{x_{iSS}}p_{iSF}^{x_{iSF}}(\theta_{i-1}-\theta_i+p_{iSF})^{x_{iFS}} \\ &\quad p_{iFFD}^{x_{iFFD}}(1-p_{iSS}-\theta_{i-1}+\theta_i-2p_{iSF}-p_{iFFD})^{x_{iFFS}}. \end{aligned}$$

After completing i testing stage, we have to estimate parameters θ_j, p_{jSS}, p_{jSF} and p_{jFFD} for $j\le i$, among which the most interesting parameter is θ_i. The maximum likelihood estimates (MLEs) of the parameters will be derived in this paper by maximizing the likelihood function L given as

$$L=\begin{cases} \binom{n_1}{x_1}\theta_1^{x_{1F}}(1-\theta_1)^{n_1-x_{1F}}, \quad \text{for } i=1 \\ \binom{n_1}{x_1}\theta_1^{x_{1F}}(1-\theta_1)^{n_1-x_{1F}}\prod_{j=2}^{i}\binom{n_j}{x_{jSS},\ x_{jSF},\ x_{jFS},\ x_{jFFD},\ x_{jFFS}} \\ \quad p_{jSS}^{x_{jSS}}p_{jSF}^{x_{jSF}}(\theta_{j-1}-\theta_j+p_{jSF})^{x_{jFS}} \\ \quad p_{jFFD}^{x_{jFFD}}(1-p_{jSS}-\theta_{j-1}+\theta_j-2p_{jSF}-p_{jFFD})^{x_{jFFS}}. \end{cases}$$

MLEs are usually obtained by maximizing the log likelihood function ln L with respect to θ_j, p_{jSS}, p_{jSF} and p_{jFFD} for $j\le i$. For the case where $i=1$, it can be easily shown that $\hat{\theta}_1=x_{1F}/n_1$. If $i\ge 2$, we have to estimate θ_j, p_{jSS}, p_{jSF} and p_{jFFD} for $j\le i$. Let us denote MLEs of θ_j, p_{jSS}, p_{jSF} and p_{jFFD} obtained after i th stage by $\hat{\theta}_{j,i}$, $\hat{p}_{jSS,i}$, $\hat{p}_{jSF,i}$ and $\hat{p}_{jFFD}$. Accordingly $\hat{\theta}_1=x_{1F}/n_1$ obtained after 1st stage is denoted by $\hat{\theta}_{1,1}$. We will show by mathematical induction that

$$\hat{\theta}_{j,i}=\hat{\theta}_{j,(i-1)},\quad \hat{p}_{jSS,i}=\hat{p}_{jSS,(i-1)},\quad \hat{p}_{jSF,i}=\hat{p}_{jSF,(i-1)}$$
$$\text{and } \hat{p}_{jFFD,i}=\hat{p}_{jFFD,(i-1)} \text{ for } j\le(i-1),$$
$$\hat{\theta}_{i,i}=\frac{x_{1F}}{n_1}+\sum_{j=2}^{i}\left(\frac{x_{jSF}}{n_j}-\frac{x_{jFS}}{n_j}\right),\ \hat{p}_{iSS,i}=\frac{x_{iSS}}{n_i},$$
$$\hat{p}_{iSF,i}=\frac{x_{iSF}}{n_i} \text{ and } \hat{p}_{iFFD,i}=\frac{x_{iFFD}}{n_i}. \tag{7}$$

When $i=2$, the likelihood equations are obtained as

$$\begin{aligned} \frac{\partial \ln L}{\partial\theta_1} &= \frac{x_{1F}}{\theta_1}-\frac{n_1-x_{1F}}{1-\theta_1}+\frac{x_{2FS}}{\theta_1-\theta_2+p_{2SF}} \\ &\quad -\frac{x_{2FFS}}{1-p_{2SS}-\theta_1+\theta_2-2p_{2SF}-p_{2FFD}}=0 \\ \frac{\partial \ln L}{\partial p_{2SS}} &= \frac{x_{2SS}}{p_{2SS}} \\ &\quad -\frac{x_{2FFS}}{1-p_{2SS}-\theta_1+\theta_2-2p_{2SF}-p_{2FFD}}=0 \\ \frac{\partial \ln L}{\partial p_{2SF}} &= \frac{x_{2SF}}{p_{2SF}}+\frac{x_{2FS}}{\theta_1-\theta_2+p_{2SF}} \\ &\quad -\frac{2x_{2FFS}}{1-p_{2SS}-\theta_1+\theta_2-2p_{2SF}-p_{2FFD}}=0 \\ \frac{\partial \ln L}{\partial p_{2FFD}} &= \frac{x_{2FFD}}{p_{2FFD}} \\ &\quad -\frac{x_{2FFS}}{1-p_{2SS}-\theta_1+\theta_2-2p_{2SF}-p_{2FFD}}=0 \\ \frac{\partial \ln L}{\partial\theta_2} &= -\frac{x_{2FS}}{\theta_1-\theta_2+p_{2SF}} \\ &\quad +\frac{x_{2FFS}}{1-p_{2SS}-\theta_1+\theta_2-2p_{2SF}-p_{2FFD}}=0. \end{aligned} \tag{8}$$

Substituting $\partial\ln L/\partial\theta_2=0$ into $\partial\ln L/\partial\theta_1=0$, the $\partial\ln L/\partial\theta_1=0$ becomes the same with $\partial\ln L/\partial\theta_1=0$ for 1 st stage. Solving $\partial\ln L/\partial p_{2SS}=0$, $\partial\ln L/\partial p_{2SF}=0$, $\partial\ln L/\partial p_{2FFD}=0$ and $\partial\ln L/\partial\theta_2=0$ simultaneously with θ_1 replaced by $\hat{\theta}_{1,1}$, we can verify that $\partial\ln L/\partial p_{2SS}=0$ and $\partial\ln L/\partial\theta_2=0$ hold only when $\theta_2=x_1F/n_1+x_{2SF}/n_2-x_{2FS}/n_2$, $p_{2SF}=x_{2SF}/n_{2SF}$ and $p_{2SS}=x_{2SS}/n_2$. Therefore

$$\hat{\theta}_{1,2}=\hat{\theta}_{1,1},\quad \hat{\theta}_{2,2}=\frac{x_{1F}}{n_1}+\frac{x_{2SF}}{n_2}-\frac{x_{2FS}}{n_2},$$
$$\hat{p}_{2SS,2}=\frac{x_{2SS}}{n_2},\quad \hat{p}_{2SF,2}=\frac{x_{2SF}}{n_2} \text{ and } \hat{p}_{2FFD,2}=\frac{x_{2FFD}}{n_2}$$

are the unique solution of the likelihood equations (8), i.e., MLEs of θ_1, θ_2, p_{2SS}, p_{2SF} and p_{2FFD}. This is the desired result. The likelihood equations for $(i-1)$st stage are given by

$$\frac{\partial \ln L}{\partial \theta_1} = \frac{x_1}{\theta_1} - \frac{n_1 - x_1}{1-\theta_1} + \frac{x_{2FS}}{\theta_1 - \theta_2 + p_{2SF}} - \frac{x_{2FFS}}{1 - p_{2SS} - \theta_1 + \theta_2 - 2p_{2SF} - p_{2FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{2SS}} = \frac{x_{2SS}}{p_{2SS}} - \frac{x_{2FFS}}{1 - p_{2SS} - \theta_1 + \theta_2 - 2p_{2SF} - p_{2FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{2SF}} = \frac{x_{2SF}}{p_{2SF}} + \frac{x_{2FS}}{\theta_1 - \theta_2 + p_{2SF}} - \frac{2x_{2FFS}}{1 - p_{2SS} - \theta_1 + \theta_2 - 2p_{2SF} - p_{2FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{2FFD}} = \frac{x_{2FFD}}{p_{2FFD}} - \frac{x_{2FFS}}{1 - p_{2SS} - \theta_1 + \theta_2 - 2p_{2SF} - p_{2FFD}} = 0$$

$$\frac{\partial \ln L}{\partial \theta_2} = -\frac{x_{2FS}}{\theta_1 - \theta_2 + p_{2SF}} + \frac{x_{2FFS}}{1 - p_{2SS} - \theta_1 + \theta_2 - 2p_{2SF} - p_{2FFD}} + \frac{x_{3FS}}{\theta_2 - \theta_3 + p_{3SF}} - \frac{x_{3FFS}}{1 - p_{3SS} - \theta_2 + \theta_3 - 2p_{3SF} - p_{3FFD}} = 0$$

$$\vdots$$

$$\frac{\partial \ln L}{\partial p_{(i-2)SS}} = \frac{x_{(i-2)SS}}{p_{(i-2)SS}} - \frac{x_{(i-2)FFS}}{1 - p_{(i-2)SS} - \theta_{(i-3)} + \theta_{(i-2)} - 2p_{(i-2)SF} - p_{(i-2)FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{(i-2)SF}} = \frac{x_{(i-2)SF}}{p_{(i-2)SF}} + \frac{x_{(i-2)FS}}{\theta_{(i-3)} - \theta_{(i-2)} + p_{(i-2)SF}} - \frac{2x_{(i-2)FFS}}{1 - p_{(i-2)SS} - \theta_{(i-3)} + \theta_{(i-2)} - 2p_{(i-2)SF} - p_{(i-2)FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{(i-2)FFD}} = \frac{x_{(i-2)FFD}}{p_{(i-2)FFD}} - \frac{x_{(i-2)FFS}}{1 - p_{(i-2)SS} - \theta_{(i-3)} + \theta_{(i-2)} - 2p_{(i-2)SF} - p_{(i-2)FFD}} = 0$$

$$\frac{\partial \ln L}{\partial \theta_{(i-2)}} = -\frac{x_{(i-2)FS}}{\theta_{(i-3)} - \theta_{(i-2)} + p_{(i-2)SF}} + \frac{x_{(i-2)FFS}}{1 - p_{(i-2)SS} - \theta_{(i-3)} + \theta_{(i-2)} - 2p_{(i-2)SF} - p_{(i-2)FFD}} + \frac{x_{(i-1)FS}}{\theta_{(i-2)} - \theta_{(i-1)} + p_{(i-1)SF}} - \frac{x_{(i-1)FFS}}{1 - p_{(i-1)SS} - \theta_{(i-2)} + \theta_{(i-1)} - 2p_{(i-1)SF} - p_{(i-1)FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{(i-1)SS}} = \frac{x_{(i-1)SS}}{p_{(i-1)SS}} - \frac{x_{(i-1)FFS}}{1 - p_{(i-1)SS} - \theta_{(i-2)} + \theta_{(i-1)} - 2p_{(i-1)SF} - p_{(i-1)FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{(i-1)SF}} = \frac{x_{(i-1)SF}}{p_{(i-1)SF}} + \frac{x_{(i-1)FS}}{\theta_{(i-2)} - \theta_{(i-1)} + p_{(i-1)SF}} - \frac{2x_{(i-1)FFS}}{1 - p_{(i-1)SS} - \theta_{(i-2)} + \theta_{(i-1)} - 2p_{(i-1)SF} - p_{(i-1)FFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{(i-1)FFD}} = \frac{x_{(i-1)FFD}}{p_{(i-1)FFD}} - \frac{x_{(i-1)FFS}}{1 - p_{(i-1)SS} - \theta_{(i-2)} + \theta_{(i-1)} - 2p_{(i-1)SF} - p_{(i-1)FFD}} = 0$$

$$\frac{\partial \ln L}{\partial \theta_{(i-1)}} = -\frac{x_{(i-1)FS}}{\theta_{(i-2)} - \theta_{(i-1)} + p_{(i-1)SF}} + \frac{x_{(i-1)FFS}}{1 - p_{(i-1)SS} - \theta_{(i-2)} + \theta_{(i-1)} - 2p_{(i-1)SF} - p_{(i-1)FFD}} = 0 \tag{9}$$

Suppose that the unique solution for the above likelihood equations is given by

$\hat{\theta}_{j,(i-1)} = \hat{\theta}_{j,(i-2)}$, $\hat{p}_{jSS,(i-1)} = \hat{p}_{jSS,(i-2)}$, $\hat{p}_{jSF,(i-1)} = \hat{p}_{jSF,(i-2)}$ and $\hat{p}_{jFFD,(i-1)} = \hat{p}_{jFFD,(i-2)}$ for $j \le (i-2)$ and $j \ge 2$. $\quad \hat{\theta}_{(i-1),(i-1)} = \frac{x_{1F}}{n_1} + \sum_{j=2}^{i-1}\left(\frac{x_{jSF}}{n_j} - \frac{x_{jFS}}{n_j}\right)$,

$\hat{p}_{(i-1)SS,(i-1)} = \frac{x_{(i-1)SS}}{n_{i-1}}$, $\hat{p}_{(i-1)SF,(i-1)} = \frac{x_{(i-1)SF}}{n_{i-1}}$ and

$\hat{p}_{(i-1)FFD,(i-1)} = \frac{x_{(i-1)FFD}}{n_{i-1}}$,

that is, Expression (7) holds. Next consider the likelihood equations for ith stage, which consist of the likelihood equations for $(i-1)$ st stage with one modification and the following four additional equations.

$$\frac{\partial \ln L}{\partial p_{iSS}} = \frac{x_{iSS}}{p_{iSS}} - \frac{x_{iFFS}}{1 - p_{iSS} - \theta_{(i-1)} + \theta_i - 2p_{iSF} - p_{iFFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{iSF}} = \frac{x_{iSF}}{p_{iSF}} + \frac{x_{iFS}}{\theta_{(i-1)} - \theta_i + p_{iSF}} - \frac{2x_{iFFS}}{1 - p_{iSS} - \theta_{(i-1)} + \theta_i - 2p_{iSF} - p_{iFFD}} = 0$$

$$\frac{\partial \ln L}{\partial p_{iFFD}} = \frac{x_{iFFD}}{p_{iFFD}} - \frac{x_{iFFS}}{1 - p_{iSS} - \theta_{(i-1)} + \theta_i - 2p_{iSF} - p_{iFFD}} = 0$$

$$\frac{\partial \ln L}{\partial \theta_i} = -\frac{x_{iFS}}{\theta_{(i-1)} - \theta_i + p_{iSF}} + \frac{x_{iFFS}}{1 - p_{iSS} - \theta_{(i-1)} + \theta_i - 2p_{iSF} - p_{iFFD}} = 0 \tag{10}$$

One modification is that $\partial \ln L / \partial \theta_{i-1} = 0$ for $(i-1)$ st stage is changed to

$$\frac{\partial \ln L}{\partial \theta_{(i-1)}} = -\frac{(x_{(i-1)FS})}{\theta_{(i-2)} - \theta_{(i-1)} + p_{(i-1)SF}} + \frac{x_{(i-1)FFS}}{1 - p_{(i-1)SS} - \theta_{(i-2)} + \theta_{(i-1)} - 2p_{(i-1)SF} - p_{(i-1)FFD}} + \frac{x_{iFS}}{\theta_{(i-1)} - \theta_i + p_{iSF}} - \frac{x_{iFFS}}{1 - p_{iSS} - \theta_{(i-1)} + \theta_i - 2p_{iSF} - p_{iFFD}} = 0.$$

However, if we substitute $\partial \ln L / \partial \theta_i = 0$ into $\partial \ln L / \partial \theta_{i-1} = 0$, $\partial \ln L / \partial \theta_{i-1} = 0$ for ith stage becomes identical to $\partial \ln L / \partial \theta_{i-1} = 0$ for $(i-1)$st stage. Therefore the likelihood equations for ith stage are composed of the likelihood equations for $(i-1)$st stage and four additional equations given by Equations (10). Letting $\hat{\theta}_{j,i} = \hat{\theta}_{j,(i-1)}$, $\hat{p}_{jSS,i} = \hat{p}_{jSS,(i-1)}$, $\hat{p}_{jSF,i} = \hat{p}_{jSF,(i-1)}$and $\hat{p}_{jFFD,i} = \hat{p}_{jFFD,(i-1)}$ for $j \le (i-1)$ and substituting them into Equations (10), it is easily shown that Equations (10) are satisfied only when $p_{iSS} = x_{iSS}/n_i$, $p_{iSF} = x_{iSF}/n_i$, $p_{iFFD} = x_{iFFD}/n_i$ and $\theta_i = (n_{1F}/n_1) + \sum_{j=2}^{i}(x_{jSF}/n_j - x_{jFS}/n_j)$. Therefore, the estimates given in Expression (7) are the unique solution of the likelihood equations for ith stage. Now proof is completed.

One noteworthy point is that the estimates obtained in the previous stages do not change as the testing proceeds. Thus we can simply rename $\hat{\theta}_{j,i}$, $\hat{p}_{jSS,i}$, $\hat{p}_{jSF,i}$ and $\hat{p}_{jFFD,i}$ as $\hat{\theta}_j$, $\hat{p}_{jSS}$, $\hat{p}_{jSF}$ and $\hat{p}_{jFFD}$ where $\hat{\theta}_1 = x_{1F}/n_1$, $\hat{\theta}_j = n_{1F}/n_1 + \sum_{k=2}^{j}(x_{kSF}/n_k - x_{kFS}/n_k)$, $\widehat{p_{jSS}} = n_{jSS}/n_j$, $\widehat{p_{jSF}} = n_{jSF}/n_j$ and $\widehat{p_{jFFD}} = n_{jFFD}/n_j$. We can further derive a useful relationship

$$\hat{\theta}_i = \hat{\theta}_{i-1} + \frac{x_{iSF}}{n_i} - \frac{x_{iFS}}{n_i} \quad \text{for } i \ge 2, \tag{11}$$

which enables us to obtain the new estimate of the failure probability by updating the estimate obtained in the previous testing stage. In order to characterize MLEs statistically, we compute the expected value and variance of $\hat{\theta}_i$. Since

$$\begin{aligned} E(\hat{\theta}_i) &= E\left(\hat{\theta}_1 + \sum_{j=2}^{i}\left(\frac{x_{jSF}}{n_j} - \frac{x_{jFS}}{n_j}\right)\right) \\ &= \frac{1}{n_1} n_1\theta_1 + \sum_{j=2}^{i}\left(\frac{1}{n_j} n_j p_{jSF} - \frac{1}{n_j} n_j p_{jFS}\right) \\ &= E\left(\frac{x_{1F}}{n_1}\right) + \sum_{j=2}^{i}\left(E\left(\frac{x_{jSF}}{n_j}\right) - E\left(\frac{x_{jFS}}{n_j}\right)\right) \\ &= \theta_1 + \sum_{j=2}^{i}(p_{jSF} - p_{jFS}) \\ &= \theta_1 + \sum_{j=2}^{i}(p_{jSF} - (\theta_{j-1} - \theta_j + p_{jSF})) \\ &= \theta_i. \end{aligned} \tag{12}$$

$\hat{\theta}_i$ is an unbiased estimator for θ_i. And the variance of this estimate is given by

$$\begin{aligned} &Var(\hat{\theta}_i) \\ &= V\left(\hat{\theta}_1 + \sum_{j=2}^{i}\left(\frac{x_{jSF}}{n_j} - \frac{x_{jFS}}{n_j}\right)\right) \\ &= V(\hat{\theta}_1) + \sum_{j=2}^{i}\frac{1}{n_j^2}V(x_{jSF} - x_{jFS}) \\ &= \frac{1}{n_1}\theta_1(1-\theta_2) + \sum_{j=2}^{i}\frac{1}{n_j^2}(V(x_{jSF}) \\ &\quad + V(x_{jFS}) - 2COV(x_{jSF}, x_{jFS})) \\ &= \frac{1}{n_1}\theta_1(1-\theta_1) + \sum_{j=2}^{i}\frac{1}{n_j^2}(n_j p_{jSF}(1-p_{jSF}) \\ &\quad + n_j p_{jFS}(1-p_{jFS}) + 2n_j p_{jSF} p_{jFS}) \\ &= \frac{1}{n_1}\theta_1(1-\theta_1) + \sum_{j=2}^{i}\frac{2p_{jSF} + (\theta_{j-1} - \theta_j)(1 - \theta_{j-1} + \theta_j)}{n_j}, \end{aligned} \tag{13}$$

and

$$Var(\hat{\theta}_i) = Var(\hat{\theta}_{i-1}) + \frac{2p_{iSF} + (\theta_{i-1} - \theta_i)(1 - \theta_{i-1} + \theta_i)}{n_i}.$$

Thus

$$\widehat{Var(\hat{\theta}_i)} = \widehat{Var(\hat{\theta}_{i-1})} + 2\frac{x_{iSF}}{n_i} + \frac{(x_{iSF} - x_{iFS})(n_i - x_{iSF} + x_{iFS})}{n_i^3}. \tag{14}$$

An estimate of $Var(\hat{\theta}_i)$ is obtained by substituting θ_j's in Expression (13) with $\hat{\theta}_j$. Thus $\hat{\theta}_i$ is a consistent estimator of θ_i as n_i tend to infinity. Similarly we can show that $\hat{p}_{iSS}$, $\hat{p}_{iSF}$ and $\hat{p}_{iFFD}$ are also unbiased for p_{iSS}, p_{iSF} and p_{iFFD} and those variances are $p_{iSS}(1-p_{iSS})/n_i$, $p_{iSF}(1-p_{iSF})/n_i$ and $p_{iFFD}(1-p_{iFFD})/n_i$, respectively.

5. Conclusion

Debugging actions in real testing and operation environments are not always performed perfectly. For that reason, we have extended an input domain-based SRGM under imperfect debugging. We can construct an consistent and unbiased estimator that can usefully apply to real software testing. However, its practicability should be validated and examined through applying to real software testing. Furthermore, since the multi-stage testing procedure requires more testing time, the trade-off between testing time and number of required inputs should be investigated in order to determine the time to stop testing.

References

[1] J. R. Brown and M. Lipow, "Testing of Software Reliability," in Proc. Int. Conf. Reliable Software, Los Angeles, CA, pp.518-527, April, 1975.

[2] R. W. Butler and G. B. Finelli, "The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software," IEEE Trans. on Software Engineering, Vol.19, pp.3-12, 1993.

[3] T. Dasu and E. J. Weyuker, "Updating Software Reliability Subject to Resource Constraints," Proc. American Statistical Association Proc. Joint Statistical Meeting, 1999.

[4] T. Downs and P. Garrone, "Some New Models of Software Testing with Performance Comparisons," IEEE Trans. on Reliability, Vol.40, pp.322-328, 1991.

[5] J. R. Dunham, "Experiments in Software Reliability : Life-Critical Applications," IEEE Trans. on Software Engineering, Vol.SE-12, pp.110-123, 1986.

[6] J. W. Duran and S. C. Ntafos, "An Evaluation of Random Testing," IEEE Trans. Soft. Eng., SE-10, pp.438-444, July, 1984.

[7] A. L. Goel, "Software Reliability Models : Assumptions, Limitations, and Applicability," IEEE Trans. on Software Eng., pp.1411-1423, Dec., 1985.

[8] B. Littlewood, "Software Reliability Modelling : Achievements and Limitations," Proc. COMPEURO, Bologna, 1991.

[9] W. H. MacWilliams, "Reliability of Large Real-Time Software," in Proc. IEEE Symp. Computer Software Reliability, New York, pp.1-6, May, 1973.

[10] J. D. Musa, "Sensitivity of Field Failure Intensity to Operational Profile Errors," Proceedings of the 5th International Symposium on Software Reliability Engineering, Monterey, Calif., pp.334-337, November, 1994.

[11] E. Nelson, "Estimating software reliability from test data," Microelectronics Reliability, Vol.17, pp.67-74, 1978.

[12] J.-Y. Park, D.-W. Seo and Y.-S. Kim, "An Input Domain-Based Software Reliability Growth Model," Transactions of Korean Information Processing Society, Vol.7, No.11, pp. 3384-3393, 2000.

[13] J.-Y. Park, S.-U. Lee and J.-H. Park, "Neural Network Modeling for Software Reliability Prediction from Failure Time Data," Journal of Electrical Engineering and Information Science, Vol.4, pp.533-538, 1999.

[14] J.-Y. Park, S.-U. Lee and J.-H. Park, "Software Reliability Prediction Using Predictive Filter," Transactions of Korean Information Processing Society, Vol.7, No.7, pp.2076-2085, 2000.

[15] A. Podgurski and E. J. Weyuker, "Re-estimation of Software Reliability After Maintenance," IEEE Trans. on Rel., Vol.1, pp.79-85, 1998.

[16] C. V. Ramamoorthy and F. B. Bastani, "Software Reliability-Status and Perspectives," IEEE Trans. Soft. Eng., Vol. SE-8, No.4, pp.354-371, July, 1982.

[17] S. N. Weiss and E. J. Weyuker, "An Extended Domain-Based Model of Software Reliability," IEEE Trans. on Software Eng., Vol.SE-14, No.10, pp.456-470, 1988.

박 중 양

e-mail : parkjy@nongae.gsnu.ac.kr

1982년 연세대학교 응용통계학과 졸업 (학사)

1984년 한국과학기술원 산업공학과 (공학석사)

1994년 한국과학기술원 산업공학과 (공학박사)

1984년 ~ 1989년 경상대학교 전산통계학과 교수

1989년 ~ 현재 경상대학교 통계학과 교수

관심분야 : 소프트웨어 신뢰성, 신경망, 선형 통계 모형, 실험 계획법 등

김 영 순

e-mail : coaskim@hanmail.net

1994년 경상대학교 통계학과 졸업(학사)

1998년 경상대학교 통계학과(이학석사)

2000년~현재 경상대학교 정보처리학과 (박사과정)

관심분야 : 소프트웨어 신뢰성, 신경망, 선형 통계 모형, 실험계획법 등

황 양 숙

e-mail : hys0615@hotmail.com

1991년 경상대학교 사범대학 수학교육학과 졸업(학사)

2000년 경상대학교 대학원 통계학과 (이학석사)

2001년~현재 경상대학교 정보처리학과 (박사과정)

관심분야 : 소프트웨어 신뢰성, 신경망, 선형 통계 모형, 실험계획법 등