초기하분포 소프트웨어 신뢰성 성장 모델: 일반화, 추정과 예측

박 중 양*·유창열**·박재홍***

요 약

최근의 개발되어 성공적으로 적용되고 있는 초기하분포 소프트웨어 신뢰성 성장 모델은 이 모델에서 중요한 역할을 하는 반응계수(sensitivity factor)를 추정 대상인 모수로 가정하고 있다. 본 논문은 먼저 다형성과정의 도착법을 반영하기 위해 반응계수를 이형분포로 가정하여 초기하분포 소프트웨어 신뢰성 성장 모델을 일반화한다. 이러한 일반화는 초기하분포 소프트웨어 신뢰성 성장 모델의 통계적 특성을 더욱 완화할 수 있게 한다. 특히 일반화된 모델의 도착을 최소자승법으로 추정하면 기존 모델에 최소자승법을 적용한 것과 같은 결과를 얻을 수 있음을 보이고, 더불어 최종추정치를 최소자승법으로 구하는 방법과 예측방법도 제시한다.

Hyper-Geometric Distribution Software Reliability Growth Model: Generalization, Estimation and Prediction

Joong-Yang Park*·Chang-Yeul Yoo**·Jae-Heung Park***

ABSTRACT

The hyper-geometric distribution software reliability growth model (HGDM) was recently developed and successfully applied to real data sets. The HGDM considers the sensitivity factor as a parameter to be estimated. In order to reflect the random behavior of the test and debug process, this paper generalizes the HGDM by assuming that the sensitivity factor is a binomial random variable. Such a generalization enables us to easily understand the statistical characteristics of the HGDM. It is shown that the least squares method produces the identical results for both the HGDM and the generalized HGDM. Methods for computing the maximum likelihood estimates and predicting the future outcomes are also presented.

1. Introduction

In recent years software systems have been widely applied to many complex and critical systems. Since failure of a software system may result in serious damage, software systems are required to be very reliable. Therefore software reliability has become one of major issues in the software system development. In order to quantitatively assess 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 Goel[1], Ramamoorthy and
The SRGMs are usually used to estimate the number of remaining faults, software reliability, and software quality assessment measures. Some of currently available SRGMs enable us to predict probabilistically the time to next occurrence of failure in the operational phase. Another class of SRGMs let us to estimate the number of software faults still remaining after the debugging process. The HGDM advocated by Tohma et al.[14] belongs to the latter class of SRGMs. A series of studies on the HGDM has been made recently by Hou, Kuo, and Chang[2,13], Jacoby and Tohma[6], Miyahara and Tohma[9] and Tohma et al.[15]. Hou, Kuo, and Chang[4, 5] developed optimal software release policies based on the HGDM.

This paper first generalizes the HGDM to make it more realistic. The generalization is concerned with the sensitivity factor which is the key factor of the HGDM. Then parameter estimation and prediction problems are considered. Section 2 briefly reviews the basic concept and formulation of the HGDM. Assumptions on the sensitivity factor are generalized in Section 3 to reflect the random behavior of the test-and-debug process. Then the generalized HGDM is derived in Section 4. Section 5 considers the parameter estimation problem for the HGDM and the generalized HGDM. The least squares and maximum likelihood methods are dealt with. It is shown that the least squares estimates for both models are identical and that the maximum likelihood estimates can be computed by the least squares method. Section 6 suggests a method for predicting the number of faults newly discovered by future test operations. The method is based on the expected value of the number of newly discovered faults on condition that the cumulative number of faults is known.

2. Review of the HGDM

In this section we briefly review the HGDM. At the beginning of the test-and-debug process a software system is assumed to have \( m \) initial faults. Test operations performed in a day or a week may be called a test instance. Test instances are denoted by \( t_i, i=1,2,\cdots \) in accordance with the order of applying them. The sensitivity factor, \( w_i \), represents the number of faults discovered by the application of test instance \( t_i \). Some of the faults detected by \( t_i \) may have been detected previously by the application of test instances \( t_j, j=1,\cdots,i-1 \). The number of faults newly discovered by \( t_i \) is not necessarily equal to \( w_i \). That is, each detected fault can be classified into two categories, newly discovered faults and rediscovered faults. Let \( N_i \) denote the number of faults newly discovered by \( t_i \) and \( C_i = \sum_{j=1}^{i} N_j \). The following assumptions are made on the HGDM.

- No new faults are introduced into the software system during the debugging process.
- Sensitivity factor \( w_i \), the number of faults discovered by \( t_i \), is the faults taken randomly out of \( m \) initial faults.
- Sensitivity factor \( w_i \) is represented as a function of \( m \) and the progress in test-and-debugging \( p_i \), i.e., \( w_i = m p_i \).

\( p_i \) is usually referred to as the learning factor. The probability that \( x_i \) faults are newly discovered by \( t_i \) on condition that \( C_{i-1} \) faults has been discovered up to \( t_{i-1} \) is then formulated as

\[
P(N_i = x_i | C_{i-1}) = \frac{(m - C_{i-1}) \binom{C_{i-1}}{w_i} x_i \binom{m}{w_i}}{x_i}.
\]

where \( \max(0, w_i - C_{i-1}) \leq x_i \leq \min(w_i, m - C_{i-1}) \) for \( i = 1,2,\cdots \), \( C_0 = 0 \) and \( x_0 = 0 \). Thus the conditional expected value of \( N_i \) is

\[
E(N_i | C_{i-1}) = (m - C_{i-1}) p_i.
\]

The expected value of \( C_i \) was obtained by Jacoby and Tohma[7] as
The sensitivity factor is the key factor in the HGDM. Various functions for \( w_i \) have been devised and successfully applied to real data sets. Functional forms of the sensitivity factor are presented in Table 1 of Jacoby and Tohma[7] and Manohara and Tohma[9]. Recently Hou, Kou and Chang[2] introduced two types of sensitivity factor based on the learning curve. They are respectively referred to as the exponential sensitivity factor and the logistic sensitivity factor. Next section discusses further on the sensitivity factor.

3. Assumptions on Sensitivity Factor

As mentioned in the previous section, the HGDM assumes that sensitivity factor \( w_i \) is the \( w_i \) faults randomly chosen from \( m \) initial faults. Denoting by \( F_i \) the set of faults detected by \( t_i \), this assumption can be divided into two statements below.

- The size of \( F_i \) is \( w_i \), an unknown constant.
- The elements of \( F_i \) are randomly chosen from \( m \) initial faults.

We now argue that the first statement does not reflect enough the testing process. Test items* for a test instance are usually selected randomly from the input domain. Different sets of test items for a test instance would discover different number of faults. It is therefore more realistic to postulate that the number of faults detected by each test instance is a random variable. Next consider the learning factor \( p_i \), which represents the degree of test workers' skill at the application of test instance \( t_i \). Assuming that all \( m \) faults are detectable with equal probability, the learning factor can be practically regarded as the probability that test instance \( t_i \) discovers a fault. The sensitivity factor is thus assumed to be a binomial random variable with parameters \( m \) and \( p_i \), i.e., for \( w_i = 0, 1, \ldots, m \)

\[
P(W = w_i) = \binom{m}{w_i} p_i^w (1 - p_i)^{m - w_i}. \tag{3}
\]

4. A Generalized HGDM

This section generalizes the HGDM based on the following assumptions on the sensitivity factor. Other assumptions remain unchanged.

- Sensitivity factor \( W_i \) is distributed as Expression (3).
- Given that \( C_{i-1} \) faults have been detected up to test instance \( t_{i-1} \) and \( w_i \) faults are detected by test instance \( t_i \), the number of faults newly discovered by \( t_i \) is distributed as Expression (1).

We first derive the conditional distribution of \( N_i \) given that \( C_{i-1} \) faults have been discovered up to test instance \( t_{i-1} \). Multiplying Expressions (1) and (3),

\[
P(N_i = x_i, W_i = w_i | C_{i-1})
\]

\[
= \binom{m - C_{i-1}}{x_i} \binom{C_{i-1}}{w_i - x_i} p_i^w (1 - p_i)^{m - w_i}
\]

\[
= \binom{m - C_{i-1}}{x_i} p_i^w (1 - p_i)^{m - C_{i-1} - x_i}
\]

\[
= \binom{C_{i-1}}{w_i - x_i} p_i^{w_i - x_i} (1 - p_i)^{C_{i-1} - w_i - x_i}
\]

Therefore

\[
P(N_i = x_i | C_{i-1})
\]

\[
= \sum_{w} P(N_i = x_i, W_i = w_i | C_{i-1})
\]

\[
= \binom{m - C_{i-1}}{x_i} p_i^w (1 - p_i)^{m - C_{i-1} - x_i}. \tag{4}
\]

This is a binomial distribution with parameters \( m - C_{i-1} \) and \( p_i \). The joint distribution of \( N_i, i = 1, 2, \ldots, n \) is then obtained as

\[
P(N_i = x_i, i = 1, 2, \ldots, n)
\]

\[
= \prod_{i=1}^{n} P(N_i = x_i | N_{i-1} = x_{i-1}, j = 1, 2, \ldots, i-1)
\]

\[
= \prod_{i=1}^{n} P(N_i = x_i | C_{i-1})
\]
\[ \prod_{i=1}^{n} \left( \frac{m - \sum x_i}{x_i} \right) = \sum \quad (5) \]
\[ \left( x_1, \ldots, x_n \right) = \prod_{i=1}^{n} \left( 1 - p_i \right) \quad (6) \]

where
\[ \binom{m}{x_1, \ldots, x_n} = \frac{m!}{x_1! \cdots x_n! (m - \sum x_i)!} . \]

Since \( \sum_{i=1}^{n} \left( p_i \prod_{j=1}^{i} (1 - p_j) \right) = 1 - \prod_{j=1}^{n} (1 - p_j) \), the joint distribution of \( N_i \), \( i = 1, 2, \ldots, n \) is a multinomial distribution with parameters \( m \) and \( p_i \prod_{j=1}^{i} (1 - p_j) \), \( i = 1, \ldots, n \). Consequently, \( C_i \) is binomially distributed, i.e.,
\[ P(C_i = c_i) = \binom{m}{c_i} \left[ 1 - \prod_{j=1}^{c_i} (1 - p_j) \right]^{m-c_i} \prod_{j=1}^{c_i} (1 - p_j)^{m-c_i}. \]

5. Parameter Estimation for the Generalized HGDM

Let \( c_i \) and \( x_i \) be the observed values of \( C_i \) and \( N_i \). Suppose that the software system is tested up to test instance \( t_n \). In order to estimate the current number of residual faults and to predict the number of residual faults after applying \( t_{n+d} \) for \( d \geq 1 \), we first need to estimate the parameters in the model. Due to the mathematical difficulty of the maximum likelihood method, the least squares method has been used for the HGDM. Tohma et al.[14] obtained the least squares estimates by minimizing
\[ \sum_{i=1}^{n} \left( c_i - E(C_i) \right)^2 . \]

This criterion was also employed in Hau, Kuo and Chang[2][3] and Jacoby and Tohma[7]. However, Tohma et al.[15] minimized
\[ \sum_{i=1}^{n} \left( x_i - E(N_i | C_{i-1}) \right)^2 . \] (9)

The minimization of Expression (9) is equivalent to the minimization of
\[ \sum_{i=1}^{n} \left( c_i - E(C_i | C_{i-1}) \right)^2 . \] (10)

since \( E(C_i | C_{i-1}) = C_{i-1} + E(N_i | C_{i-1}) \) and \( C_i = C_{i-1} + N_i \). We should note that because of sequential application of test instances, \( C_{i-1} \) has been already realized and observed at the time when \( t_i \) is applied. The distribution of \( N_i \) or \( C_i \) thus depends on \( C_{i-1} \). Therefore, it seems that the minimization of Expression (9) or (10) is more appropriate than the minimization of Expression (8). The above least squares criteria assume that the variabilities of \( C_i \)'s or \( N_i \)'s are homogenous. But this assumption does not hold for the HGDM. For example, the variance of \( N_i \) is obtained as
\[ \text{Var}(N_i | C_{i-1}) = \frac{m - C_{i-1}}{m-1} p_i (1 - p_i) . \]

Clearly \( \text{Var}(N_i | C_{i-1}) \) is not constant for all \( i \). In the circumstances the weighted least squares method is generally known to be adequate. Park et al.[11] thus suggested that the estimates be computed by minimizing
\[ \sum_{i=1}^{n} \left( x_i - E(N_i | C_{i-1}) \right)^2 \quad \text{Var}(N_i | C_{i-1}) . \]

Next we consider the problem of estimating parameters of the generalized HGDM. It is not difficult to obtain from Expressions (4), (6) and (7) that
\[ E(N_i | C_{i-1}) = (m - C_{i-1}) p_i , \]
\[ E(N_i) = m p_i \prod_{j=1}^{i} (1 - p_j) \]
and
\[ E(C_i) = m \left[ 1 - \prod_{j=1}^{i} (1 - p_j) \right] . \]

These expected values are identical to those for the HGDM. If we estimate parameters by the least squares method, the estimation and prediction results for the
generalized HGDM are the same with the corresponding results for the HGDM. This implies that the generalized HGDM performs at least as well as the HGDM.

The previous studies on the HGDM employed the least squares method mainly due to the mathematical difficulty of the maximum likelihood method. However, sometimes the maximum likelihood estimates can be computed by the least squares method. This approach is illustrated by means of the generalized HGDM. The log likelihood function is obtained from Expression (5) as

\[ L(m, \mathbf{p}) = \sum_i \left[ \ln I(m - C_{\cdot i} + 1) + x_i \ln p_i + (m - C_{\cdot i}) \ln (1 - p_i) \right] \]

\[ = \sum_i l_i \]

where \( \mathbf{p} \) is the vector of \( p_i \)'s and \( I(\cdot) \) denotes the gamma function. Instead of maximizing \( L(m, \mathbf{p}) \), we may minimize \( L(m, \mathbf{p}) = -L(m, \mathbf{p}) \) with respect to \( m \) and \( \mathbf{p} \). Note that \( l_i \)'s are all negative and \( L(m, \mathbf{p}) = \sum_i (z_i - \sqrt{-l_i})^2 \) where \( z_i = 0 \) for all \( i \). Therefore the maximum likelihood estimates are the least squares estimates for the nonlinear regression model \( z_i = \sqrt{-l_i} + \varepsilon_i \), where \( \varepsilon_i \) is the error term. The available nonlinear least squares procedures can be used for computing the maximum likelihood estimates.

6. Prediction for the Generalized HGDM

Suppose that we want to predict the number of faults newly detected by next \( d \) test instances. Such a prediction problem occurs when we determine the software release time or further test instances required to meet the given software reliability objective. This prediction problem can be solved by estimating \( E(N_{s+1} + \cdots + N_{s+d} | C_n) \).

Replacing \( n \) in Expression (6) with \( n + d \), the joint distribution of \( N_i, i = 1, 2, \ldots, n + d \) is obtained as

\[ P(N_i = x_i, i = 1, 2, \ldots, n + d) = \left( \prod_{j=1}^{m} p_j \prod_{j=m}^{m+d} (1 - p_j) \right)^{x_i} \]

\[ = \left( \prod_{j=1}^{m} (1 - p_j) \right)^{x_i} \sum_{x_{n+1}}^{x_{n+d}} \prod_{j=1}^{m+d} (1 - p_j) \]

\[ \quad \cdot \prod_{j=1}^{m} p_j \]

(11)

Division of Expression (11) by Expression (6) results in the conditional distribution of \( N_{s+i}, i = 1, 2, \ldots, d \).

\[ P(N_{s+i} = x_{s+i}, i = 1, \ldots, d | N_i, i = 1, \ldots, n) = \frac{P(N_{s+i} = x_{s+i}, i = 1, \ldots, d | C_n)}{P(N_i = x_i, i = 1, \ldots, n)} \]

\[ = \left( \prod_{j=1}^{m} (1 - p_j) \right)^{x_{s+i}} \sum_{x_{n+1}}^{x_{n+d}} \prod_{j=1}^{m+d} (1 - p_j) \]

This is also a multinomial distribution of which parameters are \( (m - C_n) \) and \( p_{s+i} \prod_{j=1}^{m} (1 - p_{s+j}) \), \( i = 1, 2, \ldots, d \). Then

\[ P(N_{s+1} + \cdots + N_{s+d} = x | C_n) = \left( m - C_n \right) \left( 1 - \prod_{j=1}^{m} (1 - p_{s+j}) \right)^{x} \]

\[ \cdot \prod_{j=1}^{m} (1 - p_{s+j}) \]

Thus

\[ E(N_{s+1} + \cdots + N_{s+d} | C_n) = \left( m - C_n \right) \left( 1 - \prod_{j=1}^{m} (1 - p_{s+j}) \right). \]

(12)

By replacing the parameters in Expression (12) with the corresponding estimates, we can predict the number of faults newly discovered by next \( d \) test instances.

7. Conclusions

SRGMs are useful statistical tools for monitoring and evaluating the quality of a software system. It is necessary to develop new SRGMs and modify existing SRGMs in order to model the test-and-debug
process more realistically. Thus we generalized the HGDM by assuming that the sensitivity factor is a binomial random variable, not a constant. The generalization enables us to easily apply the HGDM and characterize its statistical properties. Methods for parameter estimation and prediction were discussed. Further generalization will be to incorporate the concept of imperfect debugging into the generalized HGDM.

References

박 중 양  
e-mail : parkjy@nongae.gsu.ac.kr  
1982년 연세대학교 응용 통계학과 (학사)  
1984년 한국과학기술원 산업공학과 응용 통계학과 (석사)  
1994년 한국과학기술원 산업공학과 응용 통계학과 (박사)  
1984년~1989년 경상대학교 통신통계학과 교수  
1989년~현재 경상대학교 통계학과 교수  
관심분야 : 소프트웨어 신뢰성, 신경망, 신형 통계 모형, 실험계획법

박 재 홍  
e-mail : pjh@nongae.gsu.ac.kr  
1978년 중부대학교 수학과 (학사)  
1980년 중앙대학교 전산학과 (석사)  
1988년 중앙대학교 전산학과 (박사)  
1983년~현재 경상대학교 컴퓨터과 학과 교수  
관심분야 : 소프트웨어 신뢰성, 시험도구 자동화

유 창 열  
e-mail : cyyoo@nc.namhae.ac.kr  
1987년 경상대학교 전산통계학과 (학사)  
1994년 경상대학교 대학원 전자계산학과 (석사)  
1994년~1997년 경상대학교 대학원 박사과정 수료  
1996년~현재 남해전문대학 사무자동화과 조교수  
관심분야 : 소프트웨어 신뢰성, 객제지향 소프트웨어공학, 멀티미디어통신

1984년~현재 경상대학교 전산통계학과 교수  
1989년~현재 경상대학교 통계학과 교수  
관심분야 : 소프트웨어 신뢰성, 신경망, 신형 통계 모형, 실험계획법

박 재 홍  
e-mail : pjh@nongae.gsu.ac.kr  
1978년 중부대학교 수학과 (학사)  
1980년 중앙대학교 전산학과 (석사)  
1988년 중앙대학교 전산학과 (박사)  
1983년~현재 경상대학교 컴퓨터과 학과 교수  
관심분야 : 소프트웨어 신뢰성, 시험도구 자동화

유 창 열  
e-mail : cyyoo@nc.namhae.ac.kr  
1987년 경상대학교 전산통계학과 (학사)  
1994년 경상대학교 대학원 전자계산학과 (석사)  
1994년~1997년 경상대학교 대학원 박사과정 수료  
1996년~현재 남해전문대학 사무자동화과 조교수  
관심분야 : 소프트웨어 신뢰성, 객제지향 소프트웨어공학, 멀티미디어통신