The Journal of Systems & Software 168 (2020) 110643

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Optimal test activity allocation for covariate software reliability and security models[☆]

Vidhyashree Nagaraju[a], Chathuri Jayasinghe[b], Lance Fiondella[a,*]

[a] Electrical and Computer Engineering, University of Massachusetts Dartmouth, MA, USA
[b] Department of Statistics, University of Sri Jayewardenepura, Nugegoda, Sri Lanka

ARTICLE INFO

Article history:
Received 15 September 2019
Received in revised form 10 February 2020
Accepted 11 May 2020
Available online 15 May 2020

Keywords:
Software reliability growth model
Software security
Non-homogeneous Poisson process
Proportional hazards model
Covariates
Test activity allocation

ABSTRACT

Traditional software reliability growth models enable quantitative assessment of the software testing process by characterizing the fault detection in terms of testing time or effort. However, the majority of these models do not identify specific testing activities underlying fault discovery and thus can only provide limited guidance on how to incrementally allocate effort. Although there are several novel studies focused on covariate software reliability growth models, they are limited to model development, application, and assessment.

This paper presents a non-homogeneous Poisson process software reliability growth model incorporating covariates based on the discrete Cox proportional hazards model. An efficient and stable expectation conditional maximization algorithm is applied to identify the model parameters. An optimal test activity allocation problem is formulated to maximize fault discovery. The proposed method is illustrated through numerical examples on two data sets.

1. Introduction

Traditional software reliability growth models (SRGM) (Farr and Smith, 1984) characterize the fault discovery process during testing as a non-homogeneous Poisson process (NHPP). These models predict future faults as a function of testing time or effort, enabling inferences such as the number of faults remaining, additional time required to achieve a specified reliability, and optimization problems such as optimal release (Zhao and Xie, 1993) and effort allocation (Fiondella and Gokhale, 2008). However, the vast majority of these NHPP SRGM do not identify the underlying software testing activities that lead to fault discovery. Thus, effort allocation based on these models can only provide general guidance on the amount of effort to invest and limited information regarding the effectiveness of specific testing activities.

More recently, bivariate NHPP SRGM (Ishii et al., 2008) and covariate models (Rinsaka et al., 2006), which are capable of characterizing faults discovered as a function of multiple software testing activities such as calendar time, number of test cases executed, and test execution time have been proposed. Covariate models are an especially attractive alternative to testing effort models (Yamada et al., 1986) because they only introduce a single additional parameter per metric and do not require sequential model fitting procedures, although covariate models still require stable and efficient numerical methods. To realize the full potential of covariate models, generalized optimization procedures such as test activity allocation are needed to guide the distribution of limited resources among specific test activities in order to maximize fault discovery, correction, and improved reliability.

Early metrics based models include the work of Khoshgoftaar and Munson (1990), who performed regression analysis to study the relationship between software complexity metrics and the number of errors found during test and validation. Alternative estimation techniques to create linear models of software quality (Khoshgoftaar et al., 1992b) were subsequently evaluated. Khoshgoftaar et al. (1992a) also applied nonlinear regression to model the relationship between software metrics and the number of faults in a program module. Hudepohl et al. (1996) implemented these methods in the Emerald (Enhanced Measurement for Early Risk Assessment of Latent Defects) tool and applied them to telecommunications software.

More recent covariate models include the work of Rinsaka et al. (2006) combined the proportional hazards model and non-homogeneous Poisson process to provide a generalized fault detection process possessing time-dependent covariate structure. Shibata et al. (2006) subsequently extended this to a cumulative Bernoulli trial process. Okamura et al. (2010) proposed a

[☆] V. Nagaraju and L. Fiondella were supported by the US National Science Foundation under Award (#1749635). V. Nagaraju was sponsored by the Army Research laboratory, USA under Cooperative Agreement Number $W911NF-19-2-0224$.

[*] Corresponding author.
E-mail addresses: vnagaraju@umassd.edu (V. Nagaraju), chathuri@sjp.ac.lk (C. Jayasinghe), lfiondella@umassd.edu (L. Fiondella).

https://doi.org/10.1016/j.jss.2020.110643
0164-1212/© 2020 Elsevier Inc. All rights reserved.

Nomenclature

Acronyms

AIC	Akaike information criterion
BIC	Bayesian information criterion
CM	Conditional maximization
DW	Discrete Weibull
DCPH	Discrete Cox Proportional Hazards model
EM	Expectation maximization
ECM	Expectation conditional maximization
NHPP	Non-homogeneous Poisson process
SRGM	Software reliability growth model
SRM	Software reliability model
GM	Geometric
LL	Log-likelihood
MSE	Mean square error
MLE	Maximum likelihood estimation
NB	Negative Binomial
RLL	Reduced log-likelihood

Notation

M	Indigenous faults in the software system
i	Discrete time
n	Number of segments in given time interval
$B(\bullet)$	Binomial distribution
m	Number of trials
p	Probability of success
ω	Number of faults that would be detected with indefinite testing
$\boldsymbol{\beta}$	Vector of Cox model parameters
θ	Vector of model parameters
b	Shape parameter
y_i	Number of faults detected at time i
$\mathbf{x_i}$	$\left(x_{i1}, x_{i2}, \ldots, x_{ip}\right)$; The covariates associated with y_i at testing time i The covariates are *time dependent* since they depend on i
T	The time to failure of a fault, assumed i.i.d.
$F_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}$	$\Pr(T \leq i)$; cdf of T at $T = i$ in the presence of $\mathbf{x_i}$
$S_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}$	$\Pr(T > i)$; survival function of T at $T = i$ in the presence of $\mathbf{x_i}$
$p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}$	$\Pr(T = i)$ in the presence of $\mathbf{x_i}$
$h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}$	$\frac{\Pr(T=i)}{\Pr(T\geq i)} = \frac{p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}}{S_{i-1,\mathbf{x_{i-1}};\theta,\boldsymbol{\beta}}}$; hazard function of T at $T = i$ in the presence of $\mathbf{x_i}$
$F^0_{i;\theta}$	$\Pr(T \leq i)$; cdf of T at $T = i$ at baseline
$S^0_{i;\theta}$	$\Pr(T > i)$; survival function of T at $T = i$ at baseline
$p^0_{i;\theta}$	$\Pr(T = i)$ at baseline
$h^0_{i;\theta}$	$\frac{\Pr(T=i)}{\Pr(T\geq i)} = \frac{p^0_{i;\theta}}{S^0_{i-1;\theta}}$; hazard function of T at $T = i$ at baseline
ν	Number of model parameters
$H_{n;\omega,\theta}$	Mean value function

multi-factor software reliability model based on logistic regression and an efficient algorithm to estimate the model parameters. Okamura and Dohi (2014) combined Poisson regression-based fault prediction and generalized linear models (Okamura and Dohi, 2015) with metrics-based software reliability growth models. Shibata et al. (2015) implemented these methods in the M-SRAT (Metrics-based Software Reliability Assessment Tool).

Covariate models exhibit substantially improved prediction capabilities over NHPP software reliability growth models. Therefore, to enhance the utility of covariate models and encourage their use in practice, this paper makes the following primary contributions

- A software reliability growth model possessing a discrete Cox proportional hazard rate to incorporate covariates. This approach is analogous to the model introduced in Shibata et al. (2006), which has been constructed based on a discrete time model developed by Kalbfleisch and Prentice (1973). However, formulation and estimation of our model takes into account the fact that the counts of software faults in disjoint intervals of the point process are independent random variables possessing a Poisson distribution with non-homogeneous rates, which is an assumption of a NHPP by design.
- Efficient expectation conditional maximization (ECM) (Nagaraju et al., 2017) algorithms to estimate the numerical parameters of a model that best characterize the data.
- A generalization of the testing effort allocation problem (Yamada et al., 1993) to covariate models referred to as the *optimal testing activity allocation problem* to (i) maximize fault discovery within a budget constraint or (ii) minimize the budget required discover a specified number of additional faults.

The illustrations apply the model to two data sets from the literature with the ECM algorithms and then solves the optimal testing activity allocation problem. The results indicate that periodic application of testing activity allocation could more effectively guide the type and amount of specific testing activities throughout the software testing process in order to discover more faults despite limited resources.

The remainder of the paper is organized as follows: Section 2 describes the mathematical formulation of the NHPP SRGM as well as the discrete Cox proportional hazards model. Section 3 discusses proportional hazards modeling incorporating covariates and formulates the optimal test activity allocation problem to maximize fault discovery. Section 4 describes model estimation, assessment, and selection. Section 5 illustrates the proposed approach on two data sets from the literature, while Section 6 provides conclusions and future research.

2. Point process based SRGMs

This section formulates the nonhomogenous Poisson process software reliability growth model and Discrete Cox Proportional Hazards (DCPH) model upon which the subsequent covariate model is developed.

Software faults are detected at random discrete times, which are consistent with the abrupt nature of software failures. Let M be the number of faults within the software and T_j, $j =$

$1, 2, \ldots, M$ the time the jth fault is first discovered. The subscript is not used further because it is assumed that the time to failure of each fault is identically distributed. Moreover, let $\Pr(T = i) = p_i$, $i = 1, 2, \ldots, n$, where i refers to the (discrete) time at which the software fails. Furthermore, it is also possible that more than one software fault is detected in a discrete unit of time.

Like the majority of NHPP SRGM it is assumed that (i) software failures are independent from one another and (ii) faults detected are removed immediately without introducing additional faults and no new faults are introduced.

2.1. Binomial process based SRGMs

Let Y_i be the number of faults detected at time i such that $S_n = \sum_{i=1}^{n} Y_i$ is the total number of faults detected through the first n intervals. Denoting the binomial distribution with m trials and success probability p

$$B(n; m; p) = \binom{m}{n} p^{n} \, (1-p)^{m-n} \tag{1}$$

For fixed M, the distribution of S_n is

$$\begin{aligned}\Pr(S_n = y) &= B^{(n)}(y; M; \mathbf{p}_n)\\ &= \sum_{j=0}^{y} B^{(n-1)}(j; M; \mathbf{p}_{n-1}) \times\\ &\quad B(y-j; M-j; p_n) \qquad (2)\\ &= B(y; M; 1 - \prod_{i=1}^{n} \bar{p}_i) \qquad (3)\end{aligned}$$

with mean

$$E\big[S_n\big] = M\big(1 - \prod_{i=1}^{n} \bar{p}_i\big) \tag{4}$$

where $\bar{p} = 1 - p$ and

$$\mathbf{p}_n = (p_1, p_2, \ldots, p_n)^{\mathbf{T}}. \tag{5}$$

Eq. (2) can be explained by dividing an interval of time into n segments and applying the binomial theorem to two segments of length $n-1$ and 1 respectively. Eq. (3) follows from an induction argument using the convolution of two binomial distributions. It also follows intuitively from the observation that $1 - \prod_{i=1}^{n} \bar{p}_i$ is the probability that at least one failure occurs in any of the n discrete time intervals.

2.2. NHPP based SRGMs

In this section we discuss NHPP based models that have been introduced in the literature to model software faults.

2.2.1. Discrete NHPP SRGMs

Allowing M to follow a Poisson distribution with mean ω such that $\Pr(M = k) = \omega^{k}\frac{e^{-\omega}}{k!}$ provides the NHPP SRGM with distribution

$$\Pr(S_n = y) = \frac{\left(\omega(1 - \prod_{i=1}^{n} \bar{p}_i)\right)^{y}}{y!} \exp\left(-\omega\left(1 - \prod_{i=1}^{n} \bar{p}_i\right)\right) \tag{6}$$

where alternative values for ω produce different NHPP SRGMs.

2.2.2. Standard NHPP model

The standard NHPP model (Lyu, 1996), which is used to model software faults differs from the model discussed in Section 2.2.1. In the standard approach, instead of dealing with the individual probabilities $\Pr(T = i) = p_i$, we consider the cumulative probability $F_i = \Pr(T \leq i)$. Then, for fixed M,

$$\Pr(S_n = y) = \binom{M}{y} F_n^{y} \, \bar{F}_n^{M-y} \tag{7}$$

and, mixing with a Poisson distribution with mean ω gives

$$\Pr(S_n = y) = \frac{(\omega F_n)^{y}}{y!} \exp(-\omega F_n). \tag{8}$$

Comparing Equation (6) with (8) shows that the two NHPP models are not the same since

$$1 - \prod_{i=1}^{n} \bar{p}_i \neq F_n = \sum_{i=1}^{n} p_i. \tag{9}$$

Now, note that discrete time hazard function at a certain time point i is defined as

$$\begin{aligned}h_i &= \Pr(T = i | T > i - 1)\\ &= \Pr(T = i | T \geq i). \qquad (10)\end{aligned}$$

With some basic algebra it can be shown that the probability of $T = i$ can be expressed as a function of hazards as

$$Pr(T = i) = p_i = h_i \prod_{j=1}^{i-1} (1 - h_j). \tag{11}$$

Note that Eq. (11) possesses the natural interpretation that a fault detected in interval i with probability p_i, must have escaped detection in intervals $j = 1, 2, \ldots, i-1$ with hazard rate $\bar{h}(j)$ and is detected at time i with hazard rate h_i. By Eqs. (10) and (11) we then have that the corresponding discrete time survival function at time point i is

$$S_i = P(T > i) = \prod_{j=1}^{i} (1 - h_j) = \exp\left\{\sum_{j=1}^{i} \ln(1 - h_j)\right\}. \tag{12}$$

Note that if we set $p_i = h_i$ in the discrete SRGM, Eq. (6) reduces to the standard SRGM case Eq. (8). This is the only case where equality in Eq. (9) holds, since

$$\begin{aligned}1 - \prod_{i=1}^{n} \bar{p}_i &= 1 - \prod_{i=1}^{n} \bar{h}_i\\ &= 1 - \prod_{i=1}^{n} (1 - h_i)\\ &= 1 - S_n = F_n, \qquad (13)\end{aligned}$$

producing the equality in Eq. (9). Therefore, the standard SRGM given in Eq. (6) implicitly assumes the discrete hazard function as the fault-detection probability.

3. Proportional hazards modeling incorporating covariates

This section describes the formulation of a discrete Cox proportional hazards model incorporating covariates.

Testing occurs in discrete intervals $i = 1, 2, \ldots, n$. An NHPP software reliability model assumes that the point process possesses independent Poisson distributed increments. In other words, the numbers of faults detected over these n disjoint time intervals are independent. Suppose that we are interested in investigating the effect of p software test activities on fault detection. We denote the vector of software test activities in the

ith testing interval by $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ for $i = 1, 2, \ldots, n$. We will also redefine our notations noting the dependence of the functions on the parameters of the distribution of T (denoted θ) and coefficients of the p software test activities in the model (denoted $\boldsymbol{\beta} = \left(\beta_1, \beta_2, \ldots, \beta_p\right)^{\mathbf{T}}$), which must either be estimated using data or assigned values. Here test activities are any tool, technique, or procedure to detect software defects. Some examples include code reviews and interactive debugging.

The Cox proportional hazards model for a discrete process (Shibata et al., 2006) is

$$h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} = 1 - (1 - h^0_{i;\theta})^{g(\mathbf{x_i};\boldsymbol{\beta})}, \tag{14}$$

for each $i = 1, 2, \ldots, n$, where $h^0_{i;\theta}$ is known as the baseline hazard function and it is usual to assume that

$$g(\mathbf{x_i}; \boldsymbol{\beta}) = \exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}). \tag{15}$$

Proof. From the definition of the discrete time hazard function given in Eq. (10)

$$\begin{aligned} 1 - h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} &= \frac{S_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}}{S_{i-1,\mathbf{x_{i-1}};\theta,\boldsymbol{\beta}}} \\ &= \left(\frac{S^0_{i;\theta}}{S^0_{i-1;\theta}}\right)^{g(\mathbf{x}_i;\boldsymbol{\beta})} \end{aligned} \tag{16}$$

However, $1 - h^0_{k;\theta} = \frac{S^0_{k;\theta}}{S^0_{k-1;\theta}}$. Therefore, Eq. (16) is equal to

$$1 - h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} = (1 - h^0_{i;\theta})^{g(\mathbf{x_i};\boldsymbol{\beta})} \tag{17}$$

and Eq. (14) follows ■.

From Eq. (17),

$$\begin{aligned} \prod_{k=1}^{n-1}(1 - h^0_{k;\theta})^{g(\mathbf{x_k};\boldsymbol{\beta})} &= \prod_{k=1}^{n-1}\left(1 - h_{k,\mathbf{x_k};\theta,\boldsymbol{\beta}}\right) \\ &= \prod_{k=1}^{n-1}\frac{S_{k,\mathbf{x_k};\theta,\boldsymbol{\beta}}}{S_{k-1,\mathbf{x_{k-1}};\theta,\boldsymbol{\beta}}} \\ &= S_{n-1,\mathbf{x_{n-1}};\theta,\boldsymbol{\beta}} \end{aligned} \tag{18}$$

so that

$$\begin{aligned} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} &= h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} S_{i-1,\mathbf{x_{i-1}};\theta,\boldsymbol{\beta}} \\ &= \left(1 - (1 - h^0_{i;\theta})^{g(\mathbf{x_i};\boldsymbol{\beta})}\right)\prod_{k=1}^{i-1}(1 - h^0_{k;\theta})^{g(\mathbf{x_k};\boldsymbol{\beta})} \end{aligned} \tag{19}$$

follows from Eqs. (11), (14) and (18).

The Cox model incorporates covariates $\mathbf{x}$ into the hazard rate and therefore assesses the effects they may have on fault detection. If the faults are assumed to be Poisson distributed, then a likelihood analysis can be performed using data available for estimation. Hence, it is possible to infer the effect of a covariate on faults despite the presence of multiple covariates.

Mean value function

The mean number of faults detected through the nth interval is

$$H_{n;\omega,\theta,\boldsymbol{\beta}} = \omega \sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} \tag{20}$$

3.1. Baseline hazard functions

The baseline hazard function in discrete time interval i possessing parameters θ is

$$h^0_{i;\theta} = \frac{p^0_{i,\theta}}{S^0_{i-1,\theta}} \tag{21}$$

where $p^0_{\cdot,\theta}$ and $S^0_{\cdot,\theta} = 1 - F^0_{\cdot,\theta}$ respectively denote the baseline probability mass function and survival function of the discrete time distribution T at the baseline levels of software test activities in the model. Specific hazard functions including the geometric, negative binomial, and discrete Weibull are given below and can be substituted into Eq. (19) to obtain various Cox proportional hazards models.

3.1.1. Geometric (GM)

$$h^0_{i;b} = b \tag{22}$$

where b is the probability of detecting a fault and hence, $b \in (0, 1)$.

3.1.2. Negative binomial of order two (NB)

$$h^0_{i;b} = \frac{ib^2}{1 + b(i-1)} \tag{23}$$

where $b \in (0, 1)$ and 2 indicates the order.

3.1.3. Discrete Weibull of order two (DW)

$$h^0_{i;b} = 1 - b^{i^2-(i-1)^2} \tag{24}$$

where $b \in (0, 1)$ and 2 indicates the order.

3.2. Optimal test activity allocation to maximize fault discovery

Similar to the concept of effort allocation (Yamada et al., 1986) in NHPP software reliability growth models, it is possible to employ covariate models to guide resource allocation. Generalization of the testing effort concept to covariate models, requires division of resources across multiple activities, each of which possesses and effectiveness characterized by the parameter β_i but may also impose unique cost and time requirements. Examples of activities associated with traditional software reliability testing include alternative black-box testing methods (Ammann and Offutt, 2016), with costs characterized by the hourly rates charged by skilled employees or consultants, whereas examples in the context of software security testing include traditional software reliability testing methods relevant to security as well as static and dynamic testing tools and techniques for exposing vulnerabilities.

Given n intervals of observed data and a budget of B resources to allocate to r activities (covariates), maximizing the total number of faults or vulnerabilities detected, so that they can be corrected prior to release is formulated as

$$\arg\max \widehat{H}_{(n+1);\omega,\theta,\boldsymbol{\beta}} \tag{25}$$

subject to

$$\sum_{j=1}^{r} \mathbf{c}_j \mathbf{x}_{j,(n+1)} \le B$$

$$\mathbf{c}_j \left(\mathbf{x}_{j,(n+1)} - \mathbf{x}_{j,(n)}\right) = B_j \ge 0$$

where $\mathbf{c}_j > 0$ is the cost associated with an additional unit of activity j. In practice, the model can be fit to data from the first

i intervals and Eq. (25) solved based on the maximum likelihood estimates of the model and the budget to be allocated on activities during interval $(i+1)$.

Alternatively, an individual may wish to expose a specified number of additional faults or vulnerabilities with the smallest budget possible, which is formulated as

$$\arg\min \sum \mathbf{c}_j \mathbf{x}_j \tag{26}$$

subject to

$$\widehat{H}_{(n+1);\omega,\theta,\boldsymbol{\beta}} - \widehat{H}_{n;\omega,\theta,\boldsymbol{\beta}} \geq k$$

$$\beta_{j+1} - \beta_j \geq 0$$

where the first constraint ensures that the predicted number of additional faults or vulnerabilities is greater than k and the second constraint requires that each test activity allocation is nonnegative.

4. Model estimation, assessment, and selection

This section describes model estimation, assessment, and selection methods. Estimation methods include maximum likelihood estimation (MLE) and the expectation conditional maximization (ECM) algorithm as well. An example derivation of the ECM update rules is provided in the context of the geometric hazard function. Initial parameter estimation, measures of goodness of fit, and model selection are also discussed.

4.1. Maximum likelihood estimation

For the purpose of estimation and inferential procedures, the likelihood function of the process needs to be constructed. The likelihood function is merely the joint distribution of the sample of observed values. In this case, the observed values are data: $(y_i, \mathbf{x_i}, i = 1, 2, \ldots, n)$ where n is the number of testing intervals. As mentioned in Section 3, an NHPP software reliability model assumes that the point process possesses independent Poisson distributed increments. In other words, the number of faults detected over these disjoint time intervals are independent. Hence, the likelihood function of the NHPP SRGM given in Eq. (19) is

$$\begin{aligned}\mathbf{L}(\theta, \beta, \omega) &= \Pr(Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n)\\ &= \prod_{i=1}^{n} \exp\left(-\omega p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}\right) \frac{\left(\omega p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}\right)^{y_i}}{y_i!}\\ &= \exp\left(-\omega \sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}\right) \omega^{\sum_{i=1}^{n} y_i}\\ &\quad \prod_{i=1}^{n} \frac{p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}^{y_i}}{y_i!},\end{aligned} \tag{27}$$

and the corresponding log-likelihood function

$$\begin{aligned}LL(\theta, \boldsymbol{\beta}, \omega) = &-\omega \sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} + \sum_{i=1}^{n} y_i \ln(\omega)\\ &+ \sum_{i=1}^{n} y_i \ln(p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}) - \sum_{i=1}^{n} \ln(y_i!)\end{aligned} \tag{28}$$

incorporates the covariates $\mathbf{x}_i$ through $p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}$. The maximum likelihood estimates of the model parameters can be obtained by solving $\frac{\partial l}{\partial \omega} = 0$, $\frac{\partial l}{\partial \theta} = 0$, and $\frac{\partial l}{\partial \beta_j} = 0, j = 1, 2, \ldots, p$.

4.2. Expectation conditional maximization algorithm

This section describes the expectation conditional maximization (Nagaraju et al., 2017; Zeephongsekul et al., 2016) algorithm to identify the maximum likelihood estimates of a model.

The steps to obtain the CM steps of the ECM algorithm of a covariate based NHPP SRGM are as follows:

- (S.1) Step one specifies the log-likelihood function, as described in Eq. (28).
- (S.2) Step two reduces the log-likelihood function from ν to $(\nu - 1)$ parameters by differentiating the log-likelihood function with respect to ω, equating the result to zero, and solving for ω to produce

$$\widehat{\omega} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}}, \tag{29}$$

  which is then substituted into the log-likelihood function to obtain the reduced log-likelihood (RLL) function.
- (S.3) Step three derives the conditional maximization steps for the remaining $(\nu - 1)$ parameters by computing partial derivatives

$$\frac{\partial RLL}{\partial \theta} = 0 \tag{30}$$

  and

$$\frac{\partial RLL}{\partial \boldsymbol{\beta}} = 0 \tag{31}$$

  where θ denotes all parameters except ω and $\boldsymbol{\beta}$ denote coefficients related to all p covariates.
- (S.4) Step four cycles through the $(\nu - 1)$ CM-steps holding the other $(\nu - 2)$ parameters constant, applying a numerical root finding algorithm to determine the maximum likelihood estimates $\widehat{\theta}/\omega$. This process continues until a user specified convergence criterion is achieved.
- (S.5) Step five computes the MLE of ω by substituting the estimates of θ and $\boldsymbol{\beta}$ into Eq. (29), producing the MLE for all ν parameters of the model.

Steps (S.1) through (S.5) can be applied to different hazard functions introduced in Section 3.1. Step (S.4) reduces a $(\nu - 1)$-dimensional problem to $(\nu - 1)$ single-dimensional problems, enabling the application of a stable numerical method in each CM-step. Coupled with the monotonicity of the ECM algorithm, this ensures convergence to the maximum likelihood, which is especially important in an open source implementation (Nagaraju et al., 2019).

4.3. Geometric model

This section illustrates the derivation of CM steps in the context of the geometric distribution possessing a single covariate. Toward this end, step (S.1) substitutes the hazard function for a geometric model (Eq. (22)) and Eq. (19) into Eq. (28), producing

$$\begin{aligned}LL(\omega, b, \beta) = &-\omega \sum_{i=1}^{n} \left(\left(1 - (1-b)^{e^{\mathbf{x_i}\beta}}\right) \prod_{k=1}^{i-1} (1-b)^{e^{x_k \beta}} \right)\\ &+ \log(\omega) \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \log(y_i!) + \sum_{i=1}^{n} \Bigg(y_i\\ &\log\left(\left(1 - (1-b)^{e^{\mathbf{x_i}\beta}}\right) \prod_{k=1}^{i-1} (1-b)^{e^{x_k \beta}} \right) \Bigg).\end{aligned} \tag{32}$$

Step (S.2) differentiates Equation (32) with respect to ω and solves for ω to provide

$$\widehat{\omega} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n}\left(\left(1-(1-b)^{e^{\mathbf{x_i}\beta}}\right)\prod_{k=1}^{i-1}(1-b)^{e^{x_k\beta}}\right)} \tag{33}$$

and the reduced log-likelihood function is obtained by substituting Equation (33) into Eq. (32)

$$\begin{aligned} RLL(b,\beta) = & -\sum_{i=1}^{n} y_i - \sum_{i=1}^{n}\log(y_i!) + \sum_{i=1}^{n}(y_i)\log\left(\sum_{i=1}^{n} y_i \right. \\ & \left. \frac{1}{\sum_{i=1}^{n}\left(\left(1-(1-b)^{e^{\mathbf{x_i}\beta}}\right)\prod_{k=1}^{i-1}(1-b)^{e^{x_k\beta}}\right)}\right) \\ & + \sum_{i=1}^{n}\left(y_i\log\left(\left(1-(1-b)^{e^{\mathbf{x_i}\beta}}\right)\right.\right. \\ & \left.\left.\prod_{k=1}^{i-1}(1-b)^{e^{x_k\beta}}\right)\right) \end{aligned} \tag{34}$$

Step (S.3) differentiates Equation (34) with respect to b and β to produce

$$\begin{aligned} b'' := & \frac{-(1-b'')^{-1+\sum_{i=1}^{n} e^{\mathbf{x_i}\beta'}}\sum_{i=1}^{n} e^{\mathbf{x_i}\beta'}\sum_{i=1}^{n} y_i}{1-(1-b'')^{\sum_{i=1}^{n} e^{\mathbf{x_i}\beta'}}} \\ & + \frac{1}{1-b''}\left(\sum_{i=1}^{n}\left(\frac{y_i e^{\mathbf{x_i}\beta'}(1-b'')^{e^{\mathbf{x_i}\beta'}}}{1-(1-b)^{e^{\mathbf{x_i}\beta'}}}\right)\right. \\ & \left. -\sum_{i=2}^{n}\left(y_i\sum_{k=1}^{i-1} e^{x_k\beta'}\right)\right) \end{aligned} \tag{35}$$

and

$$\begin{aligned} \beta'' := & \frac{(1-b')^{\sum_{i=1}^{n} e^{\mathbf{x_i}\beta''}}\sum_{i=1}^{n} y_i \sum_{i=1}^{n}\left(\mathbf{x_i} e^{\mathbf{x_i}\beta''}\right)}{1-(1-b')^{\sum_{i=1}^{n} e^{\mathbf{x_i}\beta''}}} \\ & \log(1-b') + \left(\sum_{i=1}^{n}\left(\left(1-\frac{1}{1-(1-b')^{e^{\mathbf{x_i}\beta''}}}\right)\right.\right. \\ & \left.\left. y_i\mathbf{x_i}e^{\mathbf{x_i}\beta''} + \mathbf{x_i}e^{\mathbf{x_i}\beta''}\sum_{k=i-1}^{n} y_k\right)\right)\log(1-b') \end{aligned} \tag{36}$$

which are the CM steps of the ECM algorithm.

Step (S.4) applies Eqs. (35) and (36) to respectively update parameters b'' and β'' in alternating iterations.

4.4. Initial parameter estimation

The initial parameter estimates (Okamura et al., 2003) of parameter ω is

$$\omega^{(0)} = n \tag{37}$$

The remaining parameters of the distribution function $F(\bullet;\theta)$ in Eq. (8) are computed as

$$\theta^{(0)} := \sum_{i=1}^{n}\frac{\partial}{\partial\theta}\log\left[f(;\theta,\boldsymbol{\beta})\right] = 0. \tag{38}$$

For example, the initial estimates of the geometric model possessing a single covariate determined from Eq. (38) are

$$b^{(0)} = 1 - e^{\left(\frac{-n}{\sum_{i=1}^{n} e^{\mathbf{x_i}\beta}+\sum_{i=1}^{n}\sum_{k=1}^{i-1} e^{x_k\beta}}\right)} \tag{39}$$

and

$$\begin{aligned} \beta^{(0)} = & -n\left(\sum_{i=1}^{n}\mathbf{x_i} + \log(1-b)\left(\sum_{i=1}^{n}(\mathbf{x_i})e^{\mathbf{x_i}\beta^{(0)}}\right.\right. \\ & \left.\left. + \sum_{i=1}^{n}\sum_{k=1}^{i-1}\left(x_k e^{x_k\beta^{(0)}}\right)\right)\right)^{-1} \end{aligned} \tag{40}$$

Here, $\beta^{(0)}$ lacks a closed form expression and a numerical value of $b^{(0)}$ cannot be determined unless $\beta^{(0)}$ possesses a numerical value. Therefore, Eq. (39) can be substituted into (40) and solved for $\beta^{(0)}$ numerically. The numerical value of $\beta^{(0)}$ is subsequently substituted into Eq. (39) to obtain the initial estimate of $b^{(0)}$. Alternatively, $\beta^{(0)}$ can be held constant at 0, which simplifies to the case of no covariates.

4.5. Goodness of fit measures

This section summarizes some goodness of fit measures to assess how well a model characterizes a failure data set. A simple method based on multiple goodness of fit measures is also covered to discourage poor statistical practices such as relying solely on measures of in sample goodness of fit as opposed to a combination of predictive accuracy and information theory.

4.5.1. Akaike information criterion

The Akaike Information Criterion (Fiondella and Gokhale, 2011) is an information theoretic measure of a model's goodness of fit. The AIC quantifies the tradeoff between model precision and complexity. The AIC of model i is a function of the maximized log-likelihood and the number of model parameters (ν).

$$AIC_i = 2\nu - 2LL(\mathbf{x_i};\widehat{\omega},\widehat{\theta},\widehat{\beta}) \tag{41}$$

The term 2ν of Eq. (41) is a linearly increasing penalty function for the number of parameters, while $LL(\mathbf{x_i};\widehat{\omega},\widehat{\theta},\widehat{\beta})$ is the log-likelihood function of failure data with covariates $\mathbf{x_i}$ evaluated at the maximum likelihood estimate. Model j preserves information better than model i and is preferred with statistical significance if $AIC_{i,j} = AIC_i - AIC_j > 2.0$ (Sakamoto et al., 1986).

4.5.2. Bayesian information criterion

The Bayesian information criterion of model i is a function of the maximized log-likelihood, number of model parameters ν, and the sample size n

$$BIC_i = -2LL(\mathbf{x_i};\widehat{\omega},\widehat{\theta},\widehat{\beta}) + \nu\log(n) \tag{42}$$

The penalty term of the BIC is therefore proportional to the number of parameters ν multiplied by the logarithm of the sample size n.

4.5.3. Sum of Squares Error (SSE)

The sum of squares error, also known as the residual sum of squares, for failure count data is

$$SSE = \sum_{i=1}^{n}(\widehat{H}_{i;\omega,\theta,\beta} - Y_i)^2 \tag{43}$$

where $Y_i = \sum_{j=1}^{i} y_j$ is the cumulative number of faults observed in the first i time intervals.

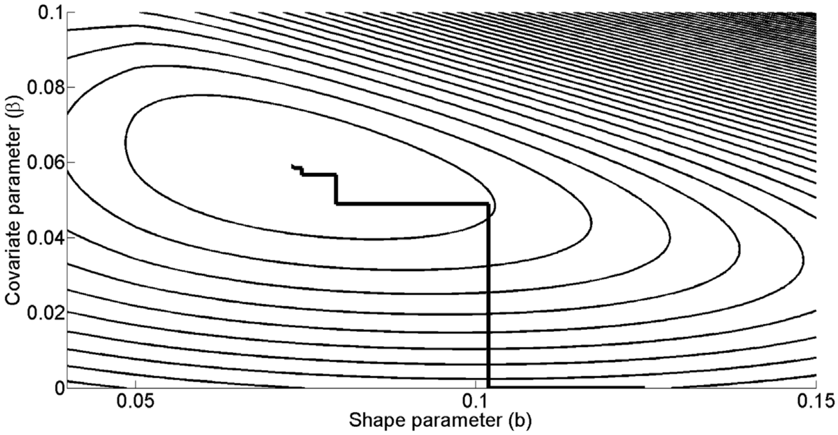

Fig. 1. ECM Iterations of covariate model with geometric hazard function applied to F covariate of DS2 data set.

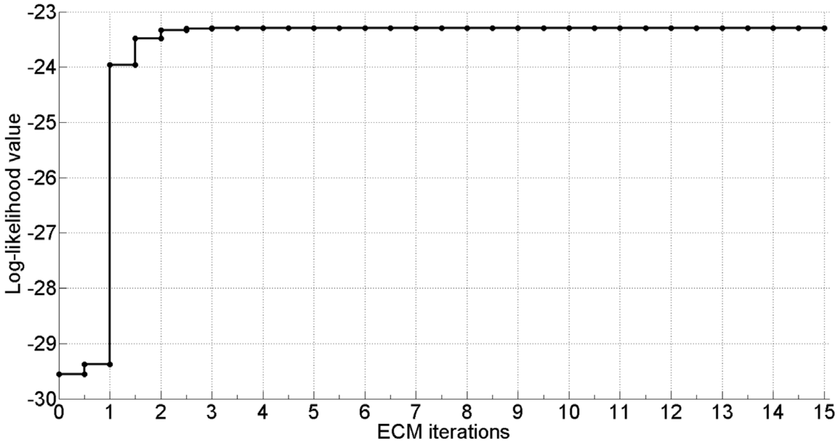

Fig. 2. Log-likelihood of covariate model with geometric hazard function applied to F covariate of DS2 data set.

4.5.4. Predictive Sum of Squares Error (PSSE)

PSSE compares the predictions of a model with data not used to perform model fitting. The predictive sum of squares error for failure count data is

$$PSSE = \sum_{i=n-\ell+1}^{n} (\widehat{H}_{i;\omega,\theta,\beta} - Y_i)^2 \tag{44}$$

where the maximum likelihood estimates of the model parameters are determined from the first $n-\ell$ intervals.

4.5.5. Model selection based on multiple goodness of fit measures

This section presents a simple method based on the critic method (Diakoulaki et al., 1995) to select a model given multiple goodness of fit measures.

Given n models and m measures, Let $f_{i,j}$ be the jth measure for the ith model. Each measure is assigned a normalized score according to

$$x_{i,j} = \frac{f_{i,j} - f_j^+}{f_j^- - f_j^+} \tag{45}$$

where f_j^+ and f_j^- respectively denote the best and worst values of a measure j across all models. Thus, $x_{i,j}$ indicates how close the jth measure of model i is to the ideal. One method to select a model is to compute the median of each model's normalized scores and recommend the model with the highest value. Alternatively, model selection can be based on the average of each model's normalized scores. The critic method, as described here, is a special case of the Analytic Hierarchy Process (AHP) (Li et al., 2006), where weights can be assigned to individual measures based on subjective properties such as relevance and past experience.

5. Illustrations

This section demonstrates the model, ECM algorithm, and optimal covariate allocation. The DS1 and DS2 data sets (Shibata et al., 2006) respectively consisting of $n = 17$ and $n = 14$ weeks of observations are employed. Both data sets possess three covariates, including execution time (E) in hours, failure identification work (F) in person hours, and computer time failure identification (C) in hours. Here, F may indicate activities such as code reviews, where one or more programmers read but did not execute code, whereas C may indicate activities where programmers tested the code and debugged interactively.

The first example graphically illustrates the application of the ECM algorithm to a model with a single covariate. The second example compares goodness of fit measures attained by the geometric, Weibull, and negative binomial models on all possible subsets of covariates contained in the DS1 and DS2 data sets. The final example assesses optimal effort allocation.

5.1. Application of ECM algorithm to covariate NHPP SRGM

Fig. 1 shows the contour plot of the log-likelihood function of data set DS2 for the model with geometric hazard function in Eq. (22) and F covariate. The CM steps made by applying Eqs. (35) and (36) of the ECM algorithm are superimposed. The initial estimate $b^{(0)} = 0.124827$ was determined with Eq. (39)

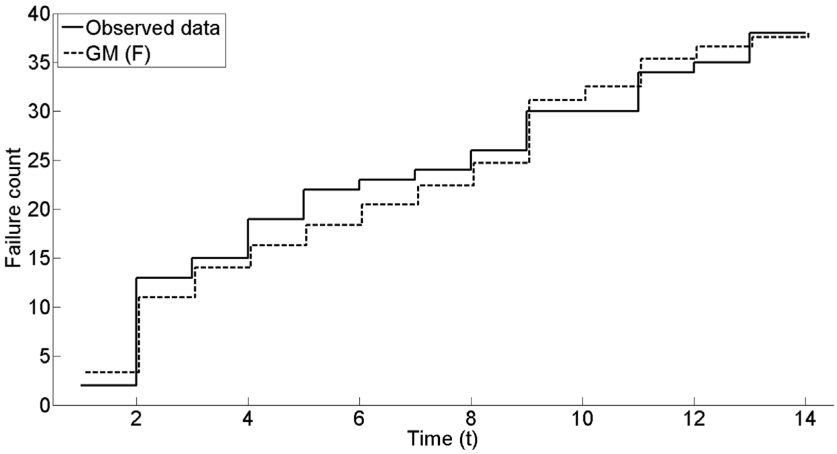

Fig. 3. Fit of covariate model with geometric hazard function applied to F covariate of DS2 data set.

Table 1
Goodness of fit measures for geometric hazard function model on DS1 and DS2.

β	ν	DS1					DS2				
		LLF	AIC	BIC	SSE	PSSE	LLF	AIC	BIC	SSE	PSSE
–	2	−41.47	86.94	88.60	433.73	**0.25**	−29.38	62.76	64.03	45.04	**3.16**
E	3	−36.13	78.25	80.75	187.00	5.79	−24.93	55.86	57.78	80.29	12.56
F	3	−31.21	68.43	**70.93**	120.12	27.41	−23.29	**52.58**	**54.51**	49.61	10.22
C	3	−35.54	77.09	79.59	219.15	4.34	−24.33	54.65	56.57	**44.25**	14.06
EF	4	−30.03	68.05	71.38	73.41	17.35	−23.27	54.54	57.11	54.24	9.98
FC	4	−30.89	69.79	73.12	128.25	85.20	−23.01	54.03	56.58	48.63	12.40
EC	4	−30.08	68.16	71.49	48.25	2.35	−24.09	56.19	58.74	61.53	14.37
EFC	5	**−28.40**	**66.81**	70.97	**36.95**	32.99	**−23.00**	56.01	59.21	45.77	11.58

by holding $\beta^{(0)} = 0.0$. Fig. 1 shows the ECM algorithm reaches the maximum likelihood estimates $\widehat{b} = 0.0729392$ and $\widehat{\beta} = 0.0589373$. Substituting these values into Eq. (33) produces the MLE of $\widehat{\omega} = 42.6185$.

Fig. 2 shows the monotonic increase in log-likelihood attained by the ECM algorithm with CM-steps. Odd steps at 0.5, 1.5, and so on correspond to updates to $\widehat{b}$, while even steps correspond to $\widehat{\beta}$.

Fig. 3 shows the DS2 data accompanied by the fit of the model with geometric hazard function to F covariate. The sum of squares error computed from Eq. (43) is 49.6143.

5.2. Goodness of fit model assessment

This section systematically compares the models with geometric, negative binomial, and discrete Weibull hazard functions considering all eight possible combinations of three covariates in data sets DS1 and DS2.

Table 1 summarizes the goodness of fit of the geometric hazard rate function for each possible combination of covariates, requiring ν parameters. Preferred combinations of covariates with respect to the LLF, AIC, BIC, SSE, and PSSE are indicated in bold. While the model fit using all three covariates achieves the highest log-likelihood value on both data sets, AIC only prefers EFC on DS1, whereas the F covariate is preferred by AIC and BIC in all other cases. For DS1, however, the BIC of EFC and F are close, suggesting that EFC may be preferred for DS1 if both AIC and BIC are taken into consideration. The SSE and PSSE provide conflicting results. Specifically, SSE prefers all three covariates on DS1 and C covariate on DS 2, whereas PSSE is lowest for a model with no covariates on both data sets. However, some combinations of one or more covariate also attain a relatively low SSE and PSSE, suggesting that practical model selection (Sharma et al., 2010) requires a tradeoff between information theoretic and predictive measures of goodness of fit.

Tables 2 and 3 provide similar analysis of the negative binomial and discrete Weibull hazard rates. For each combination of hazard function and data set, both the AIC and BIC prefer the same covariates, but do not consistently agree with the other measures. Considering all three models, both AIC and BIC recommend the NB model with F covariate for DS1 and the GM model with F covariate for DS2, whereas results based on SSE prefer NB with three covariates for both data sets, whereas PSSE recommends GM with no covariates when all three models are considered.

Since no model performs best on all measures in Tables 1–3, we apply the model selection method based on multiple goodness of fit measures. Table 4 reports the normalized values of the measures for the geometric hazard function model on DS1, computed with Eq. (45) and the values given in Table 1.

Table 4 indicates that the measures produced by covariates EFC attain the highest median while covariates EC perform best with respect to the mean of these measures.

Table 5 summarizes the results of applying the model selection method based on multiple goodness of fit measures considering all 24 combinations of three hazard functions and eight covariates.

For both data sets, the same hazard function achieves the top three ranks. Specifically, the negative binomial hazard function with F covariate is recommended for DS1, followed by the EF and EFC combinations of covariates. Similarly, the geometric hazard function with F covariate is recommended for DS2.

5.3. Optimal test activity allocation

This section illustrates optimal test activity allocation formulated in Section 3.2 in the context of the negative binomial model on DS1. For the sake of illustration, the negative binomial model was fit to each combination of covariates with from the first $n - 1$ intervals. The amount of effort originally allocated in the nth interval of DS1 was $E = 7.6$, $F = 24$, and $C = 8$. Thus,

Table 2
Goodness of fit measures for negative binomial hazard function model on DS1 and DS2.

β	ν	DS1					DS2				
		LLF	AIC	BIC	SSE	PSSE	LLF	AIC	BIC	SSE	PSSE
–	2	−36.41	76.82	78.49	208.35	3.99	−30.56	65.12	66.40	104.46	**7.89**
E	3	−32.31	70.62	73.12	70.16	3.16	−28.06	62.12	64.04	134.35	14.09
F	3	−28.80	**63.60**	**66.10**	25.94	1.16	−25.73	**57.46**	**59.38**	106.88	14.05
C	3	−31.96	69.91	72.41	81.73	11.12	−26.81	59.61	61.53	102.88	16.13
EF	4	−28.05	64.11	67.44	18.33	**0.96**	−25.52	59.03	61.59	86.55	13.52
FC	4	−28.37	64.75	68.08	20.77	2.19	−25.59	59.19	61.75	107.53	15.45
EC	4	−28.97	65.95	69.28	33.02	10.18	−26.78	61.57	64.12	95.15	16.06
EFC	5	**−27.29**	64.58	68.74	**11.89**	2.44	**−25.00**	60.00	63.20	**65.26**	14.74

Table 3
Goodness of fit measures for discrete Weibull hazard function model on DS1 and DS2.

β	ν	DS1					DS2				
		LLF	AIC	BIC	SSE	PSSE	LLF	AIC	BIC	SSE	PSSE
–	2	−35.37	74.74	76.41	136.55	18.82	−37.29	78.59	79.87	302.94	12.93
E	3	−33.20	72.40	74.90	122.82	18.46	−36.43	78.86	80.78	357.39	15.86
F	3	−29.47	**64.94**	**67.44**	41.52	**10.41**	−33.40	72.80	74.71	361.65	16.71
C	3	−33.04	72.08	74.58	111.59	21.22	−34.61	75.23	77.14	350.60	17.51
EF	4	−29.20	66.41	69.74	53.55	10.80	−31.34	70.68	73.24	216.38	16.01
FC	4	−29.28	66.56	69.89	**39.76**	12.16	−33.34	74.68	77.23	365.66	17.31
EC	4	−31.42	70.83	74.16	120.59	20.83	−33.64	75.28	77.83	248.60	**7.72**
EFC	5	**−28.91**	67.83	71.99	54.70	12.86	**−29.87**	**69.74**	**72.93**	**138.64**	16.82

Table 4
Normalized measures for geometric hazard function model on DS1.

	–	E	F	C	EF	FC	EC	EFC
LLF	0.00	0.41	0.79	0.45	0.88	0.81	0.87	1.00
AIC	0.00	0.43	0.92	0.49	0.94	0.85	0.93	1.00
BIC	0.00	0.44	1.00	0.51	0.97	0.88	0.97	0.99
SSE	0.00	0.62	0.79	0.54	0.91	0.77	0.97	1.00
PSSE	1.00	0.94	0.68	0.95	0.79	0.00	0.98	0.62
Median	0	0.44	0.79	0.51	0.91	0.81	0.97	**1.00**
Mean	0.2	0.57	0.84	0.59	0.91	0.66	**0.94**	0.92

Table 5
Top combinations of hazard functions and covariates.

Method	DS1			DS2		
	1	2	3	1	2	3
Median	NB (F)	NB (EF)	NB (EFC)	GM (F)	GM (FC)	GM (EF)
Mean	NB (F)	NB (EF)	NB (EFC)	GM (F)	GM (EF)	GM (FC)

Table 6
Optimal covariate effort allocation to DS1 with negative binomial hazard function.

β	$\widehat{H}^*_{n;\omega,\theta,\beta}$	%E	%F	%C
E	6.63	100	–	–
F	2.61	–	100	–
C	0.41	–	–	100
EF	5.30	100	–	–
FC	1.97	–	5.70	94.30
EC	0.62	40.75	–	59.25
EFC	1.55	38.03	–	61.97

a budget of $B = 39.6$ resources were allocated to each subset of covariates. Moreover, covariates were assumed to all possess unit cost ($c_i = 1.0$), but the formulation in Section 3.2 is capable of considering non uniform costs.

Table 6 lists each combination of one or more covariate, the estimated number of faults that would occur with optimal allocation ($\widehat{H}^*_{n;\omega,\theta,\beta}$) using Eq. (25), and the percent of the budget allocated to the available covariates.

Table 6 indicates that optimal allocation of effort to the model with all three covariates dedicates 38.03% of the budget to activity E and the remaining 61.97% to activity C, which respectively correspond to 15.06 and 24.54 units of the original budget. The estimated number of faults is 1.55, whereas allocating effort according to the nth interval of DS1 ($E = 7.6$, $F = 24$, and $C = 8$) estimates only 0.92 faults. These results suggest that optimal allocation could expose nearly 1.7 times as many faults, indicating allocation could have conserved significant resources. For the sake of comparison, we increased the allocations reported in the nth interval by a common factor greater than one until the number of faults estimated was equal to the optimal allocation. That is, each of the three covariates was multiplied by the same factor until 1.55 faults were achieved. It was determine that a total budget of 79.2 (about two times as much effort) would be required to identify the same number of faults as the optimal allocation strategy.

It should be noted that in all cases, these are model predictions. Therefore, the possibility of exposing the number of faults identified by effort allocation will depend on a model's predictive accuracy. Since the model selection method based on multiple goodness of fit measures recommended the negative binomial hazard function with F covariate for DS1, it may be reasonable to allocate the entire budget to F. However, it is not possible to go back in time to recommend effort allocation during the nth interval to determine if the optimal allocation would have actually exposed more faults, indicating that application of covariate models to ongoing projects will be essential to improving their usability in practice.

Fig. 4 studies the sensitivity of test activity allocation to a budget in the interval $B \in (10, 100)$.

Fig. 4 indicates that the effort allocated to the covariates converges asymptotically to slightly less than 60%, 40%, and 10% for covariates C, E, and F respectively. This occurs because the mean value function estimates a finite number of faults and therefore a finite number of faults detectable by the different covariates.

To explain the asymptotic trend in Figs. 4, 5 shows the marginal utility of allocating effort to just one of the three covariates.

The vertical lines indicate the budget allocated to the model with three covariates in Table 6, namely 15.06 units to covariate E and 24.54 to covariate C. Covariate F is the least responsive to effort, explaining why the optimal test activity allocation is divided between the other two covariates.

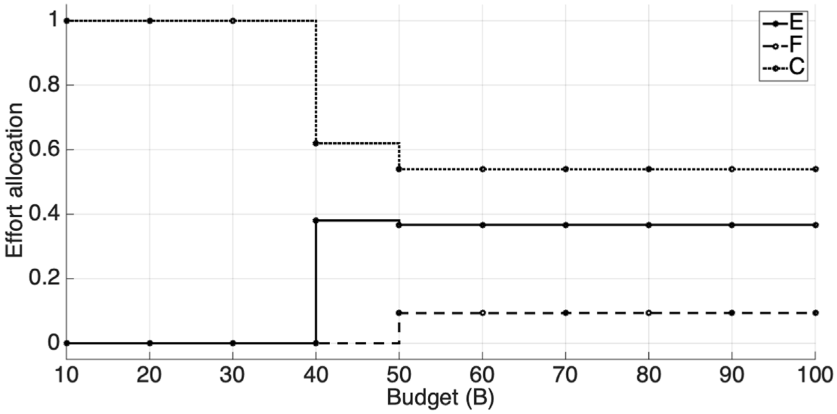

Fig. 4. Sensitivity of optimal test activity allocation to budget.

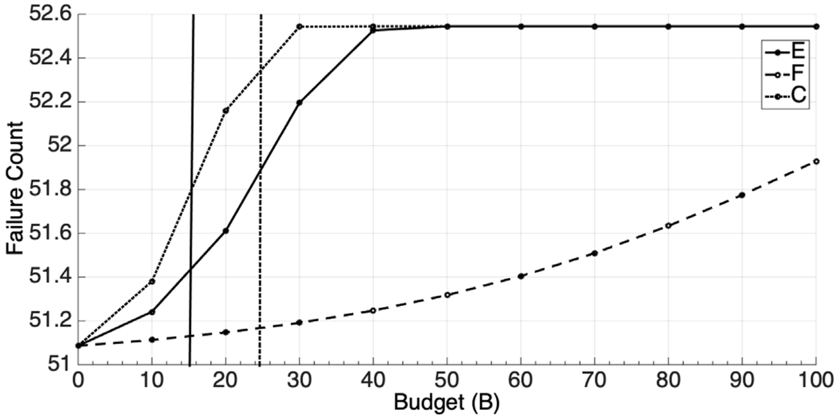

Fig. 5. Marginal utility of NB on DS1.

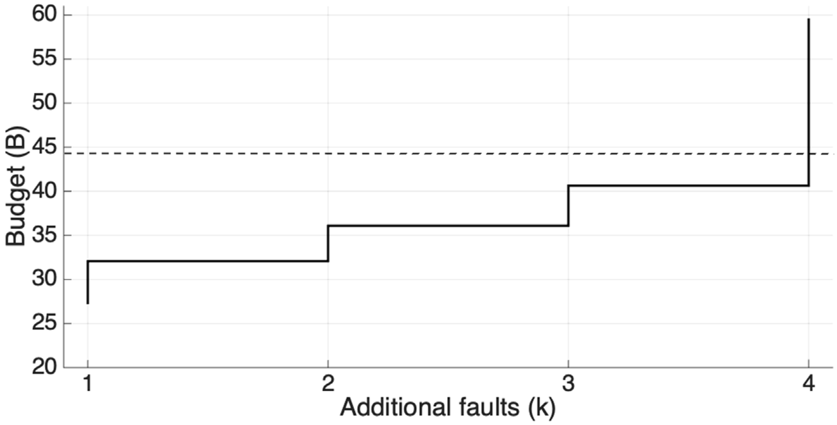

Fig. 6. Optimal test activity allocation to expose k additional faults.

Similarly, the alternative optimization formulation to minimize the budget required to expose k additional faults is illustrated by solving Eq. (26) in the context of the negative binomial model on DS1. Fig. 6 shows the budget required to expose one to four additional faults according to the model fit to the first 15 intervals. The dashed horizontal line corresponds to the 44.2 units of effort allocated in the 16th interval of DS1, which exposed two additional faults. Fig. 6, indicates that the optimal allocation to expose three additional faults is predicted to be 40.64. Thus, Eq. (26) may enable more efficient test resource allocation.

6. Conclusions and future research

This paper presents a covariate software reliability growth model and formulates the optimal test activity allocation problem to maximize fault discovery despite budget constraints or minimize the budget required to discover a specified number of faults. Expectation conditional maximization algorithms were derived and applied to two data sets from the literature. The optimal test activity allocation problem was then solved to illustrate how it can increase the number of faults discovered by distributing limited testing resources among the testing activities performed.

Possible directions for future research include: (i) the development of additional optimization problems for covariate models to guide activities during the testing process and (ii) industrial case studies to assess the relative effectiveness of alternative activities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This material is based upon work supported by the National Science Foundation, United States under Grant Number (#17496 35). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

V. Nagaraju was sponsored by the Army Research Laboratory, United States and was accomplished under Cooperative Agreement Number $W911NF-19-2-0224$. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

The authors would like to acknowledge Panlop Zeephongsekul of RMIT University who died on November 17, 2017. He was our mentor and friend. The authors would also like to thank the reviewers for their constructive comments on an initial version of the manuscript.

References

Ammann, P., Offutt, J., 2016. Introduction to Software Testing. Cambridge University Press.

Diakoulaki, D., Mavrotas, G., Papayannakis, L., 1995. Determining objective weights in multiple criteria problems: The critic method. Comput. Oper. Res. 22 (7), 763–770.

Farr, W., Smith, O., 1984. Statistical modeling and estimation of reliability functions for software (SMERFS) users guide. Tech. Rep. NAVSWC TR-84-373, Rev. 2, Naval Surface Warfare Center, Dahlgren, VA.

Fiondella, L., Gokhale, S., 2008. Software reliability models incorporating testing effort. Opsearch 45 (4), 351–368.

Fiondella, L., Gokhale, S., 2011. Software reliability model with bathtub-shaped fault detection rate. In: Annual Reliability and Maintainability Symposium. pp. Session 9D–2.

Hudepohl, J., Aud, S., Khoshgoftaar, T., Allen, E., Mayrand, J., 1996. Emerald: Software metrics and models on the desktop. IEEE Softw. 13 (5), 56–60.

Ishii, T., Fujiwara, T., Dohi, T., 2008. Bivariate extension of software reliability modeling with number of test cases. Int. J. Reliab. Qual. Saf. Eng. 15 (1), 1–17.

Kalbfleisch, J., Prentice, R., 1973. Marginal likelihoods based on Cox's regression and life model. Biometrika 60 (2), 267–278.

Khoshgoftaar, T., Bhattacharyya, B.B., Richardson, G.D., 1992a. Predicting software errors, during development, using nonlinear regression models: A comparative study. IEEE Trans. Reliab. 41 (3), 390–395.

Khoshgoftaar, T., Munson, J., 1990. Predicting software development errors using software complexity metrics. IEEE J. Sel. Areas Commun. 8 (2), 253–261.

Khoshgoftaar, T., Munson, J., Bhattacharya, B., Richardson, G., 1992b. Predictive modeling techniques of software quality from software measures. IEEE Trans. Softw. Eng. 18 (11), 979–987.

Li, H., Lu, M., Li, Q., 2006. Software reliability metrics selecting method based on analytic hierarchy process. In: IEEE International Conference on Quality Software. pp. 337–346.

Lyu, M. (Ed.), 1996. Handbook of Software Reliability Engineering. McGraw-Hill, New York, NY.

Nagaraju, V., Fiondella, L., Zeephongsekul, P., Jayasinghe, C., Wandji, T., 2017. Performance optimized expectation conditional maximization algorithms for nonhomogeneous Poisson process software reliability models. IEEE Trans. Reliab. 66 (3), 722–734.

Nagaraju, V., Shekar, V., Steakelum, J., Luperon, M., Fiondella, L., Shi, Y., 2019. Practical software reliability engineering with the software failure and reliability assessment tool (SFRAT). SoftwareX 10, 100357.

Okamura, H., Dohi, T., 2014. A novel framework of software reliability evaluation with software reliability growth models and software metrics. In: IEEE International Symposium on High-Assurance Systems Engineering. pp. 97–104.

Okamura, H., Dohi, T., 2015. Towards comprehensive software reliability evaluation in open source software. In: IEEE International Symposium on Software Reliability Engineering. pp. 121–129.

Okamura, H., Etani, Y., Dohi, T., 2010. A multi-factor software reliability model based on logistic regression. In: IEEE International Symposium on Software Reliability Engineering. pp. 31–40.

Okamura, H., Watanabe, Y., Dohi, T., 2003. An iterative scheme for maximum likelihood estimation in software reliability modeling. In: IEEE International Symposium on Software Reliability Engineering. pp. 246–256.

Rinsaka, K., Shibata, K., Dohi, T., 2006. Proportional intensity-based software reliability modeling with time-dependent metrics. In: IEEE International Conference on Computer Software and Applications, Vol. 1. pp. 369–376.

Sakamoto, Y., Ishiguro, M., Kitagawa, G., 1986. Akaike information criterion statistics. D. Reidel, Dordrecht, The Netherlands, p. 81.

Sharma, K., Garg, R., Nagpal, C., Garg, R., 2010. Selection of optimal software reliability growth models using a distance based approach. IEEE Trans. Reliab. 59 (2), 266–276.

Shibata, K., Rinsaka, K., Dohi, T., 2006. Metrics-based software reliability models using non-homogeneous Poisson processes. In: IEEE International Symposium on Software Reliability Engineering. pp. 52–61.

Shibata, K., Rinsaka, K., Dohi, T., 2015. M-SRAT: Metrics-based software reliability assessment tool. Int. J. Perform. Eng. 11 (4).

Yamada, S., Hishitani, J., Osaki, S., 1993. Software-reliability growth with a Weibull test-effort: A model and application. IEEE Trans. Reliab. 42 (1), 100–106.

Yamada, S., Ohtera, H., Narihisa, H., 1986. Software reliability growth models with testing-effort. IEEE Trans. Reliab. 35 (1), 19–23.

Zeephongsekul, P., Jayasinghe, C., Fiondella, L., Nagaraju, V., 2016. Maximum-likelihood estimation of parameters of NHPP software reliability models using expectation conditional maximization algorithm. IEEE Trans. Reliab. 65 (3), 1571–1583.

Zhao, M., Xie, M., 1993. Robustness of optimum software release policies. In: IEEE International Symposium on Software Reliability Engineering. pp. 218–225.

Vidhyashree Nagaraju is a Ph.D. candidate in the Department of Electrical and Computer Engineering at the University of Massachusetts Dartmouth (UMassD), where she received her MS (2015) in Computer Engineering. She received her BE (2011) in Electronics and Communication Engineering from Visvesvaraya Technological University in India.

Chathuri L. Jayasinghe received the B.Sc. (Hons.) degree in statistics from the University of Sri Jayewardenepura, Nugegoda, Sri Lanka, in 2007, the M.Sc. degree (with high distinction) in statistics and operations research in 2009 and the Ph.D. degree in statistics in 2013, both from the RMIT University, Australia. She was a Researcher in the School of Sciences, RMIT University and she is currently working with the University of Sri Jayewardenepura, Sri Lanka as a Senior Lecturer in statistics. Her research interests include software reliability growth models, nonparametric and parametric estimation methods, and survival analysis.

Lance Fiondella is an assistant professor in the Department of Electrical & Computer Engineering at UMassD. He received his Ph.D. (2012) in Computer Science & Engineering from the University of Connecticut. Dr. Fiondella's papers have been the recipient of twelve conference paper awards, including seven with his students. His research has been funded by the Department of Homeland Security, National Aeronautics and Space Administration, Army Research Laboratory, Naval Air Warfare Center, and National Science Foundation, including a CAREER Award.