Winter 1991

Estimation in a Marked Poisson Error Recapture Model of Software Reliability

Rajan Gupta
Old Dominion University

ESTIMATION IN A MARKED POISSON ERROR RECAPTURE MODEL OF SOFTWARE RELIABILITY

by

Rajan Gupta

M.S., University of Manitoba, 1984

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of
the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
in
COMPUTATIONAL AND APPLIED MATHEMATICS
December 1991

Approved by:

Larry D. Lee (Director)

Abstract

Estimation in a Marked Poisson Error Recapture Model of Software Reliability

Rajan Gupta

Old Dominion University, 1991

Director: Dr. Larry Lee

Nayak's (1988) model for the detection, removal, and recapture of the errors in a computer program is extended to a larger family of models in which the probabilities that the successive programs produce errors are described by the tail probabilities of discrete distribution on the positive integers. Confidence limits are derived for the probability that the final program produces errors. A comparison of the asymptotic variances of parameter estimates given by the error recapture and by the repetitive-run procedure of Nagel, Scholz, and Skrivan (1982) is made to determine which of these procedures efficiently uses the test time.

To my parents

Acknowledgements

First and foremost, I thank my advisor Larry D. Lee for the encouragement and invaluable guidance he provided during the research and writing of this dissertation.

I would also like to thank the members of dissertation committee, Ram C. Dahiya, John P. Morgan and Edward Markowski, for their helpful advice and careful reading of my thesis.

I thank NASA and the Department of Mathematics and Statistics for financial support.

I would also like to express my gratitude to all department members-staff, faculty, and fellow students- who helped to make the department a great place to work and study. Barbara Jeffery and Gayle Tarkelsen deserve special thanks for helping to simplify administrative complexities.

Cheers to Surya Dhakar for a lasting friendship which has helped to make these years so enjoyable. I especially wish to thank all faculty and staff at Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL, for sharing my part of university services during my stay in the department.

Finally, I thank all my friends and family members especially my sister Reena in Canada, for their support- academic and otherwise.

Contents

List of Tables

Chapter 1

Introduction

A great many papers have appeared concerning software reliability and reliability growth models during the debugging of a program. This is due, in part, to an early realization that the reliability of many systems may depend critically upon the reliability of computer programs; although the hardware component of these systems may be highly reliable, the total system reliability is often limited by that of the software component. Software reliability research is generally aimed at providing the capability to design and build reliable software systems in a cost-effective way.

One of the major factors contributing to the very high levels of reliability that can now be achieved in hardware systems, is the use of component redundancy to provide

tolerance to physical faults. Software redundancy techniques, such as recovery blocks and N-version programming (e.g., [1],[38]) have been proposed which aim to provide tolerance to design faults, the main cause of unreliability in software systems. These techniques are based on the implementation of functionally equivalent, but independently developed modules of diverse design, with provision for either state restoration or replication, and of some means for co-ordinating between the outputs from the replicated modules.

Some highly critical systems have relied on the construction of independently designed versions of the entire software system (e.g., Space Shuttle, A310 Airbus, railway signalling). Many database systems and telephone switching systems employ sophisticated recovery techniques which can prevent corruption of data by certain categories of software faults.

One of the more commonly used techniques for predicting software reliability utilizes reliabilty growth models. Reliability improves as a result of the process of fault identification and correction known as debugging. These models require failure data and place stringent requirements upon the testing strategy which generates the data.

To obtain more data than is provided by the usual debugging experiment, Nagel, et. al. (1984) proposed a repetitive-run procedure in which a program is restored to its original form and the debugging sequence repeated, perhaps several times, using independently generated series of inputs. The extra information provided by this procedure has been used to check the assumption of exponentially distributed inter-failure times and also to show that distinct errors may have different occurrence rates. Except for the work of Scholz (1986), models that treat specific features of this design do not seem to have been studied in the literature.

In other experiments such as seeding and tagging, inference is not based on the usual observed failure data. The method described in the following paragraphs is based on the one used for many years to estimate the size of animal and fish populations.

Feller (1957) gives a procedure for estimating the number of a certain type of fish in a lake. A new catch is made and the tagged fish as well as untagged fish are counted. Tagged fish are assumed to mix randomly with the untagged fish. The number of fish in the lake can then be estimated by assuming that the proportion of the tagged fish which are re-caught is equal to the proportion of fish in the lake contained within the second catch.

There are two ways of applying these ideas to software reliability. The first is known as seeding, originally suggested by Mills (1970), and consists of inserting a known number of bugs into a program, and monitoring the proportion of inserted bugs found during the debugging process. The second alternative is tagging, suggested by Rudner (1977) which is carried out by giving the program to two programmers. The number of faults found by the first programmer are regarded as tagged, and inferences are drawn from the proportion of the second programmer's faults which have been tagged.

There are a number of drawbacks with these methods, especially with regard to the implicit assumptions which have to be made. For example, both methods assume that all faults are equally likely to be found, seeding assumes that the seeded faults are representative of the indigenous faults, and tagging assumes that the programmers can act independently.

In seeding and tagging procedures, the estimation problem for the number of faults, however, is not onerous. The maximum likelihood estimator is biased, but a modified estimator due to Chapman (1951) has a lower bias and generally better properties, when the the number of faults is more than 50.

Another drawback is that one can not estimate quantities such as failure rates. This method is only useful for bug counting.

The most important criticism for the simple reliability growth models is that they essentially treat software as a 'black box'. No account is taken of internal structure or other known properties of the program under study. There is a need for models that can utilize the large amount of structural information usually available. Hardware reliability theory provides an interesting parallel. One of the most important achievements of this theory is the ability to combine information about component reliability with structural information about the design of the overall system. Unfortunately, software structure tends to be much more complex than hardware structure. Also, the simple component/design dichotomy is less obviously applicable to software, which can be viewed as solely levels of design.

To resolve some of the existing problems associated with models for the usual debugging experiment, Nayak (1988) introduced recapture debugging as a way to get extra information for estimating the number of faults remaining in a system. By placing counters in the software (for an alternative to software testing counters, see p. 25) we observe, in addition to the usual sequence of failure (i.e.,

error detection) times, the hitting frequencies of detected faults. Nayak's (1988) discussion concerns the Jelinski-Moranda (1972) model and procedures useful for estimating the number of remaining faults.

Chapter 2 describes the classification and motivation for a number of software reliability growth models. The main body of new material begins in Chapter 3 and concerns a family of marked Poisson process models for the recapture debugging procedure. In one form, the model describes a stationary event series and an attached Markov chain. An equivalent form, in the sense of giving the same likelihood function, is that of a nonstationary series of main events together with a collection of independent counting processes. The latter form of the model was originally suggested by Nayak (1988), although he considered only the case in which the main event series is a linear pure death process, also known as the Jelinski-Moranda (1972) model. The main contributions of the present work are (i) a procedure for estimating the probability that the final program version produces errors and (ii) a comparison of the error recapture and repetitive-run procedures in terms of the asymptotic variances of parameter estimates obtained by the two procedures. Chapter 4, the final chapter, studies, by using repetitive-run data, the goodness of fit of certain models based on parameterizations introduced in Chapter 3.

Chapter 2

Background

2.1 Introduction

The software segment of a computer system involves instructions or codes used to program the hardware system. Some of the inputs for which a specific job data set or function does not produce the desired output lead to what is termed as software failures. These failures are either due to errors in the coding of the instructions (the program) or an input that is incompatible with the design of the software system.

Early debugging designs consist of detecting and correcting a series of errors during a specified period of testing. If the software is executed on a series of inputs, it may work satisfactorily until time S_1, when the first

failure occurs. The programmer then repairs the program, it works satisfactorily until time S_2, then it is repaired, and so on. This process is sometimes referred to as a debugging experiment. The models used to describe the failure times S_1, S_2,..., are called reliability growth models since, typically the gaps $Y_i = S_i - S_{i-1}$ $(S_0 \equiv 0)$, $i=1,2,...,$ between failures will increase as faults are removed from the software.

This may not be exactly so due to the fact that the failure times are random, and thus they are subject to statistical fluctuations. A number of models have been proposed in the literature to study such failures. These models assume that failure times have distributions with parameters that depend on the residual faults in the software system. The assumed distributions reflect the software quality as faults are detected and removed from the system.

The models described in Sections 2.2-2.7 are based on assumptions concerning the failure gaps $\{Y_i\}$, the event occurrence times $\{S_1, S_2,...,\}$, or the counting process $\{N(t) : t \geq 0\}$ where $N(t)$ is determined by $N(t)=n$ if and only if $S_n < t < S_{n+1}$, with $S_0 \equiv 0$.

2.2 The Jelinski-Moranda Model

One of the earliest and most widely referenced models is that of Jelinski and Moranda (1972). The model assumes that the failure rate at any point in time is proportional to the current fault content of the program. The initial fault content is denoted by ν and the contribution of each fault by ϕ. The failure rate initially is $\nu\phi$, and decreases to $(\nu-1)\phi$ after the first fault is detected and eliminated, and so on. This model views the occurrence times of the first, second, ..., and rth failure as the first r order statistics, in a sample of size ν from an exponential distribution. The times between failures are distributed as independent exponential random variables with rate parameters $(\nu-i+1)\phi$, $i=1,2,\ldots,\nu$.

Moranda (1979) points out that most models assume that a system is restored to its initial state after repair. For reliability growth models where the failure rate varies, it is generally assumed that this rate changes deterministically or continuously with time. As noted by Littlewood (1981), the failure rate of J-M model changes in discrete steps and this is an essential feature to represent reliability growth.

An inherent assumption of the model is that each bug contributes the same amount ϕ to the overall failure rate. This has been criticized in particular by Littlewood (1981) who argues that the different frequencies of execution of different portions of code will in itself lead to different occurrence rates of faults, all other things being equal. The assumptions of constant failure rate and exponential distributions for the times between successive failures have also been criticized by Schick and Wolverton (1978).

Forman and Singpurwalla (1979) show that the maximum likelihood estimator of ν is unstable, and can be highly misleading when the number of remaining faults is anything but small. Sukert (1977) has observed that the estimate of ν does not always exist, and Littlewood and Verrall (1981) give a condition for its existence which reduces to a requirement that the data exhibit the assumed reliability growth. Littlewood and Verrall (1973) suggest a Bayesian version of the model, but they note that the improvement in estimation properties is marginal in most cases.

2.3 The Musa Execution Time Model

This is perhaps the most practical model of all and has been developed to the extent that it takes into account

various personnel and resource constraints.

The model is based on the Jelinski-Moranda (1972) model, but introduces a number of refinements. One of these is an error reduction factor B representing the average ratio of the rate of fault correction to the failure rate. There is an implicit assumption, that the fault correction rate is proportional to the failure rate - not equal to it as assumed by the Jelinski-Moranda (1972), Shooman (1973), Schick-Wolverton (1978) and Littlewood (1980) models. Also introduced is a testing compression factor C which is the average ratio of the failure rate during test to that during operational use.

This model also differs from others in its treatment of time. The program execution time is taken as the time variable, and this in turn is related to calendar time via constraints on fault correction personnel, and computer time. These aspects make the model potentially valuable as a management tool.

The assumptions of this model answer some of the criticisms of the Jelinski-Moranda (1972) model, but the underlying assumption that each fault contributes the same amount to the overall failure rate is often considered to be a weakness. Chenoweth (1981) suggests a generalization where

$$B = B_0 e^{at}$$

to allow for improved debugging in the later stages.

Since the basic model structure is identical to that of the Jelinski-Moranda model, the same parameter estimation problems are encountered.

2.4 Poisson Models

If each fault is removed the first time it produces a failure, then the failure rate will decrease as in the previous models, but it may depend on the time of the most recent repair rather than on the total number of repairs that have been made. This is a basic assumption underlying time dependent Poisson process models.

A nonhomogeneous Poisson process $\{N(t): t \geq 0\}$ with intensity function $\lambda(x)$ (e.g., Parzen, p. 252) is defined by the following conditions:

(i) $N(0)=0$.

(ii) $N(t)$, $t \geq 0$ has independent increments.

(iii) The number of events $N(s,t)$ occuring in the interval (s,t) has a Poisson distribution with mean $\int_s^t \lambda(x)dx$.

These assumptions imply that the occurrence time S_1 of the first event has distribution

$$P(S_1 > t) = P(N(t)=0)$$

$$= \exp\left(-\int_0^t \lambda(x)dx\right), \quad t \geq 0$$

$$= 1, \quad t < 0$$

By varying $\lambda(x)$ over the entire class of nonnegative functions that are integrable over bounded intervals and which satisfy

$$\lim_{0}^{t} \int \lambda(x)dx = \infty, \text{ for } t \to \infty$$

we obtain all possible continuous distributions for S_1. The importance of the nonhomogeneous Poisson process lies in the fact that the increments of the process are not required to be stationary. Most applications of the model assume that $\lambda(t)$ is either constant, increasing, or decreasing in $t \geq 0$. The mean number of events occurring in the interval $(0,t)$, denoted by $m(t)$, is given by

$$m(t) = \int_0^t \lambda(x)dx$$

If events occur at times $S_1, S_2, \ldots,$ over a fixed interval $(0,t)$ and follow a non-homogeneous Poisson process, then the joint density function of $S_1, S_2, \ldots, S_{N(t)}$ and $N(t)$ is

$$\prod_{i=1}^{n} \lambda(s_i) \exp[-\int_0^t \lambda(x)\,dx] \qquad (2.1)$$

$$(0< s_1 < s_2 <\ldots< s_n < t,\ n=0,1,2,\ldots\ )$$

where if $n=0$, the product factor is equal to one.

The most widely known Poisson model of software reliability growth was proposed by Goel and Okumoto (1979). Their model assumes an intensity function and mean of the form

$$\lambda(x) = \alpha\beta e^{-\beta x} \quad , \ x \geq 0, \ \alpha, \beta > 0$$

$$m(t) = \alpha(1-e^{-\beta t}), \ t \geq 0, \ \alpha, \beta > 0 \qquad (2.2)$$

Since $\lim m(t) = \alpha$ as $t \rightarrow \infty$, the parameter α is the expected total number of failures that may eventually occur, while large values of β imply rapid reliability growth.

Under this model the conditional joint density function of $S_1, S_2, \ldots, S_{N(t)}$, given $N(t)=n$, is

$$n! \prod_{i=1}^{n} \beta e^{-\beta s_i}(1-e^{-\beta t})^{-1}, \quad 0< s_1 < s_2 <\ldots< s_n < t \tag{2.3}$$

Thus $S_1, S_2, \ldots, S_n$ are conditionally distributed as an ordered sample of size n from a truncated exponential distribution with density function

$$\beta e^{-\beta x}(1-e^{-\beta t}), \quad 0< x < t \tag{2.4}$$

The J-M model has this same property and thus the likelihood functions given by the two models only differ in the probability distribution of N(t). That is, the J-M model implies that N(t) has a binomial distribution with parameters ν and $(1-e^{-\beta t})$, whereas any Poisson model implies that N(t) has a Poisson probability distribution.

2.5 Littlewood-Verrall Bayesian Model

Littlewood and Verrall (1973) consider the software failure process to be the result of two sources of uncertainty. One pertains to the randomness of input and the other to the state of the program. They assume that a particular subset of the input space that will cause system

failure will be encountered randomly. This leads to the assumption that successive times between failures are independently and exponentially distributed with failure rate λ_i for the ith failure. The second source of uncertainty is the fault-fixing operation. The uncertainty of fault-fixing is modeled by additionally assuming that λ_i is stochastically less than or equal to λ_{i-1}, that is,

$$P\{\lambda_i < x\} \geq P\{\lambda_{i-1} < x\}, \text{ for all } x, i.$$

The growth in reliability is thus stochastic rather than deterministic, the λ_i being regarded as random variables. Littlewood and Verrall assume that each λ_i has a gamma distribution with parameters α and $\{\psi_i\}$, where the growth function ψ_i is increasing in i and describes the quality of the programmer and the difficulty of the programming task.

Combining the two sources of randomness by Bayesian techniques leads to a decreasing failure rate, and a low failure rate as more time is observed without failure. The assumptions of this model seem reasonable, though Ramamoorthy and Bastani (1980) criticize the model as being too restrictive, because it does not permit perfect debugging (i.e., unlike for the the J-M model, removing a finite number of faults will not, according to the model, produce a perfect

program). One main problem with this model is the choice of $\{\psi_i\}$, which determines the stochastic growth pattern. It is not clear how to choose the functional forms of $\{\psi_i\}$, however Littlewood (1980) has suggested linear and quadratic functions, which lead to two and three parameter models.

2.6 Order Statistics Models

Besides the Jelinski-Moranda model a number of other models have been derived that belong to the order statistics class of models (e.g., Miller 1986). These derivations are typically based on the assumption that the failure gaps $\{Y_i\}$ are independent and exponentially distributed with rate parameters λ_i, where the λ_i are also random quantities.

Using Bayes' theorem, Littlewood (1973) derived a generalization of the Jelinski-Moranda model in which the ordered failure times S_1, S_2,..., S_n are distributed as the first n order statistics in a sample of size ν (ν is a parameter) from a Pareto distribution. Arguing from simpler assumptions, he shows that if the ith failure occurs at time $S_i=\tau$, then the failure rate Λ is the sum of $\nu-i$ *i.i.d* random variables, each having a gamma density with scale parameter $\beta+\tau$ and shape parameter α. Then Λ also has a gamma density

$$g(\lambda \mid \tau) = (\beta+\tau)^{(\nu-1)\alpha}\lambda^{(\nu-1)\alpha-1}\exp[-(\beta+\tau)\lambda]/\Gamma[\alpha(\nu-i)], \quad (\lambda \geq 0,\ \beta>0) \qquad (2.5)$$

and the distribution of Y_{i+1} given $S_i=\tau$, is

$$P(Y_{i+1} > y \mid S_i=\tau) = \int_0^\infty e^{-\lambda y} g(\lambda \mid \tau)\, d\lambda$$

$$= [(\beta+\tau)/(\beta+\tau+y)]^{(\nu-1)\alpha},\ y \geq 0 \qquad (2.6)$$

Assuming that the conditional distribution of Y_{n+1}, given the past history $S_1, S_2, \ldots, S_n$, only depends on the time S_n of entering the nth state, Littlewood also obtains the joint density of $S_1, S_2, \ldots, S_n$, which is identical to that of the first n order statistics in a sample of size ν from a Pareto distribution.

This Pareto order statistics model differs from the model in Section 2.5 in that (a) the gaps $\{Y_i\}$ are not independent and (b) the Littlewood-Verrall (1973) model of Section 2.5 does not include the total error count parameter ν. Joe and Reid (1985a,b) give an alternative, and simpler, derivation of the Pareto order statistics model and conjecture that, for any order statistics model, the maximum likelihood estimate of ν will be infinite with positive probability.

2.7 Scale Parameter Models

A key assumption made in a number of models is that the failure gaps $\{Y_i\}$ are independent and have distributions that differ only by scale parameters, the latter being a function of the serial index of events. A scale parameter family of reliability growth models is defined by letting $\{Y_i\}$ be independent with distribution functions

$$F_i(y) = G(\psi_i^{-1} y) \tag{2.7}$$

where $\{\psi_i\}$ are scale parameters and $G(x)$ is a continuous cdf with $G(0)=0$.

Examples of models within this class, the first two of which were described in Sections 2.2 and 2.5, are the following:

(i) $\psi_i^{-1} = (\nu - i + 1)\phi, \quad i = 1, 2, \ldots, \nu, \quad \phi > 0$

$G(x) = 1 - \exp(-x), \quad x \geq 0$

(i.e., the Jelinski-Moranda (1972), model)

(ii) $\psi_i = \alpha + \beta i, \quad \alpha > 0, \quad \beta > 0$

$G(x) = 1 - (1 + x)^{-\rho}, \quad x \geq 0, \quad \rho > 0$

(Littlewood and Verrall, 1973)

(iii) $\psi_i^{-1} = \alpha \exp(-\phi i)$, α, $\phi > 0$

$G(x) = 1 - \exp(-x)$, $x \geq 0$

(Cox and Lewis, 1966; Moranda, 1975)

To show that (i) and (ii) imply a limitation on reliability growth, let $Z_1, Z_2, \ldots,$ be *i.i.d* with cdf $G(x)$. Further, let

$$q_{ij} = \log\ (\psi_j/\psi_i),\ 1 \leq i<j \qquad (2.8)$$

and consider the following representation implied by (2.7):

$$\log(Y_j/Y_i) = q_{ij} + \log\ (Z_j/Z_i),\ 1 \leq i<j \qquad (2.9)$$

If each q_{ij} is bounded above by a known constant, say q_{ij}^0, that depends on i and j, then the distribution of $\log(Y_j/Y_i)$ is shifted below that of

$$q_{ij}^0 + \log\ (Z_j/Z_i)$$

The latter implies

$$P(Y_j > Y_i \exp(q_{ij}^0)) = P(\log\ (Y_j/Y_i) > q_{ij}^0)$$

$$= P(q_{ij} + \log\ (Z_j/Z_i) > q_{ij}^0)$$

$$\leq P(Z_j > Z_i)$$

$$= \int_0^{\infty} [1-G(x)]dG(x) = 0.5 \qquad (2.10)$$

Thus for $i < j$, Y_j can exceed a certain scalar multiple of Y_i with probability not exceeding 0.5.

To determine the constant q_{ij}^0 in case (i), write

$$q_{ij} = \log [(1 - \gamma i)(1 - \gamma j)^{-1}] \qquad (2.11)$$

where, $\gamma = (\nu + 1)^{-1}$, and $i, j \leq \nu$ implies $\gamma i \leq 1$ and $\gamma j \leq 1$. Since q_{ij} is nondecreasing in γ, and $\gamma \leq (j+1)^{-1}$ it follows that $q_{ij}^0 = \log(j-i+1)$ is an upper bound on q_{ij}.

Similarly, in case (ii),

$$q_{ij} = \log [(1 + \gamma j)(1 + \gamma i)^{-1}], \quad 1 \leq i < j \qquad (2.12)$$

where $\gamma = \beta/\alpha$ and q_{ij} is nondecreasing in γ. In this case the maximum value, obtained in the limit as $\gamma \rightarrow \infty$, is $q_{ij}^0 = \log(j/i)$.

The existence of known upper limits implies a limitation on reliability growth since the intervals between failures are unlikely to increase rapidly if the model giving the upper limit is the true model. However, the simplicity of

these models is attractive and they can be a useful tool for estimating the reliability achieved during debugging.

2.8 Stopping Rules

Stopping rules for deciding when a program is completely debugged have been proposed by Nayak (1988) and Goudie (1990). The context of their discussion, described in more detail in Chapter 3, is a Markov chain (X_n, Y_n) where X_n is the number of distinct errors detected by the nth epoch and $Y_n = n - X_n$. A transition occurs from $(X_n, Y_n) = (x, y)$ to $(x+1, y)$ with probability $(\nu-x)/\nu$, or to $(x, y+1)$ with probability x/ν. The parameter ν is the initial number of errors in a program and thus debugging must terminate after a fixed number of transitions occur. A well known property of Markov chains is that the waiting times W_i in states $i=1,2,\ldots,\nu-1$ (i.e., between distinct error occurrences or, equivalently, between changes in X_n) are independent random variables with geometric distributions.

Nayak (1988) proposed deciding that all errors have been eliminated when W_i first exceeds a positive integer k_i chosen so that

$$\alpha \geq P(W_i > k_i;\ \nu = i+1) \qquad (2.13)$$

where α is a prechosen error level. Since

$$P(W_1 > k_1;\ \nu = i+1) = [i/(i+1)]^x$$

(2.13) is satified if $k_i = [a_i] + 1$,

where

$$a_i = (\log \alpha) / [\log i - \log(1+i)] \qquad (2.14)$$

and [x] is the integer part of the real number x. Nayak (1988) determined the probability of correcting all errors and tabled these probabilities for various values of α and ν.

Since $P(X_1 = 1) = 1$, the likelihood function based on observing $W_1, W_2, \ldots, W_{x-1}$ and $X_n = x$ is

$$L(\nu) = \{ \prod_{j=1}^{x-1} (j/\nu)^{W_j - 1} (1 - j/\nu) \} (x/\nu)^{W_x} \qquad (2.15)$$

where $w_x = n - 1 - (w_1 + w_2 + \ldots + w_{x-1})$. This takes the simpler form (Goudie, 1990)

$$L(\nu) = c \binom{\nu}{x} / \nu^n, \qquad \nu = x, x+1, \ldots, \qquad (2.16)$$

where c does not depend upon ν.

Goudie's (1990) stopping rule is to decide that all errors have been eliminated when

$$L(x)/L(x+1) = (x+1)^{n-1}/x^n$$

achieves or exceeds a preassigned level A. Goudie (1990) derives the probability of correcting all errors and shows that the true error level is closely approximated by A^{-1}. In comparison to Nayak's (1988) procedure, the likelihood-based rule yields a small reduction in the average time taken to reach a decision.

For the usual debugging procedure, Forman and Singpurwalla (1977) propose an empirical stopping rule based upon a relative likelihood function. In a similar context Ross (1985) studies a stopping rule which takes into account the error level and provides an upper bound on the proportion of the time testing terminates with the total failure rate exceeding a prior chosen constant.

Chapter 3

Estimation in a Family of Marked Poisson Error Recapture Models

3.1 Introduction

To describe the detection, removal, and recapture of the errors in a computer program, Nayak (1988) assumed that the first occurrence times follow the Jelinski-Moranda (1972) model and that each of the errors again occur according to independent homogeneous Poisson processes with a common rate parameter ϕ. Let $\pi_1, \ldots, \pi_R$ denote the sequence of programs obtained by correcting errors in an initial program π_0 at times $S_1, S_2, \ldots, S_R$. A comparative method, known as back-to-back testing (e.g., Vouk, 1990), may be used to observe repeated error occurrences; e.g., if an error is detected

in π_{i-1} at time S_i, then comparing the outputs of π_{i-1} and π_i gives the number of times this error again occurs during the remaining test time. Since π_{i-1} and π_i differ only by the correction made at time S_i, any differences in their outputs are due to the fault that resides in π_{i-1}, which has been corrected in π_i. This method continuously replicates error detection and thus it is likely to yield more data per unit of test time than other designs.

Empirical evidence (Nagel, Scholz, and Skrivan, 1982) indicates that errors may occur with different probabilities and that errors with the highest occurrence rates are likely to be detected early. Since errors are seldom detected in the final program, inference about its reliability must be based on the error frequencies observed in the previous versions.

In this chapter we consider a family of marked Poisson process models in which the first and subsequent error occurrences are described by a Markov chain. The transition probabilities (Section 3.2) are determined by a discrete distribution G with tail probabilities $\bar{G}(i, \theta)$, which are also the probabilities that the successive programs produce errors. Nayak's (1988) model assumes G is a discrete uniform distribution with mass at $1,2,\ldots,\nu$ where ν is an integer parameter that represents the number of errors in the initial

program π_0. The family of models considered in Section 3.2 does not embed Nayak's model, although it is a more robust family since $\bar{G}(i,\theta)$ may decrease in i=0,1,2,.. at different rates for particular cases.

The probability $\bar{G}(R,\theta)$ that the final program produces errors necessarily depends on the number R of faults eliminated during the period of testing. In this way, the problem of estimating $\bar{G}(R,\theta)$ is analogous to that of estimating the number ν-R of remaining errors in Nayak's model. An estimate of $\bar{G}(R,\theta)$ is $\bar{G}(R,\hat{\theta})$ where $\hat{\theta}$ is the maximum likelihood estimator of the parameter vector θ. In Section 3.3, a scaled logarithmic function of $\bar{G}(R,\theta)$ and $\bar{G}(R,\hat{\theta})$ is shown to have a limiting distribution identical to that of a linear function of $N^{1/2}(\hat{\theta}-\theta)$ where N is the number of events observed during the period of testing. Confidence levels for estimating $\bar{G}(R,\theta)$ are obtained by simulation and compared with the nominal confidence level given by the limiting normal distribution.

Similar, though usually different models have been used in a biological context (Sandland and Cormack, 1984, Huggins, 1989, Goodman, 1953) to estimate the size of animal populations. The model studied by Sandland and Cormack (1984) seems most closely related to the model studied in this chapter, although it does not describe the time

dependency of subsequent error occurrences. A more recent paper by Nayak (1991) considers a model derived from the superposition of independent homogeneous Poisson processes.

A comparison of the asymptotic variances of parameter estimators given by recapture debugging and by the repetitive-run procedure of Nagel, Scholz, and Skrivan (1982) is made in Sections 3.5 and 3.6 to determine which of these procedures efficiently uses the test time. The comparison assumes identical models for the first occurrence times and that testing is performed for time periods of equal length under the two testing strategies.

3.2 The Model and Likelihood Function

The first and subsequent error occurrences can be modeled by a Markov chain $\{(R_i, Z_i)\}$ where R_i is the number of distinct errors detected by the ith epoch and Z_i is the error state at the ith epoch. At each epoch either a distinct new error is detected or a previously detected error again occurs. From state $(R_i, Z_i)=(r,z)$, $z=1,2,\ldots,r$, a transition occurs to state $(r+1,r+1)$ (a new error is detected) with probability $\bar{G}(r,\theta)$ or to state (r,i), $i \leq r$, whenever a previous error again occurs, with probability $g(i,\theta)$. The

times T_1, T_2, ..., at which these events occur are assumed to follow a homogeneous Poisson process, where $\{(R_i, Z_i)\}$ and $\{T_i\}$ are assumed to be independent. As a consequence of our assumption that $\{T_i\}$ is a homogeneous Poisson process, the number N of events occurring in a fixed interval (0,t) has a Poisson distribution with mean αt. The model for the event occurrence times and the first and subsequent error ocurrences can be specified by the initial error rate α and the distribution function $G(x,\theta)$ of a discrete random variable with mass $g(i,\theta)$ on the positive integers. The survivor function $\bar{G}(x,\theta)=1-G(x,\theta)$ determines the probability $\bar{G}(i,\theta)$ that the ith program produces errors.

Let B denote the set of epochs at which previously detected errors again occur; that is, $B=\{i: z_i \leq r_{i-1}, i=1,2,\ldots,n\}$. The likelihood function based on observing N=n and (R_i, Z_i, T_i), $i=1,2,\ldots,n$ is

$$\alpha^n e^{-\alpha t} \prod_{i \in B} g(z_i,\theta) \prod_{i=1}^{r} \bar{G}(i-1,\theta) = L_1(\alpha) L_2(\theta), \quad (3.1)$$

$$(0<t_1<\ldots<t_n<t,\ 1=r_1 \leq r_2 \leq \ldots \leq r_n \leq n,$$

$$z_i=1,2,\ldots,r_i,\ i=1,2,\ldots,n),$$

where $L_2(\theta) = 1$ if n=0 and where $R_n = r$ is the number of distinct

errors detected during the first n epochs; that is, during the interval (0,t).

The hitting frequencies $M_1, M_2, \ldots,$ of the first, second, etc., detected faults are given by $M_i = \sum_{j \in B} I(Z_j=i)$, where $I(A)$ is the indicator function of the set A. Since M_i does not include the first time that the ith detected error occurs, we have $\sum_{i=1}^{R} M_i + R = N$. In terms of $M_1, M_2, \ldots, M_R$, (3.1) can be written

$$\alpha^n e^{-\alpha t} \prod_{i=1}^{r} [g(i,\theta)]^{m_i} \bar{G}(i-1,\theta). \qquad (3.2)$$

It should be noted that, (3.2) reduces to the likelihood function studied by Nayak (1988) when G is a discrete uniform distribution with mass at $1,2,\ldots,\nu$ where ν is an integer parameter.

Let $n_{i0} < n_{i1} < \ldots < n_{i\,m(i)}$ denote the ordered elements of $\{j: z_j=i,\ j=1,2\ldots,n\}$, $i=1,2,\ldots,r$. Then $S_{ij} = T_{n(i,j)}$ defines the first occurrence times $S_{10} < S_{20} < \ldots < S_{r0}$, hereafter denoted by $S_1, S_2, \ldots, S_r$, and also the times $S_{i1} < S_{i2} < \ldots < S_{i\,m(i)}$ at which the ith detected error again occurs during the interval (S_i,t), $i=1,2,\ldots,r$.

The Jacobian of this transformation is equal to one and thus the joint density function of $(R, M_1, M_2, \ldots, M_R, S_1,$

$S_2, \ldots, S_R, S_{11}, S_{12}, \ldots, S_{1\,M(1)}, \ldots, S_{R1}, S_{R2}, \ldots, S_{R\,M(R)})$ is

$$\alpha^n e^{-\alpha t} \prod_{i=1}^{r} [g(i,\theta)]^{m_i} \bar{G}(i-1,\theta)$$

$$(n = r + \sum_{i=1}^{r} m_i,\ m_i = 0,1,2,\ldots,\ 0<s_1<s_2<\ldots<s_r<t,$$

$$s_i<s_{i1}<s_{i2}<\ldots<s_{i\,m(i)}<t,\ i=1,2,\ldots,r)$$

The marginal density function of $(R, M_1, M_2, \ldots, M_R, S_1, S_2, \ldots, S_R)$ is

$$\alpha^n e^{-\alpha t} \prod_{i=1}^{r} [g(i,\theta)]^{m_i} \bar{G}(i-1,\theta)\,(t-s_i)^{m_i}/m_i! \qquad (3.3)$$

$$(0<s_1<s_2<\ldots<s_r<t,\ m_i=0,1,2,\ldots,$$

$$i=1,2,\ldots,r,\ n = r + \sum_{i=1}^{r} m_i,\ r=1,2,\ldots)$$

and is $e^{-\alpha t}$ if $r=0$. Since $N = R + \Sigma_{i=1}^{R} M_i$, it follows from (3.3) that $(R, M_1, M_2, \ldots, M_R)$ is a sufficient statistic and thus applications of the model do not require observation of the event occurrence times. The latter information, however, may be useful for checking whether the spacings $Y_i = S_i - S_{i-1}$, $(S_0 \equiv 0)$ have nonexponential distributions.

Our discussion concerns the family of models defined by

$$\bar{G}(i,\theta) = \exp(-\theta_1 a_{1i} - \theta_2 a_{2i}),\ i=0,1,2,... \qquad (3.4)$$

$$g(i,\theta) = [1-\exp(-\theta_1 b_{1i} - \theta_2 b_{2i})]\ \exp(-\theta_1 a_{1,i-1} - \theta_2 a_{2,i-1}),$$

$$i=1,2,....,$$

where $\theta_i > 0$, $i=1,2$ are unknown parameter values, $b_{1i} = a_{1i} - a_{1,i-1}$, $b_{2i} = a_{2i} - a_{2,i-1}$, and $\{a_{1i}\}$, $\{a_{2i}\}$ are known constants that satisfy $a_{10} = a_{20} = 0$, a_{1i}, a_{2i} are nondecreasing in $i=0,1,2,...$, and $\lim a_{1i} = \lim a_{2i} = \infty$, as i tends to infinity. The latter conditions are implied by the requirement that $\bar{G}(i,\theta)$ be a survivor function. Particular cases of (3.4) are $a_{1i} = i$, $a_{2i} = i^2$ or $a_{1i} = i$, $a_{2i} = \log(1+i)$.

In Section (3.4), we note the equivalence of the model defined in the present Section to another form of the model in which the gaps $Y_i = S_i - S_{i-1}$, $(S_0 \equiv 0)$ between the first occurrence times have independent exponential distributions with rate parameters $\lambda_i = \alpha \bar{G}(i-1, \theta)$. In the latter context, setting $\theta_2 = 0$ and $a_{1i} = i$ in (3.4) gives the log linear rate model studied by Moranda (1975) and Cox and Lewis (1966).

Since the second factor of (3.2) does not depend on α, the maximum likelihood estimate of α is $\hat{\alpha} = n/t$. With $g(i,\theta)$

given by (3.4), the log likelihood function given by the second factor of (3.2) is

$$l_2(\theta) = \sum_{i=1}^{r} m_i \log[1-\exp(-\theta_1 b_{1i} - \theta_2 b_{2i})] - \theta_1 \sum_{i=1}^{r} (m_i+1)\, a_{1,i-1} - \theta_2 \sum_{i=1}^{r} (m_i+1)\, a_{2,i-1} \quad (3.5)$$

If $n \leq 1$, then $r \leq 1$, $m_1=0$ and $l_2(\theta)$ is constant in θ. If $n>1$ and $r=1$, then (3.5) takes its maximum value at $\theta_i = \infty$, $i=1,2$. Otherwise, $l_2(\theta)$ is concave in θ (see Appendix A), and an estimate $\hat{\theta}$ that maximizes (3.5) is the unique solution to

$$\sum_{i=1}^{r} m_i b_{ji} [\exp(\theta_1 b_{1i} + \theta_2 b_{2i}) - 1]^{-1} = \sum_{i=1}^{r} (m_i+1)\, a_{j,i-1}, \quad (j=1,2) \quad (3.6)$$

3.3 Confidence Limits

An estimate of the probability that the final program produces errors is $\bar{G}(R, \hat{\theta})$ where $\hat{\theta}$ is the maximum likelihood estimate of θ. Let $a_i = a_{1i} + a_{2i}$. Then

$$N^{1/2} a_R^{-1} [\log \bar{G}(R, \hat{\theta}) - \log \bar{G}(R, \theta)]$$

$$= -N^{1/2} a_R^{-1} [(\hat{\theta}_1 - \theta_1)\, a_{1R} + (\hat{\theta}_2 - \theta_2)\, a_{2R}] \quad (3.7)$$

In a later Section we show that (3.7) has a limiting ($t\rightarrow\infty$) normal distribution with mean zero and variance $a' I_{\theta}^{-1} a$ where $a'=(a_1, a_2)$, $a_1= \lim a_{1i}/a_i$, $a_2= \lim a_{2i}/a_i$, as $i\rightarrow\infty$, and I_{θ}^{-1} is the covariance matrix of the joint limiting distribution of $N^{1/2}(\hat{\theta}_1 - \theta_1)$ and $N^{1/2}(\hat{\theta}_2 - \theta_2)$. This assumes that the limits a_1 and a_2 are finite, which is true if $a_{1i}=i$, $a_{2i}= i^2$ and for other models within the family defined by (3.4). The basic idea behind (3.7) is related to the δ-method as discussed, for example, by Rao (1973, pp. 385-388).

Approximate 100(1-p) percent confidence limits (L_1, U_1) for $\bar{G}(R, \theta)$ are

$$L_1 = \bar{G}(R, \hat{\theta})\ \exp(-N^{-1/2} a_R\ \hat{\gamma}^{1/2}\ Z_{1-p/2})$$

$$U_1 = \bar{G}(R, \hat{\theta})\ \exp(\ N^{-1/2} a_R\ \hat{\gamma}^{1/2}\ Z_{1-p/2})$$

where $Z_{1-p/2}$ is the upper 1-p/2 percentage point of the standard normal distribution. Since R diverges ($t \rightarrow \infty$) in probability to infinity, γ can be consistently estimated by $\hat{\gamma}= b' I_{\hat{\theta}}^{-1} b$, where $b'=(b_1, b_2)$, and $b_1= a_{1R}/a_R$, $b_2= a_{2R}/a_R$.

The probability that the final program produces no errors during any subsequent time period of length y also depends on R and is

$$Q_R = \exp(-\lambda_{R+1} y), \quad y > 0$$

where, $\lambda_1 = \alpha \bar{G}(i-1, \theta)$. Letting $\hat{Q}_R = \exp[-\hat{\alpha} \bar{G}(R, \hat{\theta}) y]$, we have

$$N^{1/2} a_R^{-1} \log[(\log \hat{Q}_R)/(\log Q_R)] = N^{1/2} a_R^{-1} (\log \hat{\alpha} - \log \alpha)$$

$$- N^{1/2} a_R^{-1} [(\hat{\theta}_1 - \theta_1) a_{1R} + (\hat{\theta}_2 - \theta_2) a_{2R}] \qquad (3.8)$$

where since a_R diverges to infinity in probability, the first term on the right of (3.8) converges in probability to zero and the second term has the same limiting normal distribution as the quantity in (3.7). Confidence limits for $-\log Q_R$ are

$$L_2 = (-\log \hat{Q}_R) \exp[-N^{-1/2} a_R \hat{\gamma}^{1/2} Z_{1-p/2}]$$

$$U_2 = (-\log \hat{Q}_R) \exp[N^{-1/2} a_R \hat{\gamma}^{1/2} Z_{1-p/2}]$$

Table 3.1 shows simulated percentages of the time that the confidence limits cover $\bar{G}(R,\theta)$ and Q_R. The simulated percentages fall close to the nominal 95 percent level when E(N) is large, and are sometimes about 10 percentage points below the 95 percent level when E(N) is small. Since $\bar{G}(R, \theta)$ and Q_R will tend to take values at the extreme endpoints of the interval (0,1) whenever E(N) is large, the quantities being estimated are necessarily extreme values as $t \to \infty$. As

Table 3.1: Empirical confidence levels for estimating $\bar{G}(R,\theta)$ and Q_R, y=100; the nominal level is 0.95 with 1,000 replications; $a_{1i} = i$. $a_{2i} = i^2$; E(N), E(R), Q_c, and $\bar{G}(c,\theta)$, c = E(R) are theoretical values not obtained by simulations.

						Percent Coverage		Average Width	
E(N)	θ_1	θ_2	E(R)=c	$\bar{G}(c,\theta)$	Q_c	$\bar{G}(R,\theta)$	Q_R	$\bar{G}(R,\theta)$	Q_R
100	0.02	0.00	64.2	0.2769	0.758	87.3	86.4	0.1445	0.0923
	0.10	0.01	14.6	0.0275	0.973	89.2	87.8	0.0582	0.0548
	0.20	0.01	11.8	0.0235	0.977	87.4	86.6	0.0271	0.0258
	0.30	0.01	9.9	0.0127	0.987	86.0	85.5	0.0153	0.0147
	0.10	0.02	11.8	0.0190	0.981	90.6	90.1	0.0448	0.0427
	0.20	0.02	10.2	0.0162	0.984	89.2	89.1	0.0237	0.0226
300	0.10	0.01	17.6	0.0078	0.992	93.0	92.4	0.0134	0.0132
	0.20	0.01	14.5	0.0067	0.993	91.3	91.6	0.0072	0.0070
	0.30	0.01	12.2	0.0058	0.994	90.7	90.8	0.0144	0.0142
	0.10	0.02	14.5	0.0035	0.996	93.2	93.7	0.0097	0.0096
	0.20	0.02	12.1	0.0048	0.995	93.7	93.4	0.0053	0.0052
500	0.10	0.01	18.9	0.0042	0.996	94.0	94.8	0.0066	0.0066
	0.20	0.01	15.6	0.0039	0.996	94.1	93.8	0.0037	0.0036
	0.30	0.01	13.2	0.0033	0.997	92.2	91.6	0.0024	0.0024
	0.10	0.02	15.6	0.0016	0.998	94.8	94.7	0.0047	0.0046
	0.20	0.02	12.9	0.0027	0.997	95.1	94.7	0.0026	0.0026

shown in Table 3.1, the widths of confidence limits are similar in magnitude to $\bar{G}(c,\theta)$ and Q_c where $c=E(R)$, although the confidence limits are not designed to cover the latter quantities.

The simulations were performed by generating 1,000 realizations of the sufficient statistic $(N,R, M_1, M_1,\ldots,M_R)$. Each replicate requires generating a realization $N=n$ of a Poisson random variable having mean αt. If $R_n \leq 1$, then $\hat{\theta}_n$ is undefined; if this event occurs, it is counted as one trial for which the confidence limits do not include the quantity being estimated. If $n>1$, the sufficient statistic can be written as a function of i.i.d random variables $X_1, X_2,\ldots,$ X_n with density function $g(i,\theta)$. That is,

$$R_0=0,\ R_1=1,\ \text{and}\ R_i= R_{i-1} + I(X_i > R_{i-1}),\ i=2,3,\ldots,n$$

$$M_i = \sum_{j=1}^{n} I(X_j \leq R_{j-1},\ X_j =i),\ i=1,2,\ldots,R_n.$$

From this it follows that the sufficient statistic can be computed sequentially from independent random variables having a uniform distribution on the interval (0,1).

3.4 An Equivalent Form of the Model

We now show that the following assumptions give a likelihood function identical to (3.3).

(i) The spacings $Y_i = S_i - S_{i-1}$, $(S_0 \equiv 0)$ between the first occurrence times are independent random variables with density functions $\lambda_i \exp(-\lambda_i y)$, $y \geq 0$ where, $\lambda_i = \alpha \bar{G}(i-1, \theta)$.

(ii) Counts $M_i = M_i(S_i, t)$ of subsequent error occurrences have distributions determined by a collection $\{M_i(t)\}$ of independent homogeneous Poisson processes with rate parameters $\xi_i = \lambda_i - \lambda_{i+1}$.

(iii) $\{S_i\}$ and $\{M_i(t)\}$ are independent collections of random variables.

This form of the model, originally suggested by Nayak (1988), explicitly describes the first occurrence times and thus it directly relates error recapture models to the more common reliability growth models.

By interpreting λ_i as the hitting rate of the remaining faults after i-1 faults have been corrected, $\xi_i = \lambda_i - \lambda_{i+1}$ is then the change in this rate due to correcting the ith

detected error and ξ_1, ξ_2, ... can be interpreted as the hitting rates of the first, second, etc., detected faults. These are average rates in that the errors that are detected first, second, etc., may vary from one repetition of the experiment to another. The model assumes that the effect of correcting faults is additive in that $\xi_1 + \xi_2 + \ldots + \xi_r + \lambda_{r+1} = \lambda_1$. The parameterization $\lambda_i = \alpha \bar{G}(i-1, \theta)$ assumes λ_i decreases to zero. However, λ_i must decrease to model reliability growth and ξ_i is not a meaningful quantity otherwise.

Let $M_i = M_i(S_i, t)$ denote the number of times the error detected at time S_i again occurs during the interval (S_i, t). The joint density function of $(R, S_1, S_2, \ldots, S_R, M_1, M_2, \ldots, M_R)$ is easily obtained from the fact that $M_1, M_2, \ldots, M_R$ are conditionally, given $(R, S_1, S_2, \ldots, S_R)$, independent Poisson random variables with means $\xi_i(t-S_i)$. Since this joint density function is identical to (3.3), the model described by (i)-(iii) is equivalent to the model in Section 3.2. The intervals (S_i, t) have random length and thus the unconditional distribution of M_i is not Poisson; this seems to be the precise way that (i)-(iii) differ from the model studied by Sandland and Cormack (1984).

3.5 The Distributions of R and S_k

If the λ_i are all distinct, then $H_k(x) = P(S_k \leq x)$ can be written (Cox, 1962, p.17)

$$H_0(x) = 1,\ H_1(x) = 1-\exp(-\lambda_1 x),\ x \geq 0$$

$$H_k(x) = \sum_{i=1}^{k} \pi_{ik} [1-\exp(-\lambda_i x)],\ x \geq 0,\ \pi_{ik} = \prod_{\substack{j=1 \\ j\neq i}}^{k} \lambda_j/(\lambda_j-\lambda_i)$$

where the weights π_{ik} (possibly negative) have a sum equal to one. Since $P(R \geq k) = H_k(t)$, the distribution of R is given by $P(R=k) = H_k(t) - H_{k+1}(t)$, $k=0,1,2,\ldots$ By noting that $R \geq k$ and $S_k \leq t$ are identical events, we also have

$$P(S_k \leq x \mid R \geq k) = H_k(x) / H_k(t),\ 0< x <t \qquad (3.9)$$

$$= 1,\ t \leq x.$$

3.6 A Comparison of the Error Recapture and Repetitive-Run procedures.

Nagel, Scholz, and Skrivan (1982, 1984) proposed a repetitive-run procedure where, after restoring a program

to its original form, the debugging sequence is repeated, perhaps several times, using independently generated series of inputs. A run is initiated by randomly selecting an input according to a usage distribution, defined as part of the original problem specification. Each run consists of testing the program on a random series of inputs and correcting errors whenever they are detected. The order of detecting errors as well as the gaps between the error detection times may vary from one replication to another. The repetitive-run procedure has been used by Nagel, Scholz, and Skrivan (1982, 1984) and also by Dunham and Pierce (1985) to study the effect of debugging on the reliability of several programs. In this Section we compare the asymptotic variances of parameter estimates obtained under the error recapture and repetitive-run procedures.

Let $V_1, V_2, \ldots, V_m$ denote independent random vectors having the same distribution as $V=(R, S_1, S_2, \ldots, S_R)$. As in Section 3.4, the gaps $Y_i = S_i - S_{i-1}$, between the first occurrence times are assumed to be independent and have exponential density functions $\lambda_i \exp(-\lambda_i y)$, $y \geq 0$ where, $\lambda_i = \alpha \bar{G}(i-1, \theta)$. If testing in each replicate extends over a time period of length s, the total test time is then ms. To compare the error recapture and repetitive-run procedures, we assume test

periods of equal length (i.e., ms=t) and that the first occurrence times have the same distribution under both models. That is, λ_i is assumed to be given by $\lambda_i = \alpha \exp[-\theta(i-1)]$, where α and θ are positive scalar parameters.

Since $V_1, V_2, \ldots, V_m$ are i.i.d random vectors, it suffices to consider the log likelihood function based on one observation of V, namely,

$$l_3(\alpha,\theta) = \sum_{i=1}^{\infty} \log \lambda_i \, I(R \geq i) - \sum_{i=1}^{\infty} (\lambda_i - \lambda_{i+1}) \, S_i \, I(R \geq i) - s \sum_{i=0}^{\infty} \lambda_{i+1} \, I(R=i)$$

$$= \log\alpha \sum_{i=1}^{\infty} i \, I(R=i) + \alpha s \sum_{i=1}^{\infty} [e^{-\theta i} - e^{-\theta(i-1)}] \, (S_i/s) \, I(R \geq i) - \theta \sum_{i=1}^{\infty} [i(i-1)/2] \, I(R=i) - \alpha s \sum_{i=0}^{\infty} e^{-\theta i} \, I(R=i) \tag{3.10}$$

The information on (α, θ) given by V is

$$b_{11} = -E(\partial^2/\partial\alpha^2 \, l_3) = E(R)/\alpha^2$$

$$b_{12} = -E(\partial^2/\partial\alpha\partial\theta \, l_3) = -s \, E(R e^{-\theta R}) + s \sum_{k=1}^{\infty} C_{1k} \, E[(S_k/s) \, I(R \geq k)]$$

$$b_{22} = -E(\partial^2/\partial\theta^2 \, l_3) = \alpha s \, E(R^2 e^{-\theta R}) - \alpha s \sum_{k=1}^{\infty} C_{2k} \, E[(S_k/s) \, I(R \geq k)]$$

$C_{1k} = k\,e^{-\theta k} - (k-1)\,e^{-\theta(k-1)}$, $C_{2k} = k^2 e^{-\theta k} - (k-1)^2 e^{-\theta(k-1)}$, $k=1,2,\ldots,$

$$E[(S_k/s)\ I(R \geq k)] = (1/s)\int_0^s x\, dH_k(x) \qquad (3.11)$$

This last expression follows from (3.9) and is also given by

$$E[(S_k/s)\,I(R \geq k)] = \sum_{i=1}^{k} \pi_{ik}\,(1/\lambda_i s)[1-\exp(-\lambda_i s)] - \sum_{i=1}^{k} \pi_{ik} \exp(-\lambda_i s)$$

Let $(\tilde{\alpha}, \tilde{\theta})$ denote the maximum likelihood estimator of (α, θ) given by m replicates in the repetitive-run procedure, and let $(\hat{\alpha}, \hat{\theta})$ denote similar estimators given by the error recapture procedure. The asymptotic variances are

$$\mathrm{Var}(\hat{\alpha}) = (\alpha t)^{-1}\,\alpha^2 \qquad \mathrm{Var}(\hat{\theta}) = (\alpha t)^{-1}\, e^{\theta}(1-e^{-\theta})^2$$

$$\mathrm{Var}(\tilde{\alpha}) = (mb)^{-1}\,\alpha^2 b_{22} \qquad \mathrm{Var}(\tilde{\theta}) = (mb)^{-1}\,E(R)$$

$$b = b_{22}\,E(R) - (\alpha b_{12})^2$$

where, in the expressions given for the variances of $\hat{\alpha}$ and $\hat{\theta}$, N has been replaced by $\alpha t = E(N)$.

By substituting $t = ms$, the asymptotic relative efficiencies are

Table 3.2: Asymptotic relative efficiencies of the repetitive-run ($\tilde{\alpha}$, $\tilde{\theta}$) and error recapture ($\hat{\alpha}$, $\hat{\theta}$) estimators; efficiencies are variance ratios with var($\hat{\alpha}$) and var($\hat{\theta}$) in the numerators.

α s	θ	E(R_s)	e($\tilde{\alpha}$, $\hat{\alpha}$)	e($\tilde{\theta}$, $\hat{\theta}$)
80	0.10	22.2	0.22	0.50
	0.20	14.4	0.16	0.42
	0.30	11.0	0.13	0.34
	0.40	9.0	0.11	0.28
	0.50	7.7	0.10	0.24
100	0.10	24.2	0.20	0.49
	0.20	15.4	0.15	0.38
	0.30	11.7	0.11	0.30
	0.40	9.5	0.09	0.24
	0.50	8.1	0.08	0.20
120	0.10	25.9	0.19	0.47
	0.20	16.3	0.13	0.35
	0.30	12.3	0.10	0.26
	0.40	10.0	0.08	0.21
	0.50	8.5	0.07	0.17
150	0.10	28.0	0.17	0.44
	0.20	17.4	0.11	0.30
	0.30	13.0	0.09	0.22
	0.40	10.5	0.07	0.17
	0.50	8.9	0.06	0.13
200	0.10	30.7	0.14	0.39
	0.20	18.8	0.09	0.24
	0.30	14.0	0.07	0.17
	0.40	11.2	0.06	0.12
	0.50	9.5	0.05	0.09

$$\mathrm{Var}(\hat{\alpha})/\mathrm{Var}(\tilde{\alpha}) = b\ (\alpha s\ b_{22})^{-1} \qquad (3.12)$$

$$\mathrm{Var}(\hat{\theta})/\mathrm{Var}(\tilde{\theta}) = b[\alpha s\ E(R)]^{-1} e^{\theta}(1-e^{-\theta})^{2}$$

where these quantities depend only on αs and θ.

The calculated efficiencies in Table 3.2 are based on (3.11), (3.12), and the distribution of R given in Section 3.5. The relative efficiency varies from about 1 to 13 percent and thus the time period of testing for the repetitive-run procedure may need to be more than eight times that of error recapture to obtain the same amount of information. The greatest gain in information occurs when E(R) is small and this corresponds to programs for which the error detection rate is small. The low efficiencies in Table 3.2 are due to the fact that the repetitive-run procedure permits each error to be observed at most one time during each replicate.

3.7 Limiting Distributions

Let $\hat{\theta}_n$ denote the maximum likelihood estimator of θ based on observing $(R_1, Z_1), (R_2, Z_2), \ldots, (R_n, Z_n)$. In this section we

show that the limiting ($n \to \infty$) distribution of $n^{1/2}(\hat{\theta}_n - \theta)$ is identical to the limiting distribution that would be obtained if $\hat{\theta}_n$ were computed from i.i.d random variables X_1, $X_2, \ldots, X_n$ that have density function $g(i,\theta)$. The asymptotic covariance matrix is then I_θ^{-1}, $I_\theta = (I_{ij})$, where $I_{ij} = E_\theta\{-\partial^2 / \partial\theta_i \partial\theta_j [\log g(X,\theta)]\}$ and X has density function $g(i,\theta)$.

In terms of $X_1, X_2, \ldots, X_n$, the log likelihood function based on the second factor of (3.1) is

$$l_2(\theta) = \sum_{i=1}^{n} \log g(X_i,\theta)\ I(X_i \le R_{i-1}) + \sum_{i=1}^{n} \log \bar{G}(R_{i-1},\theta)\ I(X_i > R_{i-1})$$

where $R_0=0$, $R_1=1$, and $R_k = R_{k-1} + I(X_k > R_{k-1})$, $k=2,3,\ldots$

Let $l(\theta) = \sum_{i=1}^{n} \log g(X_i,\theta)$ and note that

$$l_2(\theta) - l(\theta) = \sum_{i=1}^{n} I(X_i > R_{i-1}) [\log g(X_i,\theta) - \log \bar{G}(R_{i-1},\theta)]$$

To simplify this last expression, let $1 = n_1 < n_2 < \ldots < n_r \le n$ denote the epochs at which $X_i > R_{i-1}$. Conditionally, given $N_1 = n_1$, $N_2 = n_2, \ldots, N_r = n_r$, and $R_n = r$, the set $X_{n(1)}, X_{n(2)}, \ldots, X_{n(r)}$, consists of independent random variables with density functions $h_k(i,\theta) = g(i,\theta)/\bar{G}(k-1,\theta)$, $i=k,k+1,\ldots$

Since the conditional distribution of $X_{N(1)}$, $X_{N(2)}$, ..., $X_{N(r)}$, depends on N_1, N_2, ..., N_R and R_n only through R_n, it is simpler to let Y_1, Y_2, ..., Y_r be independent random variables with density functions $h_k(i,\theta)$, k=1,2,...,r. Then

$$l_2(\theta) - l(\theta) = \sum_{k=1}^{R_n} \log\, h_k(Y_k,\theta) \qquad (3.13)$$

and

$$-\partial^2/\partial\theta_p\partial\theta_q \log\, h_k(i,\theta) = b_{p1}b_{q1}\,[\exp(\theta_1 b_{1i} + \theta_2 b_{2i}) - 1]^{-2}$$

$$\exp(\theta_1 b_{1i} + \theta_2 b_{2i}) \quad (p,q=1,2,\ i=k,k+1,\ldots)$$

By using the relations $e^x(e^x-1)^{-2} = (e^x+e^{-x}-2)^{-1}$ and $e^x + e^{-x} - 2 \geq x^2$, we obtain

$$|-\partial^2/\partial\theta_p\partial\theta_q\,[\log\, h_k(i,\theta)]| \leq (\theta_p\theta_q)^{-1} \quad (p,q=1,2) \qquad (3.14)$$

Our remaining discussion requires the notation

$$U_i = \partial/\partial\theta_i\, l_2(\theta),\ V_i = \partial/\partial\theta_i\, l(\theta),$$

$$W_{ik} = \partial/\partial\theta_i \log\, h_k(Y_k,\theta),\ (k=1,2,\ldots,r)$$

Since $E(W_{ik}) = 0$ and $E(W_{ik}^2) = - E[\partial^2/\partial\theta_i \partial\theta_i \log h_k(Y_k,\theta)]$

we have from (3.13) and (3.14) that

$$\text{Var } (U_i - V_i) = E\left(\sum_{k=1}^{R_n} W_{ik}^2\right)$$

$$\leq \theta_i^{-2} E(R_n), \quad (i=1,2)$$

Thus $n^{-1/2}(V_i - U_i)$ converges $(n \to \infty)$ in probability to zero providing limit $n^{-1} E(R_n) = 0$.

The representation of R_n given earlier in Section 3.3 implies

$$R_n = 1 + \sum_{i=2}^{n} I(X_i > R_{i-1})]$$

$$E(R_n) = 1 + \sum_{i=1}^{n-1} E\,[G(R_i,\theta)] \qquad (3.15)$$

where, since $n^{-1} E(R_n)$ is an average of the terms on the right of (3.15), it suffices to show that $\lim E\,[\bar{G}(R_n,\theta)] = 0$. This limit is easily obtained from the fact that R_n tends to infinity in probability, and thus further details are omitted.

Since $V = (V_1, V_2)$ is a linear function of i.i.d random variables, the preceeding discussion shows that $n^{-1/2} V$ and $n^{-1/2} U$ are asymptotically equivalent and that $n^{-1/2} U$ has a limiting bivariate normal distribution with mean vector zero

and covariance matrix I_θ. The usual consistency and Taylor series arguments (e.g., Cox and Hinkley, 1974) imply that $n^{1/2}(\hat{\theta}_n - \theta)$ has a limiting bivariate normal distribution with covariance matrix I_θ^{-1}. Since $\{T_n\}$ and $\{R_i, Z_i)\}$ are independent, $n^{1/2}(\hat{\theta}_n - \theta)$ and $n^{1/2}(n/T_n - \alpha)$ are also independent. By Theorem 8.1 of Serfozo (1975), (see also Karr, 1986, p. 406) $N^{1/2}(\hat{\alpha} - \alpha)$ and $N^{1/2}(\hat{\theta} - \theta)$ have asymptotically ($t \to \infty$) independent normal distributions with variance α^2 and covariance matrix I_θ^{-1}, respectively.

Chapter 4

Application to Repetitive-Run Data

4.1 Introduction

Nagel, Scholz and Skrivan (1984) observed that many models have been introduced in the literature, but without a clear statement about the mathematical and statistical foundations that motivated the model. They pointed out that for a consistent theoretical foundation, a deeper understanding of the process is needed. As an attempt in this direction, they conducted a series of experiments consisting of simulations conducted on code prepared according to a set of requirements and executed with randomly selected inputs. The code is initialized to an original state then tested on randomly generated inputs. Errors are corrected as they are encountered until a stopping rule is satisfied. Replication is introduced by repeating the entire process from

initialization. The following section gives further details concerning the testing process.

4.2 Description

The simulations are initiated by generating random inputs according to a distribution, called the usage distribution, defined as part of the original problem specification. After correcting an error in the program, other inputs are generated independently and the process is repeated. If for some execution an error is indicated, the error is recorded together with the number of executions since the last error, and the error is corrected.

The simulation begins with the program in its initial state. This state is reached when the program successfully compiles and correctly executes a number of predetermined test cases. These tests are defined as static tests for a given specification and the program must pass these static tests as well as successfully execute the input causing failure before simulation can be reinitiated. Once reinitiated, the process is repeated error by error until a stopping rule is satisfied. Termination of the experiment occurred when an error is detected that is too costly to fix

or an upper bound on the number of inputs is reached, whichever comes first.

Traditional tests on software force the experimenter to predict the reliability of the program from a single manifestation of the error process. To observe different realizations of the order of detecting errors the program is restored to its original state and testing is repeated, using another randomly selected input sequence. The experimental flow for each run is exactly the same except for the consequences of using different inputs. Each run may generate different random errors in different orders and with varying spacings between the error detection times.

4.3 Software Error Categories

The following list describes the category of different possible software errors in a program:

Computational Errors

- Incorrect operand in logical sequence
- Incorrect use of parenthesis
- Sign convention error
- Unit or data conversion error

Computation produces an over/under flow

Incorrect/inaccurate equation used

Precision loss due to mixed mode

Missing computation

Rounding or truncation error

Logical Errors

Incorrect Operand in logical expression

Logic activities out of sequence

Wrong variable being checked

Missing logic or condition tests

Too many/few statements in loop

Loop iterated incorrect number of times

Duplicate logic

Data Input Error

Invalid input read from correct data file

Input read from incorrect data file

Incorrect input format

Incorrect format statement referenced

End of file encountered prematurely

End of file missing

Data Handling Errors

- Data file not rewound before reading
- Data initialization not done
- Data initialization done improperly
- Variable referred to by the wrong name
- Bit manipulation done incorrectly
- incorrect variable type
- Data packing/unpacking error
- Sorting error
- Subscripting error

Data Output Error

- Data written on wrong file
- Data written according to the wrong format statement
- Data written in wrong format
- Data written with wrong carriage control
- Incomplete or missing output
- Output field size too small
- Line count or page eject problem
- Output garbled misleading

Interface Errors

- wrong subroutine called
- Call to subroutine not made or made in wrong place
- Subroutine arguments not consistent in type, units
- Subroutine called is nonexistent
- Software/data base interface error
- Software/Software interface error

Data Definition Errors

- Data not properly defined/dimensioned
- Data referenced out of bounds
- Data being referenced at incorrect location
- Data pointers not incremented properly

Data Base Errors

- Data not initialized in data base
- Data initialized to incorrect value
- Data units are incorrect

Operation Errors

- Operating system error
- Hardware error
- Operator error

Test execution error

User misunderstanding/error

Configuration control error

Documentation Errors

User manual

Interface specification

Design specification

Requirements specification

Test documentation

4.4 The Likelihood Function

The repetitive-run procedure yields several realizations of $(R, S_1, S_2, \ldots, S_R)$, where R is the number of failures observed during an interval $(0,s)$ and $S_1, S_2, \ldots, S_R$ are the ordered failure times. As before, the gaps $Y_i = S_i - S_{i-1}$ $(S_0 \equiv 0)$, $i=1,2,\ldots$ are assumed to be independent random variables with exponential density functions

$$f(y) = \lambda_i \exp(-\lambda_i y), \quad y \geq 0 \tag{4.1}$$

where

$$\lambda_i = \alpha \bar{G}(i-1, \theta) \tag{4.2}$$

Although this parameterization is identical to the one studied in Chapter 3, the two models differ in fundamental ways; e.g., except for the fact that R has a discrete distribution, the observations here have continuous distributions whereas the sufficient statistic for the error recapture model has a discrete distribution.

Let s_i denote the length of the period of testing in the ith replicate and let $V_i = (R_i, Y_{i1}, Y_{i2}, \ldots, Y_{iR(i)+1})$, i=1,2,..., m denote the observations obtained in m replicates, where $Y_{iR(i)+1}$ is the spacing between s_i and $S_{iR(i)}$ with S_{i1}, S_{i2}, $\ldots, S_{iR(i)}$ being the times at which errors are detected in the ith replicate. Under the assumption that $V_1, V_2, \ldots, V_m$ are independent random vectors and that $Y_1, Y_2, \ldots,$ have independent exponential distributions, the full likelihood function is

$$L(\alpha,\theta) = \prod_{i=1}^{m} \prod_{j=1}^{R_i} \lambda_j \prod_{j=1}^{R_i+1} \exp(-\lambda_j y_{ij}) \qquad (4.3)$$

where

$$\lambda_j = \alpha \exp(-\theta_1 a_{1,j-1} - \theta_2 a_{2,j-1}), \quad j=1,2,\ldots, \qquad (4.4)$$

and

$$a_{1,j} = j, \quad a_{2,j} = j^2$$

Let $D_k = \{i: R_i \geq k\}$, $k=1,2,\ldots$ and note that y_{ik} is an observed quantity whenever $i \in D_k$. By substituting from (4.4), the log likelihood can be written as

$$\ln L(\alpha,\theta) = A_1 \ln\alpha - \theta_1 B_1 - \theta_2 B_2 - \alpha C_1 - \alpha \bar{G}(1,\theta) C_2 - \alpha \bar{G}(2,\theta) C_3 - \ldots \tag{4.5}$$

where

$$A_1 = \sum_{i=1}^{m} R_i$$

$$B_1 = \sum_{i=1}^{m} \sum_{j=1}^{R_i} a_{1,j-1}$$

$$B_2 = \sum_{i=1}^{m} \sum_{j=1}^{R_i} a_{2,j-1}$$

$$C_k = \sum_{D_k} y_{ik} \quad (k=1,2,\ldots)$$

$$\bar{G}(k,\theta) = \exp(-\theta_1 a_{1k} - \theta_2 a_{2k})$$

$$\bar{G}(0,\theta) = 1.$$

$$k=1,2,\ldots, \tag{4.6}$$

An estimate $(\hat{\alpha},\hat{\theta})$ that maximizes (4.5) is the solution to the system of equations

$$A_1/\alpha - [C_1 + \bar{G}(1,\theta)C_2 + \bar{G}(2,\theta)C_3 + \ldots + \ldots] = 0 \qquad (4.7)$$

$$-B_1 + \frac{A_1[a_{11}\bar{G}(1,\theta)C_2 + a_{12}\bar{G}(2,\theta)C_3 + \ldots + \ldots]}{[C_1 + \bar{G}(1,\theta)C_2 + \bar{G}(2,\theta)C_3 + \ldots + \ldots]} = 0 \qquad (4.8)$$

$$-B_2 + \frac{A_1[a_{21}\bar{G}(1,\theta)C_2 + a_{22}\bar{G}(2,\theta)C_3 + \ldots + \ldots]}{[C_1 + \bar{G}(1,\theta)C_2 + \bar{G}(2,\theta)C_3 + \ldots + \ldots]} = 0 \qquad (4.9)$$

Since (4.8) and (4.9) depend only on θ, $\hat{\theta}$ can be obtained by solving the last two equations and then substituting $\hat{\theta}$ into (4.7) to get $\hat{\alpha}$.

Under the stopping rule described earlier in this chapter, the period of observation in each replicate ends with a failure. A common data modification in such cases (e.g., Pedersen, 1979) is to treat the period of observation as being a fixed interval $(0, s_1)$, where s_1 is actually the occurrence time of the last observed event.

4.5 A Test of Fit

Each replicate may produce a different final program, so the main parameter of interest is less clear than for the

error recapture procedure. However, the constants $\{a_{1i}\}$ and $\{a_{2i}\}$ must be selected prior to fitting a model.

Consider the case $a_{1i}= i$, $a_{2i}= i^2$ and the following two submodels of (4.4)

$$\lambda_{i+1} = \alpha \exp(-\theta_1 i), \quad \alpha > 0, \quad \theta_1 > 0 \tag{4.10}$$

$$\lambda_{i+1} = \alpha \exp(-\theta_2 i^2), \quad \alpha > 0, \quad \theta_2 > 0 \tag{4.11}$$

and we determine whether an adequate fit can be achieved with the latter.

Let $L(\alpha,\theta_1,0)$ denote the likelihood function corresponding to (4.10) and similarly let $L(\alpha,0,\theta_2)$ be the likelihood function corresponding to (4.11). Then

$$\ln L(\alpha,\theta_1,0) = A_1 \ln\alpha - \theta_1 B_1 - \alpha C_1 - \alpha \bar{G}(1,\theta_1,0) C_2 - \alpha \bar{G}(2,\theta_1,0) C_3 - \ldots \tag{4.12}$$

$$\ln L(\alpha,0,\theta_2) = A_1 \ln\alpha - \theta_2 B_2 - \alpha C_1 - \alpha \bar{G}(1,0,\theta_2) C_2 - \alpha \bar{G}(2,0,\theta_2) C_3 - \ldots \tag{4.13}$$

The estimating equations corresponding to (4.10) and (4.11) are obtained by setting $\theta_2=0$ in (4.8) and $\theta_1=0$ in (4.9).

Consider the following subsets of the total set of parameter values,

$$\Omega' = \{(\alpha,\theta_1,\theta_2): \alpha>0,\ \theta_1>0,\ \theta_2>0\}$$

$$\Omega_{10} = \{(\alpha,\theta_1,\theta_2): \alpha>0,\ \theta_1>0,\ \theta_2=0\}$$

$$\Omega_{01} = \{(\alpha,\theta_1,\theta_2): \alpha>0,\ \theta_1=0,\ \theta_2>0\}$$

where Ω_{10} and Ω_{01} denote the parameter sets corresponding to models (4.10) and (4.11), respectively. In this context, a test of fit of the model defined by (4.10) is a test of the composite null hypothesis $\theta_2=0$ versus the one sided alternative $\theta_2>0$.

Let W_1 be the likelihood ratio statistic for the simpler model (4.10). For a large number of replicates and under suitable regularity conditions (Cox and Hinkley, 1974), the null distribution of the likelihood ratio statistic W_1 is approximately a chi-square distribution with 1 degree of freedom where W_1 is given by

$$e^{-1/2\,W_1} = \frac{\max\limits_{(\alpha,\theta_1,\theta_2)\in\Omega_{10}} L(\alpha,\theta_1,\theta_2)}{\max\limits_{(\alpha,\theta_1,\theta_2)\in\Omega} L(\alpha,\theta_1,\theta_2)} \qquad (4.14)$$

and $\Omega = \Omega_{10} + \Omega'$ is parameter space for the full model. Similarly, with $\Omega = \Omega_{01} + \Omega'$ the likelihood ratio statistic W_2 corresponding to (4.11) can be defined in a form like (4.14).

4.6 Numerical Example

The data in Appendix B was collected by Nagel, Scholz, and Skrivan (1984) and are times between failures observed in 50 replicates when using the repetitive-run procedure. A program run consists of testing the program on a randomly selected input series and correcting errors whenever they are detected. The runs are replicated by restoring the program to its original form and testing it again on another randomly selected input series. Each run terminates when an error is detected that is too costly to fix or when an upper bound on the length of an input series is reached.

Part of the analysis of the data in Appendix B given by Nagel, Scholz, and Skrivan (1984) assumes that the waiting times (i.e., numbers of executions) between failures have independent geometric distributions with parameters $\{p_i\}$. They define p_i as as the conditional probability that a random execution of the program will result in an error given that i-1 errors have been corrected. By plotting estimates of log

p_i, i=1,2,..., they conclude that log p_i is linear in i, i=1,2,...

As noted by Miller (1988), the geometric distributions can be approximated by exponential distributions whenever the quantities p_i are small. A continuous approximation to the geometric model assumes that the waiting times have independent exponential distributions with rate parameters $\lambda_i = p_i$. Then log p_i is linear in i, i=1,2,..., only if λ_i takes the form $\lambda_i = \alpha \exp[-\theta(i-1)]$, $\alpha > 0$, where θ is positive whenever reliability improves as faults are removed from a program. The log linear rate model was proposed by Moranda (1975) and also appears in Cox and Lewis (1966).

Using the data in Appendix B and fitting the more general form of the model described in Sections 4.4 and 4.5, we obtain the estimates $(\hat{\alpha}, \hat{\theta}_1, \hat{\theta}_2)$ = (0.37, 1.07, 0.0). The maximum of the log likelihood (4.5) is -2232.92. The estimates $(\hat{\alpha}, \hat{\theta}_1) = (0.37, 1.07)$ and $(\hat{\alpha}, \hat{\theta}_2) = (0.064, 0.132)$ maximize the likelihood functions corresponding to the simpler models (4.10) and (4.11), respectively. The respective maximum values of the log likelihood are -2232.92 and -2356.41. Further the necessary calculations for W_1 and W_2 yield their respective values as W_1=0.0 and W_2=246.98. The upper tail value of the chi-square distribution with one degree of freedom at the 5 percent level of significance is 3.84. Since

W_2 exceeds this value, the full model (4.4) is preferred over the corresponding simpler model (4.11), and the small value of W_1 indicates no difference in the fit of models (4.4) and (4.10). Further, since (4.11) is rejected and (4.10) is accepted by the likelihood ratio test, (4.10) (i.e., the log linear rate model) is the best fitting model.

The Newton-Raphson method was used to solve the nonlinear estimating equations (4.8) and (4.9). We began with an initial approximation $(\theta_{10},\theta_{20})$ in the positive quadrant close to the origin and repeatedly improved it. At the ith stage when there is no further improvement, $(\theta_{1i},\theta_{2i})$ is the approximation to (θ_1,θ_2).

To check whether the estimates found above actually maximize the likelihood, the values of likelihood function were calculated inside a fairly large grid of (θ_1,θ_2) values. These values, not shown here, indicate that the calculated values of the estimates yield the maximum of the likelihood function.

Our purpose in this section is not that of proving that any particular model will give a better fit than another model. Since the rate at which errors are detected and eliminated is likely to vary from one program to another, the goodness of fit of a model must generally be examined to avoid overly optimistic prediction of reliability.

In comparison to the log linear rate model (4.10), the above results indicate that no improvement is realized by including the extra term $a_{21} = i^2$. This conclusion was also reached by Nagel, Scholz, and Skrivan (1984), not only for the data in Appendix B, but also for other programs they tested. Miller (1988) has pointed out, however, that the log linear rate model is unlikely to be a universal model. The example he gives is a system consisting of two software modules which, if tested separately, the failure times are assumed to follow a log linear rate model. If the two modules are then combined and tested as one system, he argues that the failure rates cannot then exhibit a log-linear pattern.

Appendix A

Lemma A: Show that $l_2(\theta)$ in equation (3.5) is concave in $\theta_i>0$, i=1,2.

Proof: Let $U=(u_{pq})$, p,q=1,2 where

$$U_{pq} = -\partial^2/\partial\theta_p\partial\theta_q\, l_2(\theta)$$

$$= \sum_{i=1}^{r} b_{p1}\, b_{q1} C_i$$

$$C_i = m_i \exp(\theta_1 b_{1i} + \theta_2 b_{2i})\,[\exp(\theta_1 b_{1i} + \theta_2 b_{2i}) - 1]^{-2}$$

We have $u_{11} \geq 0$ and

$$(u_{11}u_{22} - u_{12}^2)\,(\sum_{i=1}^{r} C_i)^{-2} = (\sum_{i=1}^{r} b_{1i}^2 f_i)(\sum_{i=1}^{r} b_{2i}^2 f_i) - (\sum_{i=1}^{r} b_{1i}\, b_{2i}\, f_i)^2$$

where $f_i = C_i / \sum_{j=1}^{r} C_j$. Since $b_{1i}>0$ and $b_{2i}>0$, Holder's inequality implies $u_{11}\,u_{22} - u_{12}^2 \geq 0$. Thus U is positive semidefinite and the latter implies that $l_2(\theta)$ is concave in $\theta_i>0$, i=1,2, (e.g., Beltrami, 1970, p. 74).

Appendix B

Table B.1: Times between the occurrence of program errors in m=50 replicates. (Nagel, Scholz, Skrivan, 1984)

1 3 26 72 51 1039 16 1898
1 3 45 10 257 112 6 1032
2 23 115 500 111 1969 9887
1 10 5 16 69 1059 909 256
3 2 45 96 461 129 404
1 1 29 16 100 2052 1081 1025
1 23 8 17 83 4140 8842 2120
1 1 55 75 17 33 929 597
3 62 116 147 472 2551 4353
1 4 1 88 20 40 372 940
1 1 24 3 129 255 3514 12613
1 58 36 105 777 83 2380
1 4 13 37 43 6 435 789
2 5 5 17 175 323 3865
1 1 1 44 698 210 462
2 36 112 57 348 857 248 440
1 3 82 220 227 898 4906 310
1 4 25 179 46 634 2261 2947
1 2 24 23 237 132 866 27273
3 4 13 17 65 1165 70 1236
2 50 62 64 89 2504
1 1 1 26 109 137 5662 2359
1 3 20 9 2285 171 5491

```
2 3 26 164 434 1040 2890
1 3 24 21 3 375 108 445 330
3 2 14 210 390 412 308 2162
1 2 32 24 133 213 292 1532 1771
1 1 24 93 118 833 462 758
1 3 5 73 130 393 361 8285 622
1 3 52 26 115 707 22 1855 1252
1 1 15 60 114 135 110 38
1 4 2 55 77 1007 89 391
1 5 89 173 351 80 765 4801
1 3 17 2 543 4171 4262
1 6 1 44 28 771 862 2292
1 10 104 39 113 1058 622
1 2 41 12 258 120 3207
1 1 56 98 91 949 455 9238
1 2 1 8 120 2232 42 137 444
3 1 57 29 272 254 1281 5914
1 2 1 7 34 35 1717 714
1 2 55 25 87 172 143 1382
3 1 11 25 88 344 614 675 294
2 4 23 98 81 89 1057 354 43
1 6 17 215 186 1061 1789
3 1 26 18 123 13 1088 710
2 3 5 8 3 743 199 2477
1 16 8 84 648 86 62
1 1 25 22 70 335 1037 4270
1 2 13 63 7 77 306 924
```

Bibliography

[1] Avizienis, A.(1975). Fault tolerance and fault intolerance: Complementary approaches to reliable computing. *Proc. Int. Conf. Reliable Software*, pp. 458-464.

[2] Beltrami, E. J.(1970). *An Algorithmic Approach to Nonlinear Analysis and Optimization*. New York: Academic Press.

[3] Chapman, D. G.(1951). Some properties of the hypergeometric distribution, *Univ. Calif. Publications in Statistics*, 1, 7, pp. 131-160.

[4] Chenoweth, H. B.(1981). Modified Musa theoretic software reliability, *Proc. Annual Reliability and Maintainability Symposium*, pp. 353-356.

[5] Cox, D. R.(1962). *Renewal Theory*. London: Methuen.

[6] Cox, D. R. and Hinkley, D.V.(1974). *Theoretical Statistics*. London: Chapman and Hall.

[7] Cox, D. R. and Lewis, P. A. W.(1966). *The Statistical Analysis of Series of Events*. London: Methuen.

[8] Feller, W.(1957). *An Introduction to Probability Theory and its Applications, Vol. 1, 2nd Edn.* New York: Wiley.

[9] Foreman, E. H. and Singpurwalla, N.D.(1977). An empirical stopping rule for debugging and testing computing software. *JASA*, vol. 72, pp. 750-757.

[10] ———— (1979). Optimal time intervals for testing hypotheses on computer software errors. *IEEE Trans. Rel.*, vol. R-28,3, pp. 250-253.

[11] Goel, A. L.(1980). A summary of the discussion on an analysis of competing software reliability models. *IEEE Trans. on Software Engg.*, vol. SE-6,5 pp. 501-502.

[12] Goel, A. L. and Okumoto, K.(1979). Time-dependent error detection rate model for software reliability and other performance measures. *IEEE Trans. on Rel.*, vol. R-28, pp. 206-211.

[13] Dunham, J. R., and Pierce, J. L.(1985). An Experiment in Software Reliability, NASA CR-172553. NASA Langley Research Center, Hampton, VA.

[14] Goodman, L.(1953). Sequential sampling tagging for population size problems. *Ann. Math. Statist*, 24, 56-69.

[15] Goudie, I. B. J.(1990). A likelihood-based stopping rule for recapture debugging. *Biometrika*, 77, pp. 203-206.

[16] Huggins, R. M.(1989). On the statistical analysis of capture experiments. *Biometrika*, 76, 133-140.

[17] Jelinski, Z. and Moranda, P. B.(1972). Software reliability research. *Statistical Computer Performance Evaluation*, W. Frieberger, Ed., New York: Academic Press, pp. 465-484.

[18] Joe, H. and Reid, N.(1985a). Estimating the number of faults in a system. *JASA*, vol. 80, pp. 222-226.

[19] ——— (1985b). On the software reliability models of Jelinski-Moranda and Littlewood. *IEEE Trans. on Rel.*, vol. R-34, pp. 216-218.

[20] Karr, A. F.(1986). *Point Processes and their Statistical Inference*. New York: Marcel Dekker, Inc.

[21] Littlewood, B.(1980). The Littlewood-Verrall model for software reliability compared with some rivals. *J. Systems and Software*, 1, pp. 251-258.

[22] ————(1981). Stochastic reliability growth: A model for fault removal in computer programs and hardware designs. *IEEE Trans. on Rel.*, vol. R-30, pp. 313-320.

[23] Littlewood, B. and Verrall, J. L.(1973). A Bayesian reliability growth model for computer software. *IEEE Symposium on Computer Software Reliability*. New York, April 30 - May 2, 1973, pp. 70-76.

[24] ———(1981). On the likelihood function of a debugging model for computer software reliability. *IEEE Trans. on Rel.*, vol. R-30,2 pp. 145-148.

[22] Miller, D. R.(1986). Exponential order statistics models of software reliability growth. *IEEE Trans. on Software Engg.*, vol. SE-12,1 pp. 312-327.

[26] ————(1988). Making Statistical Inferences about Software Reliability, NASA CR-4197. NASA Langley Research Center, Hampton, VA.

[27] Mills, H. D.(1970). On the Statistical Validation of Computer Programs, IBM FSD unpublished paper.

[28] Moranda, P. B.(1975). Prediction of software reliability during debugging. *Proc. Annual Reliability and Maintainability Symposium*, pp. 327-332.

[29] ———— (1979). Event-altered rate models for general aeliability analysis, *IEEE Trans. Rel.*, vol. R-28,5, pp. 376-381.

[30] Musa, J. D.(1975). A theory of software reliability and its application. *IEEE Trans. on Software Engg.*, vol. SE-1, 3 pp. 312-327.

[31] Nagel, P. M., Scholz F. W., and Skrivan, J. A.(1982). Software Reliability: Repetitive-Run Experimentation and Modeling, NASA CR-165036. NASA Langley Research Center, Hampton, VA.

[32] ————(1984). Software Reliability: Additional Investigations into Modeling with Replicated Experiments, NASA CR-172378. NASA Langley Research Center, Hampton, VA.

[33] Nayak, T. K.(1988). Estimating population size by recapture sampling. *Biometrika*, 75, pp. 113-120.

[34] ——(1991). Estimating the number of component processes of a superimposed process. *Biometrika*, 78, pp. 75-81.

[35] Parzen, E.(1957). *Modern Probability Theory and its Applications, Vol. 1, 2nd ed.*, New York: Wiley.

[36] Pedersen, B. V.(1979). Approximating conditional distributions by the mixed Edgeworth-saddle point expansion. *Biometrika*, 68,3, pp. 597-604.

[37] Ramamoorthy, C. V. and Bastani, F. B.(1980). Modeling of the software reliability growth process. *COMPAC*, pp. 161-169.

[38] Randell, B.(1975). System structure for software fault tolerance. *IEEE Trans. on Software Engg.*, vol. SE-1, 6 pp. 220-232.

[39 Ross, S. M.(1985). Software reliability: The stopping rule problem. *IEEE Trans. on Software Engg.*, vol. SE-11, 12 pp. 1472-1476.

[40] Rao, C. R.(1973). *Linear Statistical Inference and its Applications, 2nd ed.*, Wiley Eastern Limited: New Delhi.

[41] Rudner, B.(1977). Seeding/Tagging Estimation of Software Errors: Models and Estimates. USAF Report RADC-TR-77-15.

[42] Sandland, R. L. and Cormack, R. M.(1984). Statistical inference for Poisson and multinomial models for capture-

recapture experiments. *Biometrika*, 71, 27-33.

[43] Schick, G. J., and Wolverton, R. W.(1978). An analysis of competing software reliability models. *IEEE Trans. on Software Engg.*, vol. SE-4, No. 2, pp. 104-120.

[44] Scholz, F. W.(1986). Software reliability modeling and analysis. *IEEE Trans. on Software Engg.*, vol. SE-12,1 pp. 25-31.

[45] Serfozo, R. F.(1975). Functional limit theorems, for stochastic processes based on embedded processes, *Adv. Applied Probability* 7, 123-139.

[46] Shooman, M. L.(1973). Operational testing and software reliability estimation during program development. *IEEE Symp. Comput. Software Reliability*, pp. 51-57.

[47] Sukert, A. N.(1977). An investigation of software reliability models, *Proc. Annual Reliability and Maintainability Symposium*, pp. 353-359.

[48] Vouk, M. A.(1990). Back-to-back testing. *Information and Software Technology*, 32, 34-45.

Biography

Rajan Gupta was born on August 22, 1956 in Bijnor town of Uttar Pradesh province of India. He graduated from University of Delhi in May 1975 with a B.Sc. (Hons.) in Mathematics, and M. Sc. Mathematical Statistics in May 1977. Soon after he began his career as Junior Research Scientist at Delhi Telephones. In 1982, he emigrated to Canada, where he carried out graduate studies at the University of Manitoba, and received a Master's degree in Applied Statistics in May 1984. During 1990-91, he was Asst. Professor of Statistics, at Embry-Riddle Aeronautical University, Daytona Beach. His former position include Biostatistician with McNeil Pharmaceuticals. He was awarded a Ph.D. in Statistics from Old Dominion University in December, 1991.

He has coauthored two papers on software reliability. This research was sponsored by the Fault-Tolerant Systems Branch at NASA Langley Research Center, grant number NAG-1-835.

His future interests include statistical methodology in biopharmaceutical research such as oncology, cardiovascular and AIDS research.