AFOSR-TR-97

0356

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Nov. 15, 1996	Final Technical

4. TITLE AND SUBTITLE
Stochastic Software Reliability: Modeling of Software Failures

5. FUNDING NUMBERS
F49620-96-1-0307

94-1 0130

6. AUTHOR(S)
Gordon M. Kaufman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Sloan School of Management MIT
50 Memorial Drive
Cambridge, MA 02142-1347

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 Donean Ave., Ste B115
Bolling AFB DC 20332-8080

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Given a test history consisting of a record of times of software failures, together with a record of both the time at which each failure occurred and the *type* of that failure: 1) How many faults of each type remain in the software? 2) How much added time on test is required to uncover a pre-specified number of faults? 3) If testing continues for a given increment of time, how many faults of each type will be uncovered?

Stochastic models of software failures composed of a super-population process that generates elements of a finite population of elements that are then successively sampled is used to construct both Bayesian and non-Bayesian methods of parameter and predictive inference. Predictive validation of these models is done using NASA/GODDARD Software Engineering Lab data.

14. SUBJECT TERMS
Software Reliability, Stochastic Modeling, Bayesian Inference, NASA/Goddard

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

To: "Jon Sjogren"<jas@src.umd.edu>
Date: November 15, 1996
From: Gordon M. Kaufman
Sloan School of Management
Masachusetts Institute of Technology
50 Memorial Drive
Cambridge, MA 02142-1347
Subject: Final Technical Report

Probabilistic Modeling of Software Reliability

Grant # F49620-94-1-0130 research on software reliability was designed to answer three software management questions: given a test history consisting of a record of times of software failures together with *the type of failure that occurred at each failure time,*

- How many faults of each type remain in the software?
- How much additional time on test is required to uncover a pre-specified number of faults?
- If testing continues for say, T more units of time, how many faults of each type will be observed?

In contrast to most probabilistic models of software reliability that appear in the literature, the models employed generate answers to these questions *explicitly account for distinctions among types of faults and allows computation of predictive distributions of the number of each type of fault remaining subsequent to a test history*. This is a particularly important generalization that allowed us to model NASA/GODDARD Software Engineering Laboratory data describing the results of testing of a large software system for which NASA/GODDARD software engineers classified faults into six distinct fault types and recorded the number of each type of fault found during discrete intervals of time on test during each of several test phases.

Both Bayesian and non-Bayesian approaches to inference and prediction were adopted in this research. Fault occurrence was modeled as follows: Numbers $N_1,....,N_K$ of faults of type 1,...,K are generated by a super-population process. Then a finite population composed of N_i faults of type i=1,...,K is successively sampled. Successive sampling captures reliability growth. Neither parameters of the super-population process, nor those of the finite population, are known with certainty.

The non-Bayesian approach to inference and prediction employed unbiased estimation procedures. These procedures were shown to be asymptotically equivalent to a form of conditional maximum likelihood estimation. A Monte Carlo study of the behavior of unbiased and of conditional maximum likelihood estimators in the presence of small samples was conducted. Both performed reasonably well, even for very small samples. [1]

The Bayesian approach requires assigment of a prior distribution to parameters of both the super-population process generating the number of faults of each type residing in the software and to the parameters that generate times to discovery of faults of each type once the number of each type is fixed. Then both posterior-to-the-data distribution of these parameter sets, calculated via Bayes Theorem, as well as post-data predictive distributions are computed. In particular, predictive distributions for the number of faults of each type remaining, for the number of faults to be observed in an additional fixed time on test interval, and for the incremental time on test to discovery of a pre-specified number of faults are computed. Markov Monte Carlo Methods are employed to carry out computation of Bayesian post-data predictive distributions. They are *the* key to feasible time computation.

Numerical results, including predictive distributions based on NASA/GODDARD Software Engineering Laboratory acceptance test phase data were presented in an invited talk in September 1995 at the Third World Meeting of the International Society for Bayesian Analysis in Oaxaca, Mexico and in an invited talk at the Spring meeting of the Institute for Operations Research and Management Science in Washington DC in May 1996. These particular post-data predictive

distributions required evaluation of 11-dimensional integrals. Gibbs and griddy Gibbs Markov Chain Monte Carlo sampling schemes were employed to this end.

Once post-data predictive distributions can be calculated, it is possible to do predictive validation of the structure of the underlying data generating process model. This is done by splitting test data into an *early* sample and a *late* sample. The early sample is combined with a prior distribution on parameters and, in turn, a post-early sample predictive distribution of the number of faults of each type that will be discovered with additional time on test corresponding to that of the late sample is computed. Predictive estimates of the number of each fault discovered in the late sample are then compared with *observed* numbers of faults in the late sample. Predictive validation was done using NASA/GODDARD acceptance phase data.[3]

An exposition of the Bayesian approach is given in [3]. This paper is being revised to incorporate a study of the role of *reference priors* [4] on predictive distributions for number of faults of each fault type remaining in the software after completion of a test phase

Ongoing research tasks are:

1) Extension of predictive validation testing to other test phases.

2) Study how covariates such as *time to correct a fault* might be incorporated into the model.

3) Recast the finite population successive sampling scheme to capture reliability decay as well as reliability growth.

References

[1] Kaufman, G. M. (1996) Successive Sampling and Software Reliability. *J. Stat. Planning and Inference , Vol. 49, pp. 343-369.*

[2] Tanner, M.A. (1993) *Tools for Statistical Inference: Method for the Exploration of Posterior Distributions and Likelihood Function.* Springer-Verlag , 2nd Ed.

[3] Kaufman, G.M. (1995) *Bayes, Empirical Bayes and Software Reliability.* In progress. Presentations at Third World Meeting of the International Society for Bayesian Analysis, Oaxaca, Mexico, October 1995 and at the Institute of Management Science and Operations Research Spring Meeting in Washington DC, May 1996

[4] Berger, J.O. and Bernardo, J.M. (1992) *"Ordered Group Reference Priors with Application to the Multinomial Problem" Biometrika, Vol. 79, No. 1, pp 25-37.*