Comparative Analysis of Bayesian and Classical Approaches for Software Reliability Measurement

Thierry W. Ketchiozo
Reliability and Maintainability Division, NAVAIR, Patuxent River, MD and Department of Engineering Management and Systems Engineering George Washington University, Washington, DC

Timothy Eveleigh
Department of Engineering Management and Systems Engineering George Washington University, Washington, DC

Shahryar Sarkani
Department of Engineering Management and Systems Engineering George Washington University, Washington, DC

Thomas H. Holzer
Department of Engineering Management and Systems Engineering George Washington University, Washington, DC

Peter A. Keiller
Department of Systems and Computer Science, Howard University, Washington, D.C

ABSTRACT

Most software reliability growth models (SRGMs) use a classical approach to infer parameters and they have shown limitations in prediction accuracy. Bayesian approaches have been claimed to be more successful than classical approaches in certain situations as they allow the incorporation of prior information into models. The goal of this study was to investigate if the use of Bayesian methods can improve the predictability of SRGMs. The classical and Bayesian methods were used to estimate Musa Basic Execution Time (MBET) model's parameters (total number of failures and failure rate) using software inter-failure time. The time to the next failure was then predicted. The classical and Bayesian approaches showed comparable predictability performance, but the Bayesian approach was more consistent at estimating model parameters even with limited data. Our results demonstrate the usefulness of Bayesian approaches in software reliability prediction.

1-Introduction

Software errors cost the US economy more than $59 billion annually (Dhillon, 2013). Over the last two decades, several software reliability growth models (SRGMs) have been developed. The majority of these SRGM models either consistently overestimates or underestimates the quality of the software, as noted by Keiller and Mazzuchi (2002). To resolve this issue, it has been suggested that improving the accuracy of existing SRGMs is more important than developing new models (Okamura and Dohi, 2009). This may be achieved by using the most appropriate method to estimate models parameters, ranking the existing models, and improving their performance measure.

The estimation of the model's parameters is critical for the accuracy of software reliability prediction. The parameter's estimation of the mathematical function of the model should be done

in such a way that there is a perfect or almost perfect fit of the mathematical function with the failure data. Parameters' estimation can be done using either the classical or the Bayesian approaches. The maximum likelihood is the most widely used and popular parameter estimation technique, because of its relative mathematical simplicity as compared to other approaches. This classical method consists in solving a set of equations to define the values of the parameter that maximize the probability that the observed data are derived from a distribution with these parameters. The maximum likelihood is generally considered suited for larger sample size. However, to use this technique, the practitioner has to know the exact underlying distribution of the data of interest.
Bayesian approaches allow incorporating prior information into the parameters' analysis. However, only a handful of software reliability modeling studies have used Bayesian approaches. While some researchers have developed SRGMs based on Bayesian inference and claimed that the use of this approach improves reliability prediction, to our knowledge, a systematic comparison of classical and Bayesian approaches has never been conducted. It is therefore important to determine if the use of Bayesian approaches for parameter estimation could significantly improve SRGMs predictability. This study has two main goals: first to compare the accuracy of reliability prediction of software failure data sets using the classical and Bayesian approaches; and then to develop a framework for comparison of SRGMs.

2- Methods

Model and Datasets

The Musa Basic Execution Time (MBET) model, a non-homogeneous Poisson Process (NHPP) model, was used to analyze software failures from six datasets: three datasets from the historical collection of John Musa: USROOF, MDS1 and MDS3 (Keiller and Mazzuchi, 2002); one failure dataset from the Philips TV software development project (Almering, 2007), the SYS20B dataset (Sukert, 1976); and the PV400 dataset collection of software inter-failures time collected on a single-user workstation over 4 years by Peter Mellor at the Centre for Software Reliability (City University, London, England)(Keiller, 1995). All of these datasets represent software inter-failure times between user-perceived failures. A survey conducted by IEEE 1633 in 2014 indicated that the MBET is one of the most used and most trusted model amount hundreds of available SRGMs. The Computer Aided Software Reliability Estimation (CASRE) was used to implement the maximum likelihood approach. CASRE is the most popular and most used for software reliability measurements by a wide variety of researchers (Musa, 2004). Bayesian inference using a Markov Chain Monte Carlo (MCMC) simulation with Gibbs sampling was implemented using the OpenBUGS software (Lunn et al., 2013), the most widely used software package for fitting Bayesian models using MCMC.

3- Results

The MBET model parameters (the total number of failures (μ), and the failure rate (λ)) were estimated using either the classical (maximum likelihood) or the Bayesian method in six previously published datasets using different percentage of data (10%, 30%, 50%, 70%, and 90%). The use of different percentages of the datasets allows determining the volatility of the estimation.

Parameter Estimation

The model parameters' estimation are presented in Table 1. With the classical approach,

parameters were rarely estimated when less than 50% of the datasets were used. There was an important variability in the estimated total number of failures when compared to the observed data, indicating that this approach is not very reliable for estimating failure numbers. In contrast, the failure rate appears to be more stable as it usually varied within 2 folds across the different ranges of data used for inference.

In contrast to the classical approach, we were able to estimate model's parameters in all six datasets using as little as 10% of the data with the Bayesian approach. It is interesting to note that the total number of failures was very consistent with similar quantities for the observed data samples selected for all six datasets. Similarly, the failure rate was very consistent when different amounts of data were used for inference. Thus, the Bayesian approach appears to be very stable in estimating the model parameters.

Model Validation

Three of the tested datasets (MDS1, MDS3 and PV400) were used to validate the model. The validation was done on this subset because it is unlikely that additional information would be gained by using all six datasets as the same methodology was used for parameter estimation. The relative error (RE) was calculated as the ratio of the difference between the inferred total numbers of failures at the end of the testing with the total number of failures observed over the total number of failures observed. In general, the RE was small with both the classical and the Bayesian approaches for all the datasets (Figure 1). This indicates that the methodology used for parameter estimation is valid. Interestingly, the RE was generally smaller for the Bayesian method than it was with the classical. This is in line with our previous observation of more consistent prediction of total number of failures with Bayesian than with classical methods.

Reliability Growth

The Laplace test (Keiller et al, 2002) was used to assess reliability growth. This test allows determining if the system undergoes reliability growth. By using only datasets that show reliability growth, this method provides a better model fit because all the bad datasets are not further analyzed. In the six datasets tested, the Laplace test values were between -2 and 2, indicating a stable reliability growth (Lyu, 1996).

Time to Failure Prediction

Figure 2 represents the time to failure prediction using Bayesian, Classical approaches and the original cumulative time to failure. The observed cumulative time to failure follows an exponential curve for all the datasets, indicating that MBET is a suitable model to study these datasets. The prediction using the Bayesian approach appears to mimic more closely the observed cumulative failure than the prediction with the classical approach. This suggests that the Bayesian approach is not only more stable at estimating model parameters, but, also has a better predictability performance.

Comparative Analysis of the Classical and Bayesian Approaches

A quantitative assessment of the difference in prediction between the classical and Bayesian approaches was conducted on three of the datasets (MDS1, MDS3 and PV400) using an adaptive approach that attributes different weight on performance measures given their reliability prediction impact. Five performance criteria were used in this analysis including the Mean Square error (MSE), the Mean absolute Error (MAE), the Sum of Squared Error (SSE), the Bias and the predictive ratio risk (PRR). The weighted value for each criterion was then calculated as

the product of the weight of each performance measure by the performance measure. Then, the permanent value (z) was determined as the weighted mean value of all the criteria; smaller value indicates good fit and better predictability. The Bayesian approach had a better predictability for the MSD1 and the PV400 datasets (MSD1: z: 129429.325 and 0, PV400, z: 1667795373.4 and -9556.1 for classical and Bayesian, respectively). In contrast, the classical approach (z=-114.12) was better than the Bayesian approach (z=39968.49) for MDS3.

Sensitivity Analysis of the Bayesian Approach

The better predictability of the time to failure observed with the Bayesian approach was not surprising given its consistency at estimating parameters and its smaller relative error value. Importantly, the Bayesian approach performance heavily relies on the accuracy of the prior distribution. In this study, because no information was available on the data collection, we first used a uniform prior distribution. To evaluate the impact of the prior distribution on the Bayesian predictability, we conducted a one way sensitivity analysis using the same previous three datasets and changing only the failure rate prior distribution from the uniform to the Beta distribution while keeping the prior distribution for the total number of failures unchanged. The RE was then calculated (Figure 3). The RE using two different prior distributions were clearly different, indicating that the prior distribution can significantly impact model's predictability performance.

4-Discussion and Conclusion

The goal of this project was to evaluate the potential usefulness of Bayesian approaches for software reliability prediction. In this preliminary study, we used the MBET model because a recent survey by IEEE has indicated that this model is one of the most commonly used for software reliability prediction. The Bayesian approach prediction outperformed the classical approach prediction for two of the three software failure datasets analyzed. In Bayesian inference, parameter estimation is influenced by the prior distribution of the parameters which reflect prior knowledge on either model parameters or software architecture and components. In the absence of software metrics, knowledge of software architecture or expert opinion regarding a particular software, it would be almost impossible to accurately determine the prior distribution for the model's parameter. For instance, Singpurwalla used expert opinion to build prior distribution for the Logarithmic-Poisson Execution time model and reported that this model performed well at predicting the time to the next failure (Campodonico and Singpurwalla, 1994). In this study, the software was taken as a black box meaning we had no idea about the software structure, the software architecture, the software functionalities that have been tested, and the testing methodology and process. In these conditions it is impossible to pull up any prior information needed to build with confidence and accuracy the two model's parameter prior distribution, consequently, we used a non-informative prior distribution called: "Uniform Distribution" where we used data from the classical approach to guide our selection of the prior distribution parameters. The use of this prior distribution may have contributed to the apparent good performance of the Bayesian approach in this study.

According to Norman (1999), many studies in software reliability suffer from a variety of flaws ranging from model misspecification to the use of inappropriate data. To circumvent this issue, we used previously published datasets. However, as we were not involved in and have little information on the process used to collect software failure data, we cannot be sure that the correctness in the data collection was not a problem in this experimental study. For example,

only the last 100 failures time of the SYS20B dataset have been released to the public. So it is unclear if using the full failure dataset would have impacted the results. Software reliability engineering process has sardonically been described as a garbage in/garbage out process (Lyu, 2007) to indicate that the accuracy of its output is bounded to the precision of its input. Thus, data collection, clearly play a crucial role in the success of software reliability measurement. Therefore, as stipulated by the working group IEEE 1633 in 2014, it is time for researchers to come up with a unified framework that will be used by everybody for data collection. In the absence of such framework, each researcher will collect data using their own approach, leading to difficulties in comparing results across studies.

Previous studies have reported that parameters' estimation using the maximum likelihood inference was usually accurate when at least 33% of the data was used (Keiller and Mazzuchi, 2005), but, this observation was not supported by our study. This discrepancy could be related to the difference in algorithms used in the two studies. In this study, the maximum likelihood estimation was computed using the Newton-Raphson root-finding procedure, whereas Keiller and Mazzuchi used an updated version of the Nelder-Mead searching procedure. The Newton-Raphson method was reported to be faster than the Nelder-Mead method, but the Newton-Raphson method sometimes diverges, whereas the Nelder-Mead method always converges (Musa, 1987). It is important to identify ways to assess the convergence of the Bayesian estimation. Cowles (2013) suggested that to ensure that the right parameters are obtained, three different MCMC should be used at different initial values for the parameters and enough iteration should be run for the three chains to converge.

While many studies have investigated ways to enhance predictability of SRGMs using classical approaches for parameter estimation, to our knowledge this is the first study aiming at directly compare parameter estimation and reliability prediction using the classical and Bayesian approaches. Such comparative analyses are important to determine if more resources should be allocated to study Bayesian approaches in software reliability prediction. Our results indicate that the Bayesian approach is very stable to estimate parameters even when limited amount of data is available and can have a better predictability performance than classical approaches. However, two of the three datasets analyzed showed that Bayesian has a better predictability; maybe the collection of the data played a role or the model used. Therefore future studies are needed using larger datasets and SRGMs with different assumption should further explore if the use of Bayesian approaches could improve software reliability prediction. Indeed most of the SRGMs used in previous studies are static whereas we know that Software testing or Software development is a dynamic process; perhaps the use of SRGMs taking into account this important property might lead to greater predictability improvement.

References

Keiller, P.A. and Mazzuchi, T. A. (2002). Investigating a specific class of software Reliability growth models. 2002 IEEE proceedings annual Reliability and Maintainability Symposium

Dhillon, B.S (2013). Computer System Reliability Safety and Usability. CRC Press 2013

Okamura, H. and Dohi, T (2009). Software Reliability Modeling Based on Capture-Recapture Sampling. IEICE Transactions 92-A (7): 1615-1622

Almering, V. van Genuchten, M, Cloudt, G. and Sonnemans, P. (2007). Using software reliability growth models in Practice. IEEE software 24 (6), 82-88

Sukert, A.N. (1976). A Software Reliability Modeling Study. Rome Air Development Center, Technical Report RADC-TR-76-247, Rome, New-York

Lyu, M. (1996). Software Reliability Engineering Handbook, IEEE Computer Press

Campodonico, S and N. D. Singpurwalla (1994). A Bayesian Analysis of the Logarithmic-Poisson Execution Time Model Based on Expert opinion and failure data. IEEE transaction on Software Engineering, Vol. 20, NO. 9.

Musa, Iannino, Okumoto (1987). Software Reliability Measurement, Prediction, and application. McGraw-Hill

Lyu, M. R (2007) Software Reliability Engineering: A roadmap. IEEE computer Society.

Keiller, P.A and T. A. Mazzuchi (2005). Addressing the performance of two software reliability modeling methods. IEEE RAMS

Cowles, M.K (2013). Applied Bayesian Statistics. Springer

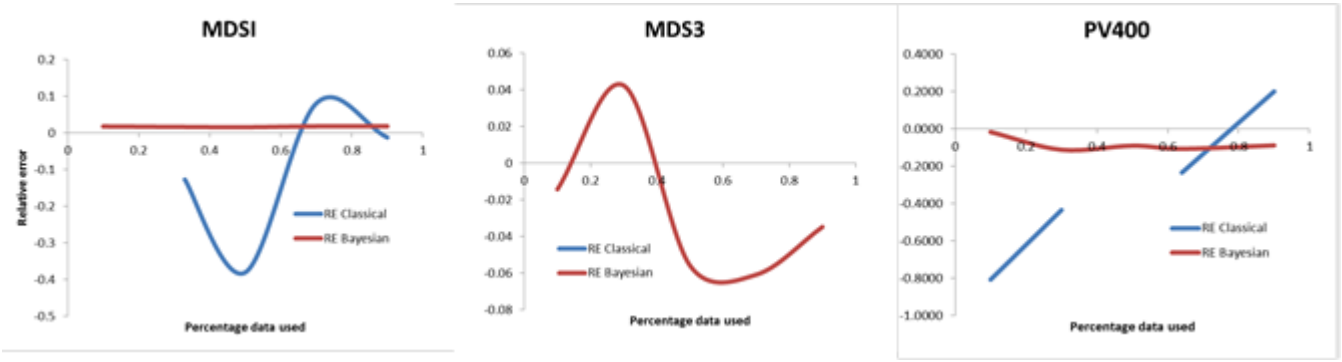

Figure 1: Validation of the Classical and Bayesian approaches

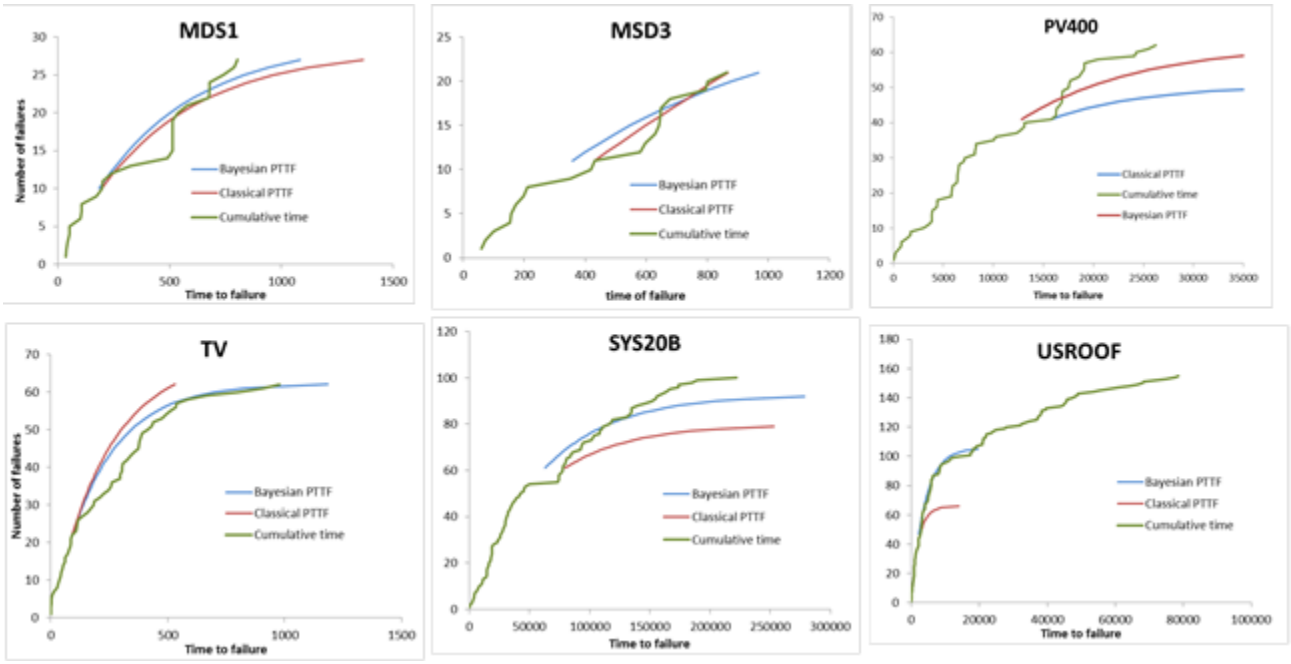

Figure 2: Time to the next failure prediction using Classical and Bayesian approaches.

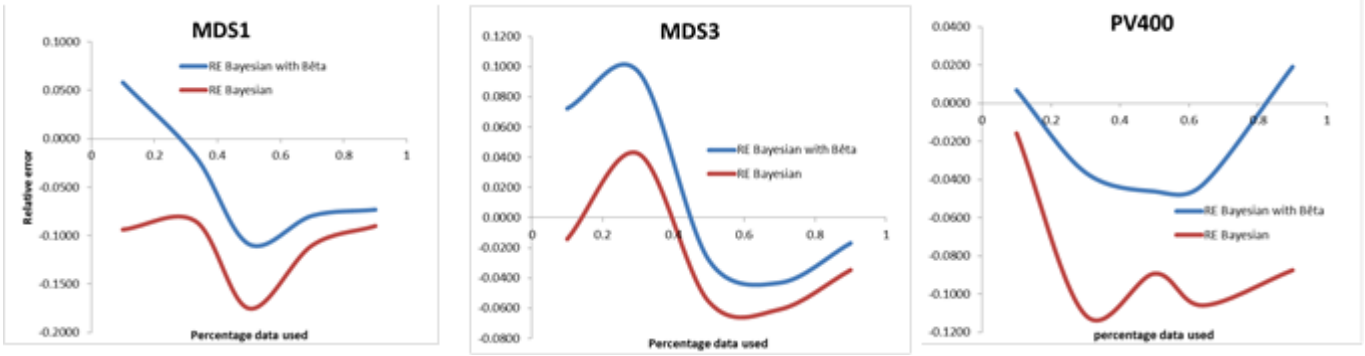

Figure 3: Sensitivity Analysis