Science Foundation Ireland

LERO Centre for Science, Engineering Technology

Work Package:
D5
An extension to the Goel and Okumoto1 model of software reliability

---

Project Reference:	10/CE/I1855
Date:	May 29, 2013
IBM SoW Title:	-
Total Number of Pages:	3

Executive Summary

This deliverable corresponds to a poster presented at the 33rd Conference on Applied Statistics in Ireland in 2013.

Statistical modelling of the reliability of soft- ware dates back to Jelinski and Moranda. Data can be either based on time intervals be- tween bugs, Type-I or the cumulative number of bugs seen to date, Type-II. This research looks at Type-II data extracted from the bug database for Mozilla and looks specifically at Firefox bugs. Research Question: How should we model the uncertainty surrounding the decision to re- lease software when we have historical information on previous releases and covariate information?

An extension to the Goel and Okumoto[1] model of software reliability

Seán Ó Ríordáin(seoriord@tcd.ie) and Simon P Wilson (swilson@tcd.ie)
Trinity College Dublin

1. Data and Research Question

Statistical modelling of the reliability of software dates back to Jelinski and Moranda[2]. Data can be either based on time intervals between bugs, Type-I or the cumulative number of bugs seen to date, Type-II. This research looks at Type-II data extracted from the bug database for Mozilla and looks specifically at Firefox bugs.

Research Question: How should we model the uncertainty surrounding the decision to release software when we have historical information on previous releases and covariate information?

2. Goel-Okumoto Model

The 1979 paper by Goel and Okumoto[1] was groundbreaking in modelling the failure phenomenon with a Non-Homogeneous Poisson Process (NHPP). The model makes the assumption that the rate bugs are found is proportional to the number remaining undiscovered in the system, i.e. when there are few bugs left, they are harder to find:

$$\frac{d}{dt}m(t) \quad \propto \quad a - m(t)$$

Where the mean value function $m(t)$ is:

$$m(t) = a(1 - be^{-bt})$$

Where a is the total number of bugs in the system and b can be interpreted as a measure of how hard the testers are working and finding bugs. The expected number of errors remaining at time t is:

$$E\{\bar{N}(t)\} = ae^{-bt}$$

Recognizing distinct test regimes, before and after release, with common a; but b before release is different to that after release which we call b'. We can then re-state the Goel and Okumoto model as:

$$\lambda(t) \quad = \begin{cases} ae^{-bt} & \text{if } t \leq t_{\text{Rel}} \\ ae^{-b't} & \text{if } t > t_{\text{Rel}} \end{cases}$$

$$E\big(N(t)\big) \qquad = \int_0^t \lambda(u)du$$

$$\begin{aligned} MVF = \quad & a(1 - e^{-bt}) \qquad \text{for } t \leq t_{\text{Rel}} \\ \text{or} = \quad & a(1 - e^{-bt_{Rel}}) + \int_{t_{\text{Rel}}}^t ae^{-b'u}du \\ = \quad & a(1 - e^{-bt_{Rel}}) + a(e^{-b't_{Rel}} - e^{-b't}) \\ & \text{for } t > t_{\text{Rel}} \end{aligned}$$

3. Firefox Data

Bug data from for Firefox was extracted from `https://bugzilla.mozilla.org/` for versions 5 to 20 , the "Rapid Releases", as of 6/3/2013.

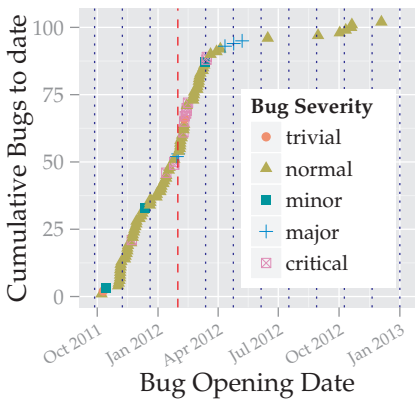

Figure 1: Cumulative Number of Bugs Reported against Firefox version 10

4. Fitting Goel-Okumoto

I have examined the Bayesian inference approach to fitting data to the Goel and Okumoto model carried out by McDaid and Wilson[3], for bugs in Mozilla Firefox. This was done by evaluating the likelihood over a semi-adaptive grid for a, b and calculatng the posterior.

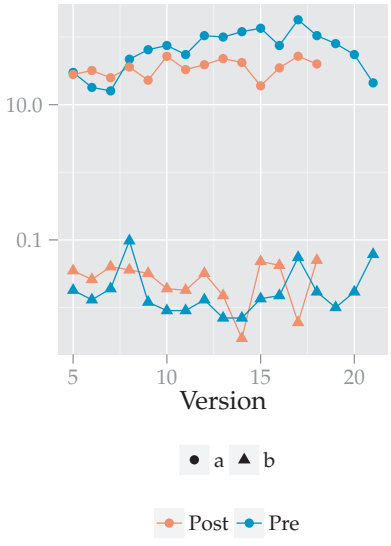

Figure 2: Progression of Goel and Okumoto a, b with Firefox Version number

5. Extensions

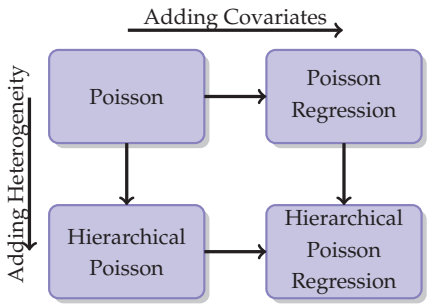

Figure 3: Relationships between the models.

Ravishanker[5] has demonstrated models using Markov Switching.

6. Conclusions

Murphy[4] discusses the difficulty in modelling software bugs. This research looks to use covariates and Markov Switching to better model the number of bugs in Open Source software and the rate that they will be found and to help improve decisions around the release process.

7. References

[1] Amrit L. Goel and Kazu Okumoto. Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE Trasactions on Reliability*, R-28(3):206–211, August 1997.

[2] Zo Jelinski and Paul B Moranda. Software reliability research. *Statistical computer performance evaluation*, pages 465–484, 1972.

[3] Kevin McDaid and Simon P. Wilson. Deciding how long to test software. *The Statistician*, 50(Part 2):117–134, 2001.

[4] Brendan Murphy. The difficulties of building generic reliability models for software. *Empirical Software Engineering*, 17(1-2):18–22, 2012.

[5] Nalini Ravishanker, Zhaohui Liu, and Bonnie K. Ray. Nhpp models with markov switching for software reliability. *Computational Statistics & Data Analysis*, 52(8):3988–3999, 2008.

Acknowledgement

This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie).