Discussion

Comments on '*Evolutionary neural network modelling for software cumulative failure time prediction*'

Bev Littlewood

Centre for Software Reliability, City University,

Northampton Square, London EC1V 0HB, UK

b.littlewood@csr.city.ac.uk

Abstract

This paper [1] purports to present a useful means of predicting the cumulative failure time function for software reliability growth. In fact, the nature of the 'prediction' is too simplistic to be of use. Furthermore, the authors' claims for the accuracy of the predictions appear to be without value.

1 Introduction

The authors' objective in this paper is to make one-step-ahead predictions of the cumulative failure plot of software reliability growth, i.e. (in the authors' notation) the plot of total elapsed time, x_i, as a function of failure number, i. The authors' approach to solving the problem is via evolutionary neural network modelling involving 'modification of Levenberg-Marquardt algorithm with Bayesian regularization.' In what follows I intend to show that:

- The prediction problem that is solved here is so simplistic as to be of no practical utility;
- The 'solution' to the problem is unnecessarily complex: a trivial alternative will perform as well;
- The assessment of predictive accuracy, used by the authors in their claims for the efficacy of their approach, is flawed.

2 Critique

Software reliability growth modelling concerns the stochastic process of successive failures of a computer program under some kind of testing or operational use. In its simplest form, as in [1], it is assumed that when a failure occurs the fault that caused the failure is removed with certainty. As time passes, therefore, the reliability will be seen to improve.

The stochastic process can be characterised by the successive inter-failure time random variables, T_1, T_2, …, T_i, …, or, alternatively, by the occurrence epochs of the successive failures, $X_1, X_2, \ldots, X_i, \ldots$ Clearly

$$X_j = \sum_{k=1}^{j} T_k$$

Reliability growth would show itself in the tendency for successive inter-failure times to become larger: e.g. they might be stochastically ordered.

If we denote by lower case variables the realisations of (upper case) random variables, e.g. x_i is a realisation of X_i, the authors describe their prediction problem as follows: '…we want to forecast x_{i+1} by use of $(x_1, x_2, \ldots, x_i)$ …'

This is an odd formulation: what does it mean to 'predict' the *realisation* of the random variable X_{i+1}? Since the realisation of X_i *is* known – it is x_i – any prediction of X_{i+1} is essentially equivalent to predicting the next inter-failure time, T_{i+1}. However such a prediction is expressed, it must take account of our uncertainty about the values these random variables will take. This seems to be completely absent from the authors' formulation of the problem.

This is unfortunate, because the authors make strong claims for their approach when compared with others. In fact they seem to be comparing apples and oranges. The probabilistic modelling approaches [2] attempt to solve the real – and much more difficult – problem of predicting in the face of uncertainty. Specifically, in the context of this problem, they obtain predictive *distributions* for random variables such as X_{i+1}, T_{i+1}.

It is also worth noting that these existing approaches allow more interesting predictions to be made than the simple one-step-ahead ones obtained here.

One interpretation of the authors' results might be that their approach gives a 'best guess' of the value that X_{i+1} will take. Unfortunately, they do not tell the reader what 'best' means here (it cannot simply mean that it optimises some objective function in their neural network, because that would be tautological). It does not appear to represent any particular point on the distribution of X_{i+1} (a predictive mean, or median, *might* be of some interest).

Even if one were to take the authors' claims at face value, what possible value is there in a one-step-ahead predictive plot of cumulative failure time? It does not, for example, even provide an estimate of current reliability. Since the plot is irregular, its 'slope' cannot be used to obtain an estimate of the rate of occurrence of failures.

Finally, the authors' claims for efficacy seem flawed. The goodness-of-fit comparisons of predicted with actual cumulative time plots are useless: almost any predictor would look good on such plots. For example, simply adding the previous inter-failure time, t_i, to x_i will give predictions of x_{i+1} that are virtually indistinguishable on the authors' goodness-of-fit plots from the ones they show.

The authors' relative error (RE) measure is similarly worthless as an absolute measure of accuracy. They define this as

$$RE = \left| \frac{\hat{x}_i - x_i}{x_i} \right|$$

where $\hat{x}_i$ is the predicted value of the cumulative failure time x_i. Since a one-step-ahead prediction of the cumulative failure time is equivalent, as noted above, to a prediction of the next inter-failure time, *RE* can easily be re-expressed as

$$RE = \left| \frac{\hat{t}_i - t_i}{\sum_{k=1}^{i} t_k} \right|$$

This clearly decreases, as *i* increases, in some stochastic sense. The authors' *RE* amounts to expressing 'error' in the prediction of the next inter-failure time *as a proportion of the sum of all previous inter-failure times*. It is a trivial observation that this 'error' can be made a small as you like by making *i* large enough. It is thus nonsense to claim, as the authors do, that small values of *RE* vindicate their approach.

This problem of validating the accuracy of predictions is one that has been studied extensively in the literature on probabilistic modelling of software reliability growth. Techniques of real depth and sophistication have been borrowed from developments in mathematical statistics: see [2, 3], for example.

3 Summary and conclusion

The authors purport to pose, and solve, a useful software reliability prediction problem. Unfortunately the problem they pose is unrealistically simplistic, as it ignores the essentially stochastic nature of the situation. Because of this the results have no practical utility. The authors' claims for the accuracy of their approach are flawed, even in their own very restricted terms.

I do not believe the paper was worthy of publication in RESS.

References

[1] L. Tian and A. Noore, "Evolutionary neural network modelling for software cumulative failure time prediction," *Reliability Engineering and System Safety*, vol. 87, pp. 45-51, 2005.

[2] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, "Evaluation of Competing Software Reliability Predictions," *IEEE Trans. on Software Engineering*, vol. 12, pp. 950-967, 1986.

[3] S. Brocklehurst and B. Littlewood, "New Ways to get Accurate Reliability Measures," *IEEE Software*, vol. 9, pp. 34-42, 1992.