Sādhanā Vol. 34, Part 2, April 2009, pp. 235–241. © Printed in India

Sequential Bayesian technique: An alternative approach for software reliability estimation

S CHATTERJEE*, S S ALAM† and R B MISRA‡

*Department of Applied Mathematics, Indian School of Mines University, Dhanbad 826 004
†Department of Mathematics, Indian Institute of Technology, Kharagpur 721 302
‡Reliability Engineering Centre, Indian Institute of Technology, Kharagpur 721 302
e-mail: chatterjee_subhashis@rediffmail.com; alam@maths.iitkgp.ernet.in; ravi@ee.iitkgp.ernet.in

MS received 8 October 2007; revised 15 July 2008

Abstract. This paper proposes a sequential Bayesian approach similar to Kalman filter for estimating reliability growth or decay of software. The main advantage of proposed method is that it shows the variation of the parameter over a time, as new failure data become available. The usefulness of the method is demonstrated with some real life data.

Keywords. Software reliability; Bayesian sequential estimation; Kalman filter.

1. Introduction

As computers are used in various fields of life including business and safety critical systems, software faults have become the major factor that causes critical problems. Hence, there exists an increasing demand for highly reliable software. Software reliability models provide quantitative measures of the reliability of a software system during its development phase. Research activities in the field of software reliability have been conducted since early seventies. Detail studies about software reliability are given in (Xie 1991, Musa *et al* 1987, Shooman 1968). Some important software reliability growth models (Jelenski & Moranda 1972, Shooman 1972, Schick & Wolverton 1978, Musa 1975, Littlewood & Verrall 1973, Xie 1987, Goel & Okumoto 1979, Singpurwalla & Soyer 1985, Yamada *et al* 1983, Yamada *et al* 1984, Yamada *et al* 1986, Yamada *et al* 1993) have been developed considering perfect debugging and immediate error removal. Incorporating some realistic issues like imperfect debugging and learning process of software developers some other important software reliability growth models (Chatterjee *et al* 1997, Sumita & Shantikumar 1986, Fakhre-Zakeri & Slud 1995, Zeephongsekul *et al* 1994, Xie *et al* 1993, Pham 1996, Chatterjee *et al* 1998, Gokhale *et al* 2006, Dai *et al* 2005, Xie *et al* 2004, Chatterjee *et al* 2004, Park & Lee 2003) have also been developed.

Software undergoes several stages of testing before it is put into operation. In every stage of testing, modification and correction are made with the hope of increasing reliability. It is very important to know, whether a particular modification or series of modifications lead to the growth of reliability, so that a software engineer can decide when to stop the process of testing. As errors are removed from the software the time between failures gradually increases. With the knowledge of time between failures, it is important to know

(i) whether the modifications made in software are beneficial,
(ii) whether the modifications lead to overall growth or decay of reliability, and
(iii) about next time between failures.

Considering these points researchers have proposed software reliability models using autoregressive process (Singpurwalla & Soyer 1985).

In this paper, sequential Bayesian estimation procedure (Soman & Misra 1993) is used for estimating reliability of software. Several regression models like forward section, backward eliminations, step-wise and all sub-set regressions are available in the literature. All these techniques are one shot or batch processing in nature, since the model parameter estimates are calculated are based on the entire data set. One has to repeat the whole process again if new data set is added to the old data set or to get a new estimate. This is computationally undesirable. The sequential estimation technique described here to accomplish this task is more efficient than the available regression techniques. A simple power law model has been used here. Application of the proposed technique has been illustrated with real life data for model validation.

2. General sequential estimation procedures

A sequential maximum a posteriori estimation procedure based on Bayesian approach is discussed here. The procedure is capable of utilizing the prior information. Let the general regression model be

$$Y = B_0 + \sum_{j=1}^{q-1} B_j X_j + \varepsilon. \tag{1}$$

The equation for the Bayesian estimation of the model parameters, $\hat{B}$, is given as

$$\hat{B} = M + PX^T Q^{-1}(Y - XM), \tag{2}$$

where P is the covariance matrix of estimators $(q \times q)$, given as

$$P = (X^T QX^{-1} + V^{-1}). \tag{3}$$

$\hat{B}$: estimated parameter vector $(q \times 1)$
M : mean value of parameter vector $(q \times 1)$ known from the prior information
X : independent variable matrix $(n \times 1)$
V : covariance matrix of Bknown from prior information
Q : covariance matrix of errors.

Substituting $B_i = B_{i+1,} M = M_i, Y = Y_{i+1,} P = P_{i+1,} V = P_{i,} X = X_{i+1}$ and $Q = C_{i+1}$ we get the recursive form of equation (2) and (3). Here C is a $m \times m$ diagonal covariance matrix of error and m is the number of observations. Substituting the above expressions in equation (2) and (3) we get

$$B_{i+1} = B_i + P_{i+1} X_{i+1}^T C_{i+1} (Y_{I+1} - X_{i+1} B_i) \tag{4}$$

and

$$P_{i+1} = (X_{i+1}^T C_{i+1}^{-1} X_{i+1} + P_i^{-1})^{-1}. \tag{5}$$

From matrix inversion theorem we know that,

$$(A + BCD)^{-1} = A^{-1} - A^{-1} B (C^{-1} + DA^{-1} B)^{-1} DA^{-1}.$$

Hence equation (5) may be written as follows

$$P_{i+1} = P_i - P_i X_{i+1}^T (X_{i+1} P_i X_{i+1}^T + C_{i+1}). \tag{6}$$

Let $R = P_i X_{i+1}^T C_{i+1} \& H = X_{i+1}$. Then the following matrix identity holds

$$(I + RH)^{-1} R = R(I + HR)^{-1}.$$

Therefore, substituting the values of R and H we get,

$$\begin{aligned}
(I + P_i X_{i+1}^T C_{i+1} X_{i+1})^{-1} P_i X_{i+1}^T C_{i+1} &= P_i X_{i+1}^T C_{i+1}^{-1} (I + X_{i+1} P_i X_{i+1}^T C_{i+1}^{-1})^{-1} \\
(X_{i+1}^T C_{i+1} X_{i+1} + P_i^{-1})^{-1} P_i X_{i+1}^T C_{i+1} &= P_i X_{i+1}^T C_{i+1}^{-1} (I + X_{i+1} P_i X_{i+1}^T C_{i+1}^{-1})^{-1} \\
(X_{i+1}^T C_{i+1}^{-1} X_{i+1} + P_i)^{-1} P_i^{-1} X_{i+1}^T C_{i+1}^{-1} &= P_i X_{i+1}^T C_{i+1}^{-1} C_{i+1} (C_{i+1} + X_{i+1} P_i X_{i+1}^T)^{-1} \\
P_{i+1} X_{i+1}^T C_{i+1}^{-1} &= P_i X_{i+1}^T (X_{i+1} P_i X_{i+1}^T + C_{i+1}^{-1})^{-1}. \qquad (7)
\end{aligned}$$

Substituting equations (6) & (7) in (4) & (5) we get,

$$A_{i+1} = P_i X_{i+1}^T \tag{8}$$

$$D_{i+1} = C_{i+1} + X_{i+1} A_{i+1} \tag{9}$$

$$K_{i+1} = A_{i+1} D_{i+1} \tag{10}$$

$$E_{i+1} = Y_{i+1} - X_{i+1} B_i \tag{11}$$

$$B_{i+1} = B_i + K_{i+1} E_{i+1} \tag{12}$$

$$P_{i+1} = P_i - K_{i+1} A_{i+1}^T. \tag{13}$$

Equations (8) to (13) are the governing equations for the sequential estimation procedure of the parameters. If the number of observations is one then no matrix inversion is involved and the computation becomes efficient. Thus for one observation, equation (8) to (13) may be rewritten as follows;

$$A_{i+1} = \sum P_{uk,i} X_{k,i+1} \tag{14}$$

$$D_{i+1} = \sigma_{i+1}^2 + \sum X_{k,i+1} A_{k,i+1} \tag{15}$$

$$K_{u,i+1} = \frac{A_{u,i+1}}{D_{i+1}} \tag{16}$$

$$E_{i+1} = (Y_{i+1} - \sum X_{k,i+1} B_{k,i}) \tag{17}$$

$$B_{u,i+1} = B_{u,i} + K_{u,i+1} E_{i+1} \tag{18}$$

$$P_{uv,i+1} = P_{uv,i} - K_{u,i+1} A_{v,i+1}, \tag{19}$$

where $u = 1, 2, 3, \ldots, q$, $v = 1, 2, 3, \ldots, q$, q is the number of parameters and σ_{i+1}^2 is the variance of Y_{i+1}. Here in equation (15) S is used instead of σ_{i+1}^2 to denote the error variance obtained from linear regression method. So equation (15) becomes

$$D_{i+1} = S + \sum X_{k,i+1} A_{k,i+1}. \tag{15A}$$

3. Estimation of model parameters

In the following paragraphs the description of the model is followed by the parameter estimation using proposed algorithm.

Let $X_t = X_{t-1}^{\theta} \delta$ where θ is constant and values of $\theta > 1$ means growth of reliability and $\theta < 1$ means decay of reliability. δ is the error due to some uncertainty in power law. Taking natural logarithm on both sides we get

$$\log X_t = \theta \log X_{t-1} + \log \delta \tag{20}$$

$$\text{or } Y_t = \theta Y_{t-1} + B_1, \tag{21}$$

where $Y_t = \log X_t$, $B_1 = \log \delta$.

To apply the above-mentioned algorithm for general sequential procedure given in equations (14) to (19) the expression (21) becomes

$$Y = B_1 X_1 + B_2 X_2,$$

where Y is Y_t, $B_1 = \log \delta$, $B_2 = \theta$, $X_2 = Y_{t-1}$ and X_1 is a dummy variable taking a constant value 1. Here t denotes the stage of testing and X_t denotes the time between failures. For illustration purpose System 40 data (Musa 1979) is used. The value of Y_t i.e. the original failure data is given in table 1 and the estimated value of B_1 and B_2 are given in table 2 for each stage of testing t. Figure 1 shows the variation of B_2 i.e. θ with t.

4. Conclusion

A Bayesian sequential estimation procedure for estimating software reliability is developed and illustrated with System 40 data of (Musa 1979). The main advantage of this method over others is, it can use the prior estimates of the parameters and can show the variation of parameters over a time as new failure data are available. The objective of the work is to

Table 1. Value of Y_t corresponding to stage of testing t.

t	Y_t	t	Y_t	t	Y_t	t	Y_t	t	Y_t
1	9·574	22	8·046	43	4·700	64	9·574	85	12·720
2	9·104	23	10·845	44	10·002	65	10·450	86	14·199
3	7·965	24	9·741	45	11·012	66	10·586	87	11·370
4	8·648	25	7·544	46	10·862	67	12·720	88	12·202
5	9·989	26	8·594	47	9·437	68	12·598	89	12·279
6	10·196	27	11·039	48	6·664	69	12·086	90	11·366
7	11·639	28	10·119	49	9·229	70	12·276	91	11·366
8	11·627	29	10·178	50	8·967	71	11·960	92	14·411
9	6·492	30	5·894	51	10·353	72	12·024	93	8·333
10	7·901	31	9·546	52	10·987	73	9·287	94	8·07
11	10·267	32	9·619	53	12·607	74	12·495	95	12·202
12	7·683	33	10·385	54	7·154	75	14·556	96	12·783
13	8·89	34	10·636	55	10·003	76	13·327	97	13·258
14	11·59	35	8·333	56	9·86	77	8·946	98	12·753
15	8·349	36	11·314	57	7·86	78	14·782	99	10·353
16	9·043	37	9·487	58	10·575	79	14·896	100	12·489
17	9·602	38	8·139	59	10·929	80	12·139		
18	9·379	39	8·671	60	10·660	81	9·798		
19	8·586	40	6·461	61	12·497	82	12·09		
20	8·787	41	6·461	62	11·374	83	11·382		
21	8·779	42	7·965	63	11·915	84	13·367		

Table 2. Estimated values of δ and θ corresponding to each stage of testing t.

t	δ	θ	t	δ	θ	t	δ	θ	t	δ	θ	t	δ	θ
1	1·01	1·02	22	1·0144	0·9952	43	1·0203	0·9845	64	1·0317	0·9847	85	1·0402	0·9841
2	1·0206	1·0586	23	1·0138	0·9927	44	1·0182	0·9786	65	1·0318	0·9805	86	1·0403	0·9835
3	1·0174	1·043	24	1·0164	1·0018	45	1·0244	0·9841	66	1·0322	0·9820	87	1·0401	0·9857
4	1·0134	1·0237	25	1·0158	0·9971	46	1·0248	0·9868	67	1·0322	0·9825	88	1·0411	0·9817
5	1·0145	1·0285	26	1·0143	0·9892	47	1·0248	0·9867	68	1·0325	0·9862	89	1·0412	0·9828
6	1·0168	1·0392	27	1·0154	0·9927	48	1·0246	0·9836	69	1·0325	0·9862	90	1·0412	0·9831
7	1·0164	1·0372	28	1·0173	0·9997	49	1·0235	0·9782	70	1·0326	0·9855	91	1·0412	0·9822
8	1·0182	1·0471	29	1·0169	0·9962	50	1·0257	0·9819	71	1·0326	0·9861	92	1·0412	0·9824
9	1·0174	1·0415	30	1·0169	0·9965	51	1·0257	0·9817	72	1·0326	0·9872	93	1·0415	0·9859
10	1·0096	0·9931	31	1·0147	0·9828	52	1·0264	0·9845	73	1·0326	0·9861	94	1·0439	0·9776
11	1·0118	0·9998	32	1·0187	0·9894	53	1·0263	0·9842	74	1·0328	0·9819	95	1·0439	0·9776
12	1·0152	1·0125	33	1·0188	0·9899	54	1·027	0·9895	75	1·034	0·9859	96	1·0464	0·9806
13	1·0119	0·9946	34	1·0193	0·9923	55	1·0278	0·9764	76	1·0336	0·9895	97	1·0464	0·9814
14	1·0136	1·0005	35	1·0194	0·9932	56	1·03	0·9803	77	1·0342	0·9874	98	1·0467	0·9821
15	1·0167	1·0144	36	1·0186	0·9866	57	1·03	0·9804	78	1·0354	0·9803	99	1·0472	0·9818
16	1·0137	0·9928	37	1·0207	0·9935	58	1·0294	0·9771	79	1·038	0·9867	100	1·0486	0·9793
17	1·0145	0·996	38	1·0202	0·9882	59	1·0312	0·9810	80	1·0378	0·9872	101	1·0472	0·9814
18	1·0151	0·9987	39	1·0196	0·9851	60	1·0312	0·9820	81	1·0392	0·9826			
19	1·0149	0·9976	40	1·02	0·9864	61	1·0312	0·9818	82	1·0393	0·9795			
20	1·0142	0·9943	41	1·0188	0·9819	62	1·0312	0·9854	83	1·0401	0·9822			
21	1·0144	0·9951	42	1·0189	0·9821	63	1·0317	0·9834	84	1·0401	0·9815			

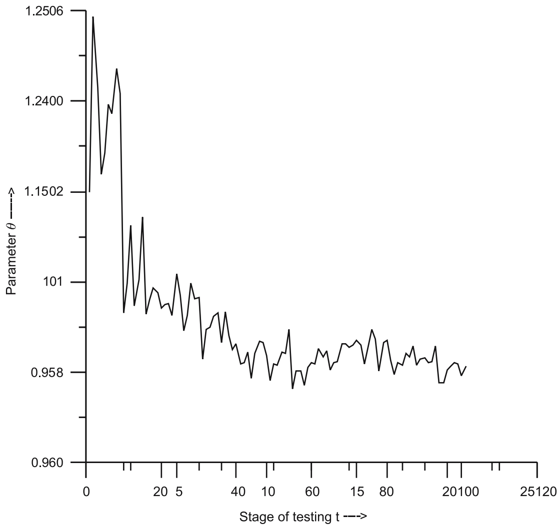

Figure 1. Variation of θ w.r.t. stage of testing t.

establish an alternative simplified approach for estimating reliability of software. The proposed method will be computationally very simple and efficient for software engineers. One can use this technique at any stage of testing.

Author acknowledges University Grants Communication (UGC), New Delhi for financial help to Dr S Chatterjee in the project number F.No.33-115/2007(SR).

References

Chatterjee S, Misra R B, Alam S S 1997 Joint effect of test effort and learning factor on software reliability and optimal release policy. *Int. J. Sys. Sci.* 28(4): 391–396

Chatterjee S, Misra R B, Alam S S 1998 A generalized shock model for software reliability. *Comput. Elect. Eng.-An Int. J.* 24: 363–368

Chatterjee S, Misra R B, Alam S S 2004 N-version programming with imperfect debugging. *Comput. Elect. Eng.* 30(6): 453–463

Dai Y S, Xie M, Poh K L 2005 Modelling and analysis of correlated software failures of multiple types. *IEEE Trans. Rel.* 54(1): 100–106

Fakhre-Zakeri I, Slud E 1995 Mixture models for reliability of software with imperfect debugging: Identifiably of parameters. *IEEE Trans. Rel.* 44: 104–113

Goel A L, Okumoto K 1979 A time-dependent error detection rate model for software reliability and other performance measure. *IEEE Trans. Rel.* R-28: 206–211

Gokhale S S, Lyu M R, Trivedi K S 2006 Incorporating fault debugging activities into software reliability models: A simulation approach. *IEEE Trans. Rel.* 55(2): 281–292

Jelinski Z, Moranda P B 1972 Software reliability research statistical computer performance evaluation. W Freiberger, Ed. Academic, NY, 465–484

Littlewood B, Verrall J L 1973 A bayesian reliability growth model for computer software. *Appl. Statist.* 22: 332–346

Musa J D 1975 A theory of software reliability and its application. *IEEE Trans. Software Eng.* SE-1: 312–327

Musa J D 1979 Software reliability data, *New York: DACS, RADC/ISISI*

Musa J D, Iannino A, Okumoto K 1987 Software reliability measurement. Prediction, Application, McGraw-Hill Int. Ed.

Park D H, Lee C H 2003 Markovian imperfect software debugging model and its performance measures. *Stochastic Analysis And Applications* 21(4): 849–864

Pham H 1996 A software cost model with imperfect debugging random life cycle and penalty cost. *Int. J. Sys. Sci.* 27: 455–463

Schick G J, Wolverton R W 1978 An analysis of competing software reliability model. *IEEE Trans. Software Eng.* SE-4: 104–120

Shooman M L 1968 Probabilistic reliability: An engineering approach. (NY: McGraw Hill)

Shooman M L 1972 Probabilistic models for software reliability prediction. Statistical Computer Performance Evaluation, W Freiberger, Ed. Academic, NY, 485–502

Singpurwalla N D, Soyer R 1985 Assessing (Software) reliability growth using a random co-efficient autoregressive process and its ramifications. *IEEE Trans. Software Eng.* SE-11: 1456–1464

Soman K P, Misra K B 1993 On Bayesian estimation of system reliability. *Microelectronic Reliability* 33: 1455–1459

Sumita U, Shantikumar J G 1986 A software reliability model with multiple-error introduction & removal. *IEEE Trans. Rel.* R-35: 459–462

Xie M 1987 A shock model for software reliability. *Microelectronic Reliability* 27: 717–724

Xie M 1991 Software reliability modelling. World Scientific Press

Xia G, Zeephongsekul P, Kumar S 1993 Optimal software release policy with a learning factor for imperfect debugging. *Microelectronic Reliability* 33: 81–86

Xie M, Dai Y S, Poh K L 2004 Distributed system availability in the case of imperfect debugging process. *International Journal of Industrial Engineering-Theory Applications and Practice* 11(4): 396–405

Yamada S, Ohba M, Osaki S 1983 S-shaped reliability growth modelling for software error detection. *IEEE. Trans. Rel.* R-32: 475–478

Yamada S, Ohba M, Osaki S 1984 S-shaped software reliability growth models and their applications. *IEEE. Trans. Rel.* R-33: 289–291

Yamada S, Ohteria H, Narihisa H 1986 Software reliability growth models with testing-effort. *IEEE. Trans. Rel.* R-35: 19–23

Yamada S, Hishitani J, Osaki S 1993 Software reliability growth with a weibull test-effort: A model application. *IEEE. Trans. Rel.* R-42: 100–105

Zeephongsekul P, Xia G, Kumar S 1994 Software-reliability growth model: Primary failures generate secondary-faults under imperfect debugging. *IEEE Trans. Rel.* 43: 408–413