7th WSEAS Int. Conf. on MATHEMATICAL METHODS and COMPUTATIONAL TECHNIQUES IN ELECTRICAL ENGINEERING, Sofia, 27-29/10/05 (pp362-366)

An Autoregressive Time Series Software Reliability Growth Model with Independent Increment

GUO JUNHONG[1,2], LIU HONGWEI[1], YANG XIAOZONG[1]
School of Computer Science and Engineering
[1]Harbin Institute of Technology
Harbin, Xidazhi street No.92 150001
School of Computer Science and Technology
[2]Heilongjiang University
Harbin, Xuefu road No.74 150080
CHINA

Abstract: - Goel-Okumoto model is a nonhomogenuous Poisson process software reliability growth model, which is commonly used in software reliability analysis and prediction. But it requires a large number of failure data and its parameters estimation methods are very complicated. This paper transforms Goel-Okumoto model into one-order autoregressive stochastic time series model with independent increment. Numerical simulation examples show the simplicity and effectiveness of the proposed method. Software reliability time series modeling provide new method to solving the problem of software reliability.

Keywords: - Software reliability growth model; Autoregressive time series model; Goel-Okumoto model; Software reliability evaluation

1 Introduction

Software reliability is a key factor in software development process and software quality, so it has played an important role in the process of software developing and software testing. The technique of software reliability assessment by using software reliability growth models(SRGMs) is one of the most important approaches of software reliability engineering [1,2].

Many SRGMs have been created since the first one appeared in 1972 [3]. SRGMs represent the relationship between the time span of software testing and the number of detected errors. SRGMs based on nonhomogeneous Poisson process(NHPP) are most commonly used. Goel-Okumoto NHPP model has a strong influence on software reliability modeling [4].

To apply Goel-Okumoto model, it is necessary to know how well the model suits actual observation data. And in order to obtain accurate software reliability estimation, it requires a large number of failure data which are not usually available until the system has been tested for a long time. Many software reliability engineers are more interested in estimating the software reliability as early as possible [5,6].

The parameter estimation techniques of Goel-Okumoto SRGM use Maximum Likelihood Estimation(MLE) and Least Squares Estimation (LSE) usually [7]. Its calculation methods are very complex.

This paper proposes a method for early and more accurate software reliability prediction by making use of time series analysis. Its calculation methods are simple. This paper is organized as follows. Section 2 presents software reliability modeling based on time series and its feasible. In section 3 transforms Goel-Okumoto model to auto-regressive(AR) time series model with independent increment and section 4 simulation of testing data is given. Finally, a brief conclusion of the obtained is presented.

2 Software Rreliability Modeling Based On Time Series And Its Feasibility

Time series analysis theory is a method of describing statistics character of dynamic data, which can set up time series model from limited sample data, its advantage is convenience and practicality. Time series analysis method is well studied in some statistical literatures. However, its use in reliability engineering is rather limited.

Time series prediction can be stated as follows: given a finite sequence $X_1, X_2, X_3, ..., X_t$, predicting the continued sequence $X_{t+1}, X_{t+2}, \ldots$. For example, $\{X_t\}$ can be viewed as the stochastic failure intervals or the number of failures per time interval [8].

During software reliability analysis, software failure data are cumulative number of software failures or failure intervals mainly. That is to say, Software reliability failure data are discrete data sequence, whether it is steady or not, which can be used to evaluate software reliability by applying proper time series model [9]. We'll deduce the relevant software reliability time series model from Goel-Okumoto model. This modeling method prove classical SRGMs also can be presented in the way of time series model.

3 Transform Goel-Okumoto Model To AR Time Series Model With Independent Increment

Goel-Okumoto software reliability growth model is one of the famous NHPP models with the following mean value function

$$m(t) = a(1 - e^{-bt}) \quad a > 0, b > 0 \tag{1}$$

where a is interpreted as the number of initial faults and the parameter b is the fault detection rate which is related to the reliability growth rate in the testing process.

Goel-Okumoto model is based on the following assumptions [10]:

1. The error removal process follows a non-homogeneous Poisson process.
2. The software system is subject to failures at random time caused by errors remaining in the system.
3. The proportionality is a constant over time. The mean number of errors detected in the time interval $(t, t + \Delta t]$ by the current test-effort is proportional to the mean number of remaining errors in the system.
4. Each time a failure occurs, the error which caused it is removed immediately, and no new errors are introduced.

Since the mean value function of Goel-Okumoto model is $m(t) = a(1 - e^{-bt})$, the cumulative number of software failures on time $t = kT$ will be $m(kT) = a(1 - e^{-bkT})$, which is denoted as $M(k)$, where T is an unit time. If defining $X(k)$ as number of software failures in time interval $[(k-1)T, kT)$, and $\mathrm{X}(0) = 0$, then $X(k) = M(k) - M(k-1)$, and

$$\begin{aligned} X(k) &= M(k) - M(k-1) \\ &= a(1 - e^{-bkT}) - a(1 - e^{-b(k-1)T}), \\ &= ae^{-b(k-1)T}(1 - e^{-bT}) \end{aligned}$$

then $M(k)$ can be described as follows:

$$\begin{aligned} M(k) &= M(k-1) + X(k) \\ &= e^{-bT} M(k-1) + (1 - e^{-bT})a \end{aligned} \tag{2}$$

Let $\theta = e^{-bT}$ and $\delta = (1 - e^{-bT})a$, equation (2) can be rewritten as

$$M(k) = \theta M(k-1) + \delta \tag{3}$$

so the Goel-Okumoto model is transformed to the first order AR time series model with independent increment δ,

$$M(k) = \theta M(k-1) + \delta + e(k) \tag{4}$$

where $e(k)$ is zero-average white noise. The parameter θ and δ can be estimated by exponential-weighted recursive least-squares method [11]. Assuming $\Phi(k) = [M(k-1) \quad 1]^{\mathrm{T}}$ and $\Theta(k) = [\theta \quad \delta]^T$, equation (4) can be rewritten as

$$M(k) = \Phi^T(k)\Theta(k) + e(k) \tag{5}$$

and the least-squares estimation value of $\Theta(k)$ is computed as follows:

$$\begin{aligned} \hat{\Theta}(k+1) &= \hat{\Theta}(k) + k(k+1) \\ &\quad [M(k+1) - \Phi^T(k+1)\hat{\Theta}(k)] \end{aligned} \tag{6}$$

$$K(k+1) = \frac{P(k)\Phi(k+1)}{\omega + \Phi^T(k+1)P(k)\Phi(k+1)} \tag{7}$$

$$P(k+1) = \frac{1}{\omega}[I - K(k+1)\Phi^T(k+1)]P(k) \tag{8}$$

where ω is forgetting factor, $0 < \omega < 1$, and $\omega = 0.56$ in the simulation. The initial value is $\hat{\Theta}(0) = [0 \quad 0]^T$ and $P(0) = 10^4 I$, where I is an unit matrix.

The parameters of Goel-Okumoto model can be estimated by the following equation (9) and equation (10).

$$\hat{b} = -\frac{1}{T}\ln(\hat{\theta}_N) \tag{9}$$

$$\hat{a} = \frac{\hat{\delta}_N}{1-\hat{\theta}_N} \tag{10}$$

The $p-step$ fore-predication of the cumulative number of failures is computed by $\hat{M}(N+p) = \hat{\theta}_N \hat{M}(N+p-1) + \hat{\delta}$.

4 Simulations

Software testing data in table 1 comes from Data7 in chapter 17 of document [12], where Day is the test time in days and CF is cumulative number of software failures.

Table 1 A set of software failure data

Day	CF	Day	CF	Day	CF	Day	CF
0	4	**38**	186	**65**	374	**87**	494
2	11	**40**	193	**66**	379	**88**	496
4	21	**41**	200	**67**	386	**89**	497
9	34	**43**	205	**68**	393	**90**	508
11	42	**45**	212	**69**	407	**91**	509
16	55	**48**	218	**70**	420	**92**	511
17	59	**49**	224	**71**	434	**93**	513
20	66	**50**	228	**72**	445	**94**	517
22	74	**51**	240	**73**	447	**95**	518
23	75	**52**	246	**74**	451	**96**	522
24	81	**53**	253	**75**	455	**97**	523
26	94	**54**	261	**76**	458	**98**	524
27	101	**55**	272	**77**	464	**99**	526
28	110	**56**	278	**78**	470	**100**	527
29	118	**57**	287	**79**	473	**101**	528
31	123	**58**	294	**80**	476	**102**	529
32	133	**59**	306	**81**	480	**103**	530
33	140	**60**	318	**82**	481	**104**	532
34	151	**61**	333	**83**	483	**105**	533
35	156	**62**	347	**84**	484	**107**	535
36	164	**63**	354	**85**	486		
37	177	**64**	363	**86**	491		

Table 2 Comparison of observed data and estimated data of Goel-Okumoto (GO) model and AR model

Day	Observed Data	GO Estimated Data	GO Estimated Error	AR Estimated Data	AR Estimated Error
5	21	64.1607	43.1607	26.5674	5.5674
15	42	172.2497	130.2497	42.2006	0.2006
25	81	258.0441	177.0441	85.3148	4.3148
35	156	326.1426	170.1426	162.4513	6.4513
45	212	380.1950	168.1950	206.1012	-5.8988
55	272	423.0986	151.0986	269.1035	-2.8965
65	374	457.1528	83.1528	371.3654	-2.6346
75	455	484.1830	29.1830	454.6383	-0.3617
85	486	505.6379	19.6379	485.0549	-0.9451
95	518	522.6676	4.6676	519.6806	1.6806

Table 3 Comparison of observed data and predicted data of Goel-Okumoto (GO) model and AR model

Day	Observed Data	GO Predicted Data	GO Predicted Error	AR Predicted Data	AR Predicted Error
98	524	527.0553	3.0553	525.9227	1.9227
99	526	528.4516	2.4516	526.8576	0.8576
100	527	529.8160	2.8160	527.6491	0.6491
101	528	531.1492	3.1492	528.3191	0.3191
102	529	532.4520	3.4520	528.8865	-0.1135
103	530	533.7251	3.7251	529.3667	-0.6333
104	532	534.9691	2.9691	529.7734	-2.2266
105	533	536.1846	3.1846	530.1176	-2.8824
106	533	537.3725	4.3725	530.4091	-2.5909
107	535	538.5332	3.5332	530.6558	-4.3442

Applying time series AR model and Goel-Okumoto model discussed above to the actual testing data, several numerical illustrations of software reliability evaluation are showed as the following.

In Table 2, we select cumulative number of software failures from day 0 to day 97 above Table 1, and obtain the comparison of observed data and estimated data of Goel-Okumoto model and AR model. We can find that the estimated errors of AR model are far much less than observed data and the ones got by Goel-Okumoto model. In other words, Goel-Okumoto model has large estimated errors, especially in the beginning of estimation. And in Table 3, we use above observed data to gain prediction data from day 98 to day 107. We have the same comparison of observed data and predicted data and find that the predicted data of AR model approach the observed data better than Goel-Okumoto model.

Fig.1 and Fig.2 illustrate failures data of the two models, including observed data and estimated data and predicted data. Because AR model can attach importance to near failures data, we can see that the model has better fitting than Goel-Okumoto model. In Fig.3 and in Fig.4, we can obtained the parameter estimation of Goel-Okumoto model, $\hat{a}$=588.2016, $\hat{b}$=0.0231. And in Fig.5 and Fig.6, we also have the parameter estimation of AR model, $\hat{\theta}$= 0.8466 and $\hat{\delta}$=81.6023. Using the parameters estimation of AR model, we can get more accurate results.

As in many cases, people pay more attention to the relative error between observed data and estimated data. To compare further, define

$$\text{relative error } = \frac{\hat{M}(i) - M(i)}{M(N)}.$$

Fig.7 and Fig.8 show the relative error of Goel-Okumoto model and AR model. We can find the relative error of AR model is less than that of the Goel-Okumoto model obviously. The relative errors of Goel-Okumoto model are positive number. However, some the relative errors of AR model are positive and some are negative. These data wave near by zero and is very small.

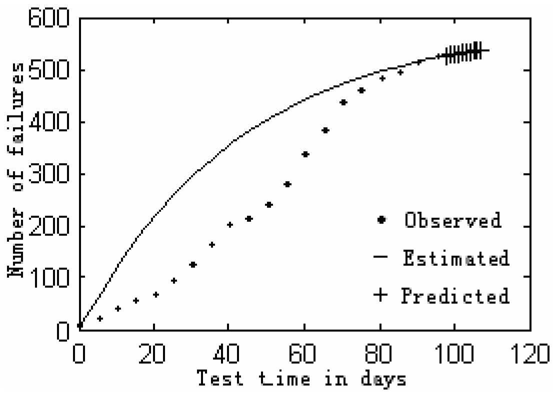

Fig.1 Goel-Okumoto model

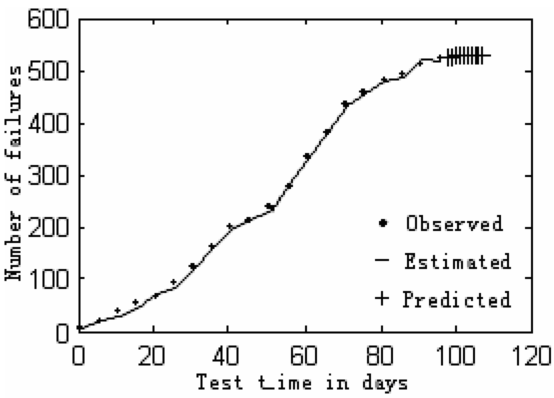

Fig.2 AR time series model with independent increment

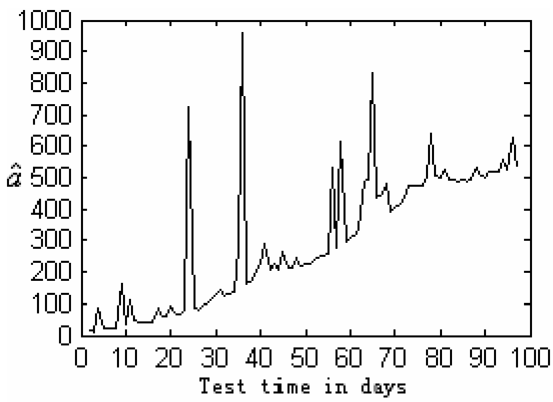

Fig.3 Estimated $\hat{a}$

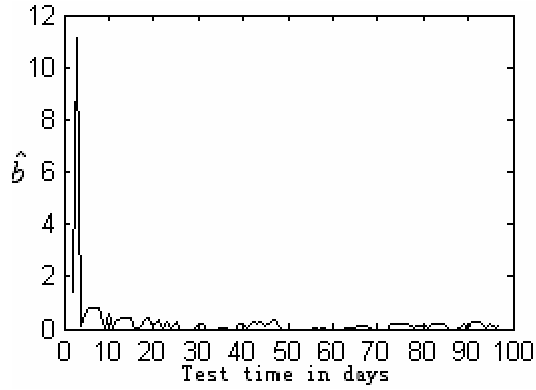

Fig.4 Estimated $\hat{b}$

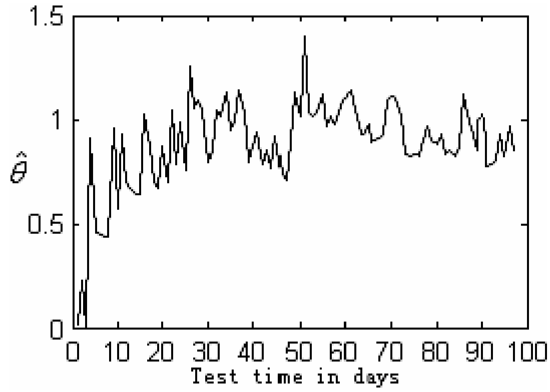

Fig.5 Estimated $\hat{\theta}$

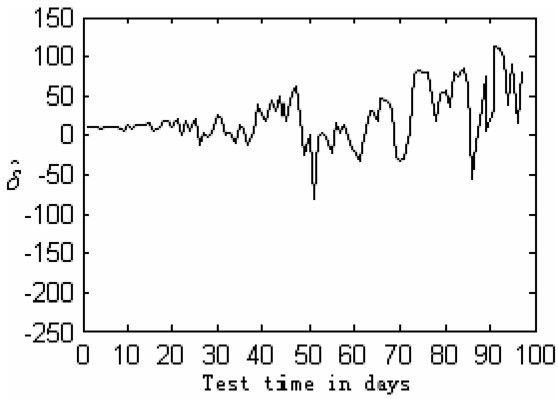

Fig.6 Estimated $\hat{\delta}$

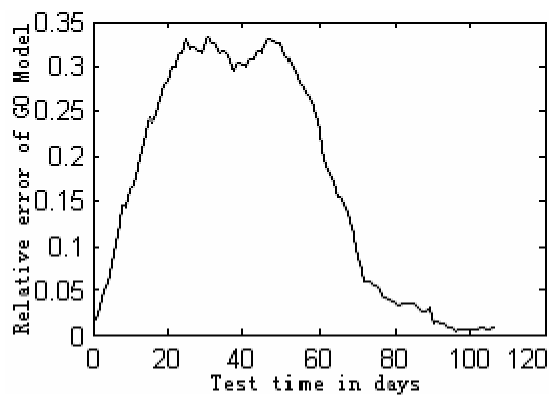

Fig.7 Relative error of GO Model

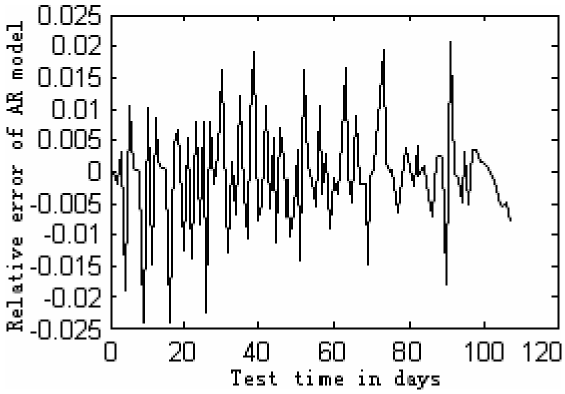

Fig.8 Relative error of AR model

Next, we calculate the sum of square errors (SSE) between the observed data and the estimated data to evaluate the predictive validity of each model. SSE is defined as follow,

$$SSE = \sum_{i=1}^{n} (\hat{M}(p+i) - M(p+i))^2, \quad p = 97, \quad n = 10.$$

Table 4 SSE of GO Model and AR Model

GO Model	109.5439
AR Model	44.2203

We can find that in Table 4 the result of SSE of Goel-Okumoto model is greater than that of AR model. It is clear that AR model is superior in predictive validity.

5 Conclusion

We select the famous NHPP model--Goel-Okumoto SRGM as reference and transform it to AR time series model with independent increment based on time series analysis method. Time series AR model gives simple methods to fitting different testing data. Numerical simulations show that the estimation and prediction of AR model based on time series are more effective and accurate compared to Goel-Okumoto model. The results of estimation and prediction ability of the proposed model are superior to to that of Goel-Okumoto model. Software reliability time series modeling provide a new method to evaluate and predict software reliability.

References:

[1] Takamasa Nara, Masahiro Nakata, Akihiro Ooishi, Software Reliability Growth Analysis-Application of NHPP Models and Its Evaluation, *IEEE Transactions on Reliability*, HASE 1996: 222-227.

[2] Zhang, X.; Pham, H., An analysis of factors affecting software reliability, *Journal of Systems and Software* 2000 , 43-56.

[3] J.Musa, A.Jannino, and K.Okumoto, *Software Reliability- Measurement, Prediction, Application*, McGraw Hill, New York,1987.

[4] Liang Yin and Kishor S. Trivedi, Confidence Interval Estimation of NHPP-Based Software Reliability Models, *Proceeding of the Tenth International Symposium on Software Reliability Engineering*, Boca Raton, Florida, Nov. 1999, PP. 6-11.

[5] Wu HP., Technology Analysis of Software Reliability model, *Modern Defense Technology*, 1998, 26(2): 32-41.

[6] M. Xie, G.Y. Hong, C. Wohlin, Software reliability prediction incorporating information from a similar project, *The Journal of Systems and Software*, 49(1999), 43-48.

[7] Stephen HKan, *Metrics and Models in Software Quality Engineering* Second Edition, Publishing House of Electronics Industry, 2004.

[8] Wang ZL., *Time series Analysis,* Publishing House of Chinese Statistics, 2000.

[9] Amerit L Goel., Software reliability models: Assumptions,limitations and applicability, *IEEE transaction on software engineering*, 1985, SE11 (12): 1411~1425.

[10] Huang Xizi, *Software Reliability、Security and Quality Assurance,* Publishing House of Electronics Industry, 2002.

[11] Dengzili, *Theory and Application of Optimal Filtering: Modern Time Series Analysis Method,* Publishing House of Harbin Institute of Technology, 2000.

[12] M. R. Lyu, Ed., *Handbook of Software Reliability Engineering*, McGraw Hill, 1996.