

ARIMA AR(1) 모형을 이용한 소프트웨어 미래 고장 시간 예측에 관한 연구

김희철* · 신현철**

요 약

소프트웨어 고장 시간은 테스트 시간과 관계없이 일정하거나, 단조 증가 혹은 단조 감소 추세를 가지고 있다. 이러한 소프트웨어 신뢰모형들을 분석하기 위한 자료척도로 자료에 대한 추세 검정이 개발되어 있다. 추세 분석에는 산술평균 검정과 라플라스 추세 검정 등이 있다. 추세분석들은 전체적인 자료의 개요의 정보만 제공한다. 본 논문에서는 고장시간을 측정하다가 시간 절단이 될 경우에 미래의 고장 시간 예측에 관하여 연구되었다. 고장 시간 예측에 사용된 고장 시간자료는 소프트웨어 고장 시간 분포에 널리 사용되는 와이블 분포에서 형상모수가 1이고 척도모수가 0.5를 가진 난수를 발생된 모의 자료를 이용 하였다. 이 자료를 이용하여 시계열 분석에 이용되는 ARIMA 모형 중에서 AR(1) 모형과 모의실험을 통한 예측 방법을 제안하였다. 이 방법에서 ARIMA 모형을 이용한 예측방법이 효율적임을 입증 하였다.

The Study for Software Future Forecasting Failure Time Using ARIMA AR(1)

Hee-Cheul Kim * · Hyun-Cheul Shin **

ABSTRACT

Software failure time presented in the literature exhibit either constant, monotonic increasing or monotonic decreasing. For data analysis of software reliability model, data scale tools of trend analysis are developed. The methods of trend analysis are arithmetic mean test and Laplace trend test. Trend analysis only offer information of outline content. In this paper, we discuss forecasting failure time case of failure time censoring. The used software failure time data for forecasting failure time is random number of Weibull distribution(shape parameter 1, scale parameter 0.5), Using this data, we are proposed to ARIMA(AR(1)) and simulation method for forecasting failure time. The practical ARIMA method is presented.

Key words : Arima (AR(1)) Method, Time Censoring, Forecasting Failure Time

* 남서울대학교 산업경영공학과

** 백석문화대학 컴퓨터정보학부

1. 서 론

소프트웨어 테스트 단계에서 소프트웨어 고장수(Number of failure)와 고장간격시간에 의해 소프트웨어 고장현상을 수리적으로 모형화하면 소프트웨어에 대한 평가를 보다 쉽게 할 수 있으며 신뢰도 모형에 의해 소프트웨어 고장수, 소프트웨어 고장발생간격시간, 소프트웨어 신뢰도 및 고장률 등의 신뢰성 평가측도들이 추정되어 미래의 고장시간을 예측할 수도 있다.

소프트웨어 고장시간은 수명자료가 된다. 따라서 비음(Nonnegative)의 값을 가지기 때문에 이 분야에서는 주로 지수분포, 와이블분포, 감마분포 등 일반화 감마 분포(혹은 어랑분포)가 많이 사용되어 왔다[1, 2, 12].

이 방면에서는 수명분포에 따른 모형의 효율성이 대다수 연구되었다. 이러한 신뢰모형들을 분석하기 위한 자료적으로 자료에 대한 추세 검정이 개발되어 있다. 추세 분석에는 산술평균 검정(Arithmetic mean test)과 라플라스 추세 검정(Laplace trend test) 등이 있다[3]. 산술평균 검정결과 고장시간이 증가함에 따라 산술 평균이 거의 증가 추세를 보이고 있으면 신뢰성장(Reliability growth) 속성을 가지고 있고 라플라스 추세 검정의 결과도 라플라스 요인(Factor)이 증가하다가 고장시간이 증가함에 따라 음수로서 감소하면 신뢰성장(Reliability growth) 속성을 나타내고 있다고 해석한다[3].

다른 측면으로 소프트웨어 고장시간에 대하여 품질관리 측면에서의 공정능력 지수를 제시하여 공정분석을 시도하기도 하였다[4]. 즉, 기존에 효율적 모형비교에 사용되는 실제자료를 이용하여 공정분석을 통하여 소프트웨어 고장 자료에 대한 공정 등급을 판정[5]하여 효율적 모형을 판단하는데 사용되기도 하였다

본 논문에서는 고장시간을 측정하다가 시간 절단이 될 경우에 미래의 고장시간 예측에 관하여 연구되었다. 고장시간 예측에 사용된 고장시간 자료

는 소프트웨어 고장시간 분포(수명분포)에 널리 사용되는 와이블 분포에서 형상모수가 1이고 척도모수가 0.5를 가진 난수를 발생된 모의 자료를 이용하였다. 이 자료를 이용하여 시계열 분석에 이용되는 ARIMA 모형 중에서 AR(1) 모형[6, 7, 8]과 모의실험[9]을 통한 예측 방법을 제안하였다. 이 방법에서 ARIMA 모형을 이용한 예측방법이 효율적임을 입증하는 것을 본 논문의 목적으로 한다.

본 논문의 제 2장에서는 관련연구로서 ARIMA 모형과 모의실험 방법을 서술하였고 제 3장에서는 모의 자료에 대한 고장시간 예측분석을 시도하였으며 제 4장에서는 결론을 나열하였다.

2. 관련연구

2.1 Arima 모형

ARIMA(Auto Regressive Integrated Moving Average 혹은 Box-Jenkins 모형)모형은 단기예측에 많이 이용되는 시계열 모형으로서 이러한 모형의 비정상 시계열 모형을 $ARIMA(p, d, q)$ 라고 표시하고 d 차 차분을 적용시켜 정상 시계열로 전환시킨 모형이 $ARMA(p, q)$ 라고 표시한다.

단, p, q 는 차수를 나타낸다. 이러한 $ARMA(p, q)$ 모형의 특수한 형태 $AR(1)$, $MA(1)$, $ARMA(1, 1)$ 가 존재한다. 이러한 특수모형들의 모형 선택은 일반적으로 ACF(Autocorrelation Function)와 PACF(Partial Autocorrelation Function)를 이용하여 수행한다[8].

본 논문에 사용되는 자료에 대한 PACF의 패턴이 (그림 1)에 나타내었다. 이 그림에서 자기상관(Partial Autocorrelation)함수의 패턴은 지수적으로 감소하는 패턴을 나타내고 있고 (그림 2)에서의 부분자기상관(Partial Autocorrelation Function)는 시차 1 이후에 절단되는 모습을 보이고 있어서 AR(1) 모형으로 판정할 수 있다[7].

2.2 모의 실험

본 논문에서의 모의실험은 평균(최근 고장시간)과 표준편차를 이용하여 다음과 같이 시행하였다.

$$\text{최근고장시간} + (\text{표준편차}) \times N(0, 1^2)\text{난수} \quad (2.1)$$

단, $N(0, 1^2)$ 은 평균이 0이고 분산이 1인 표준화 정규분포를 의미한다.

식 (2.1)을 1000번 반복하여 평균을 그 추정값으로 간주(미래고정시간 추정 1시점)하고 다시 이 추정값을 다시 식 (2.1)의 최근고장시간으로 대입하여 1000번 반복하여 그 평균을 추정값(미래 고장시간 추정 2시점)으로 간주하는 방법으로 모의 실험을 수행하였다. 물론 일반적인 방법은 최근고장시간 대신에 평균값을 선택하여 실험하는 경우 [9]가 일반적이거나 이 고장시간을 시점에 대한 시계열로 간주하여 평균고장시간 대신에 최근고장시간을 이용하였다.

3. 소프트웨어 미래 고장 시간 예측 분석

본 절에서는 소프트웨어 모형에 관한 고장시간 예측 분석을 위한 자료로서 와이블 분포에서 형상모수가 1이고 척도모수가 0.5를 가진 30개의 난수를 미니탭 소프트웨어(14버전)[10]에서 발생된 모

의 자료를 이용하였다. 이 자료에 대한 기초 통계량은 <표 1>에 요약하고 자료정보는 <표 2>에 요약되었다.

<표 2> 난수를 이용한 모의 고장 간격 및 고장 시간

시점(i)	Failure interval time	Failure time
1	0.10201	0.10201
2	1.17769	1.2797
3	0.66425	1.94395
4	0.31958	2.26353
5	0.57483	2.83836
6	0.19287	3.03123
7	1.06067	4.0919
8	3.22685	7.31875
9	0.36044	7.67919
10	0.37916	8.05835
11	0.12041	8.17876
12	0.64499	8.82375
13	0.14304	8.96679
14	0.00572	8.97251
15	0.45208	9.42459
16	0.33392	9.75851
17	0.21902	9.97753
18	0.81095	10.78848
19	0.00439	10.79287
20	0.17146	10.96433
21	0.51644	11.48077
22	0.16121	11.64198
23	0.04724	11.68922
24	0.65704	12.34626
25	0.65299	12.99925
26	0.27384	13.27309
27	0.49578	13.76887
28	1.43003	15.1989
29	1.0137	16.2126
30	0.42861	16.64121

<표 1> 모의자료에 대한 기초통계량

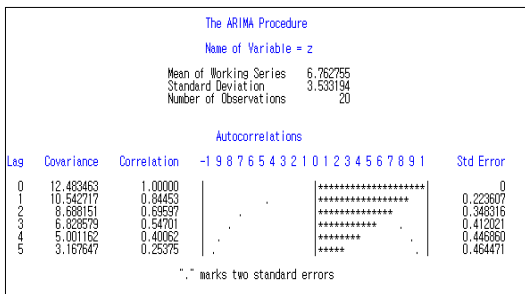
기초통계량	
표준 오차	0.821521679
중앙값	9.59155
표준 편차	4.499659551
분산	20.24693607
첨도	-0.571486237
왜도	-0.406972362
범위	16.5392
최소값	0.10201
최대값	16.64121
합	270.50724
관측수	30

의 고장 자료에서 20점 시점까지의 고장 자료는 미리 알고 있고 21시점부터 30시점까지 고장시간을 예측하는 방법과 그 결과를 비교하여 효율적 방법에 대하여 알아보하고자 한다.

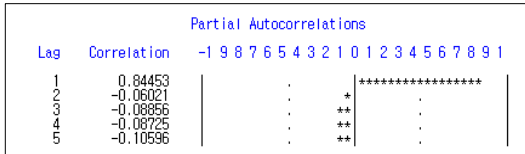
3.1 ARIMA(AR(1)) 모형을 이용한 고장 시간 예측

<표 2>에 나타난 고장시간 20시점까지 시계열 분석을 위해서 SAS(9.1버전)을 이용하였다. SAS 소프트웨어를 이용한 ACF(Autocorrelation Func-

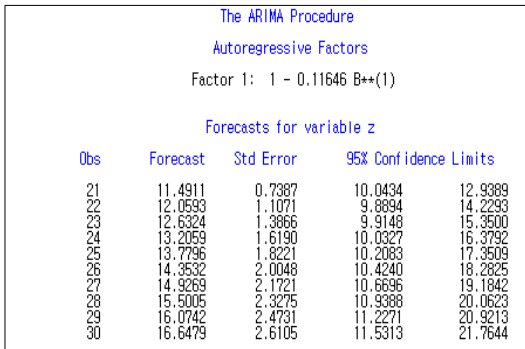
tion)와 PACF(Partial Autocorrelation Function)의 패턴은 각각 (그림 1)과 (그림 2)에 나타내었다. PACF의 패턴이 (그림 1)에 나타내었다. 이 그림에서 자기상관(Partial Autocorrelation) 함수의 패턴은 지수적으로 감소하는 패턴을 나타내고 있고 (그림 2)에서의 부분자기상관(Partial Autocorrelation Function)은 시차 1 이후에서 절단되는 모습을 보이고 있어서 AR(1) 모형으로 판정할 수 있다[8]. 따라서



(그림 1) SAS의 ACF(Autocorrelation Function) 패턴



(그림 2) SAS의 PACF(Partial Autocorrelation Function)



(그림 3) SAS의 예측 고장시간 결과물

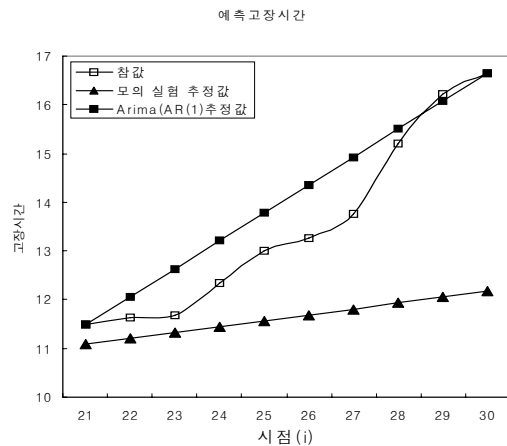
AR(1) 모형을 판정된다. 따라서 AR(1) 모형을 적용하여 21시점부터 30시점 까지 예측한 패턴이 (그림 3)에 나타내었다.

(그림 3)의 예측값에서 21시점은 약 11.49, 마지막 시점인 30시점은 약 16.65로 예측되었다.

3.2 모의 실험을 이용한 고장 시간 예측

<표 2>에 나타낸 고장시간 20시점 까지 시계열 분석을 위해서 SAS(9.1버전) 소프트웨어를 이용하였다. 본 논문에서의 모의실험은 평균(최근 고장시간)과 표준편차를 이용하여 시행하였다.

즉, 앞 절의 식 (2.1)에서 표준편차는 <표 1>의 기초통계량을 요약한 표에서 나타난 4.4997을 사용하였다. 예를 들어 21시점을 예측하기 위해서는 <표 2>의 20시점의 고장 시간 10.9643을 최근 고장시간으로 간주하여 1000번 반복하여 평균을 계산하였더니 11.0846이 추정되었고 유사한 방법으로 22시점을 예측하기 위해서 최근 고장시간을 11.0846으로 하여 다시 1000번 수행하면 11.2051이 추정된다. 유사한 방법으로 예측하면 23시점에서 30시점까지 각각 11.3257, 11.4463, 11.5668, 11.6874, 11.8080, 11.9286, 12.0491, 12.1697으로 예측값을 추



(그림 4) 참값과의 비교

정할 수 있다.

3.3 예측된 고장 시간의 비교

<표 2>의 고장 자료에서 20점 시점까지의 고장자료는 미리 알고 있다고 가정하고 21시점부터 30시점까지 고장시간을 예측하기 위하여 AR(1) 모형을 적용하고 또한 모의실험을 통하여 실제값(참값)과 비교하였다. 비교한 결과는 (그림 4)에 나타내었다. 이 그림에서 참값과의 비교에 있어서 모의실험을 이용한 예측보다는 AR(1) 모형을 적용하여 예측한 결과가 더 효율적임을 알 수 있다.

4. 결 론

소프트웨어 신뢰성은 개발의 최종단계에 있는 테스트 공정이나 실제 사용단계에 있어서 소프트웨어 내에 존재하는 고장 수나 고장 발생시간에 의해서 효과적으로 평가할 수 있는 상황으로 그 평가 기술이 중요하게 된다. 따라서 소프트웨어 개발의 테스트 공정이나 실제사용단계에 있어서 고장 발생 환경이나 고장 발생 현상을 수리적으로 모형화가 가능하면 평가를 할 수 있다. 테스트 시간이나 혹은 실행 시간, 발생된 고장 수와 고장 발생시간과의 관계를 효율적으로 관리함으로써 소프트웨어 신뢰도를 성장시킬 수 있다. 이러한 과정을 소프트웨어 성장 과정이라고 볼 수 있다.

신뢰모형들을 분석하기 위한 자료척도로 우선 자료에 대한 추세 검정이 개발되어 있다.

추세 분석에는 산술평균 검정(Arithmetic mean test)과 라플라스 추세 검정(Laplace trend test) 등이 있다.

그러나 이런 추세 분석들은 전체적인 자료의 개요의 정보만 제공한다.

본 논문에서는 고장시간을 측정하다가 시간 절단이 될 경우에 미래의 고장시간 예측에 관하여 연

구되었다. 이러한 고장시간 예측을 통해서 소프트웨어 제조공정이 소프트웨어 제품의 설계과정에서 계획한 대로 얼마나 균일한 제품을 생산할 수 있는지를 반영하는 할 수 있다. 만일 어느 제조공정에서 생산되는 제품의 품질변동에 이상이 있음에도 불구하고 어떠한 조치나 개선 없이 그 공정을 계속해서 가동시킨다면 그 제품의 품질은 크게 떨어질 것이다. 따라서 본 논문은 소프트웨어 시스템 설계와 관련 업종에 종사자들로 하여금 중도 전달된 자료를 예측하여 분석할 수 있으므로 중도 전달된 자료를 예측방법을 사전 정보로 활용하면 소프트웨어 품질의 개선에 도움이 되고 수명분포를 활용한 모형 선택에 있어서도 하나의 척도로 사용 할 수 있다.

향후 이러한 사전 정보 공정 분포를 이용한 베이지안적 접근 방법이나 수리적인 추정과 검정 부분에 대한 수리적인 접근이 기대된다.

참 고 문 헌

- [1] 김희철, 최유순, 박종구, “어랑분포를 이용한 NHPP 소프트웨어 신뢰성장 모형에 관한 연구”, 한국해양정보통신학회논문지, 제10권, 제1호, pp. 7-14, 2006.
- [2] 김희철, “카이제곱 NHPP에 의한 소프트웨어 신뢰성 모형에 관한 연구”, 한국컴퓨터정보학회논문지, 제11권, 제1호, pp. 45-53, 2006.
- [3] S. S. Gokhale and K. S. Trivedi, “A time/structure based software reliability model”, Annals of Software Engineering. Vol. 8, pp. 85-121, 1999.
- [4] 김희철, 신현철, “소프트웨어 고장간격시간에 대한 공정능력분석에 관한 연구”, 정보·보안 논문지, 제7권, 제2호, 2007.
- [5] 김광수, “품질경영”, 한울출판사, pp. 192-196, 2005.

- [6] 김태영 외 2, “사회과학논문작성과 통계자료분석”, pp. 187-194, 대영문화사, 2003.
- [7] 김해경, 이명숙, “시계열 분석”, pp. 67-97, 경문사, 2005.
- [8] 박유성, 허명희, “시계열 자료분석”, 제3장, 제4장, 자유아카데미, 1998.
- [9] 홍종선, “SAS와 통계자료분석”, pp. 590-605, 탐진, 1997.
- [10] 이레테크 미니텡 사업부, “Minitab 실무완성”, pp. 7.1-8.8, 이레테크, 2001.
- [11] J. F. Lawless, “Statistical Models and Methods for Lifetime Data”, John Wiley & Sons, New York, 1981.
- [12] H. Pham and L. Nordmann and X. Zhang, “A General Imperfect-Software -Debugging Model with S-Shaped Fault-Detection Rate”,

IEEE Trans. on reliability, Vol. 48, No. 2, pp. 169-175, 1999.

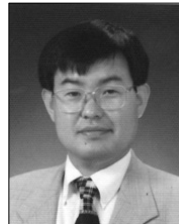


김희철

1992년 동국대학교 통계학과
(이학석사)

1998년 동국대학교 통계학과
(이학박사)

2005년~현재 남서울대학교 산
업경영공학과 전임강사



신현철

2002년 원광대학교 컴퓨터공학
과(공학박사)

1994년~현재 백석문화대학 컴
퓨터정보학부 부교수

2004년 (주)아이비루션 자문위원