GOMPERTZ BASED SPRT: MLE

[1] Kavita, [2]Dr. Pardeep Saini
[1]Research Scholar, [2]Supervisor
[1-2] Department of Computer Science & IT, Sunrise University, Alwar, Rajasthan

ABSTRACT

Sequential analysis of statistical science could be adopted to decide the reliability / unreliability of the developed software very quickly. The procedure adopted is the sequential likelihood ratio test (SPRT). It is designed for continuous monitoring and is an ongoing statistical analysis performed repeatedly during data collection. It is used in anomaly detection and decision making for electronics, structures and process control. The probability-based SPRT proposed by Wald is very general and can be used either for many different probability distributions. The data is repeatedly re-evaluated and a decision is made:

1. Reject the null hypothesis and stop collecting data
2. Fail to reject the null hypothesis and stop data collection
3. Continue to collect data until a decision about the null hypothesis can be reached

(Figure: Sequential Probability Ratio Test diagram)

The SPRT establishes thresholds that take the form of parallel lines, one representing the expected result and the other a significantly different result. When the value of calculated test statistic falls outside these thresholds, a conclusion can be made and data collection is stopped. The parameters are estimated using the Maximum Likelihood Estimation method. The Gompertz model is applied to five sets of existing software reliability data and the results are analyzed.

INTRODUCTION

The Wald procedure is particularly relevant when data are collected sequentially. Sequential analysis differs from classical hypothesis testing, where the number of cases to be tested or collected is fixed at the beginning of the experiment. In classical hypothesis testing, data collection is done without analysis and consideration of the data. After collecting all the data, analysis is done and conclusions are drawn.

In sequential tests however, each case is analyzed directly after collection, the data collected up to that point are then compared to certain threshold values incorporating new information obtained from the freshly collected case. This approach allows for inferences to be made during data collection, and a final conclusion can possibly be reached at a much earlier stage.

Because data collection can be stopped after fewer cases and decisions made earlier, the savings in human life and suffering and financial savings can be significant.

When analyzing software failure data, we are often concerned with either the time between failures or the number of failures in a given time interval. If we further assume that the average number of recorded faults in a given time interval is directly proportional to the length of the interval and the random number of occurrences of faults in the interval is explained by a Poisson process, then we know that the probability equation stochastic process representing the occurrences of faults is given by a homogeneous Poisson process.

WALD'S SEQUENTIAL TEST FOR THE POISSON PROCESS

A big advantage of sequential tests is that they require fewer observations (time) on average than tests with a fixed sample size. SPRTs are widely used for statistical quality control in manufacturing processes.

The SPRT for homogeneous Poisson processes is described below.

Let $N(t)$be a homogeneous Poisson process which counts the number of faults up to time T. Suppose we are testing a system (for example software system where faults are detected over time). We want to estimate the failure rate.

We cannot expect to estimate λ exactly; however we want to reject the system with high probability if our data indicates that the failure rate is greater than λ_1 and accept with high probability if it is less than λ_0.

As always with statistical tests there is some risk of getting incorrect answers. We need to enter two small numbers α and β where:

- α is the manufacturer's risk
- β is the consumer's risk

With specific choices of λ_0 and λ_1 such that $0 < \lambda_0 < \lambda_1$, the probability of finding N(t) failures in time span (0,t) with λ_0 as the failure rate is given by

$$Q_0 = \frac{e^{-\lambda_0 t}(\lambda_0 t)^N}{N!}$$

$$Q_1 = \frac{e^{-\lambda_1 t}(\lambda_1 t)^N}{N!}$$

The ratio

$$\frac{Q_1}{Q_0}$$

at any time T is considered to be a measure of deciding the truth with respect to $\lambda_0$or λ_1.

The SPRT decision rule is to decide in favour of $\lambda_0$or $\lambda_1$by tracking the number of failures at time later than T whenever

$$\frac{Q_1}{Q_0} \geq A$$

$$\frac{Q_1}{Q_0} \leq B$$

$$B < \frac{Q_1}{Q_0} < A$$

We continue testing.

Where

$$A = \frac{1-\beta}{\alpha}$$

$$B = \frac{\beta}{1-\alpha}$$

SEQUENCE TEST FOR SRGMS

For the Poisson process, the expected value $N(t) = \lambda t$is called the average number of failures recorded at time T. This is also called the mean value function of the Poisson process.

If we consider the mean value function to be a Poisson process with general function $m(t)$, the probability equation becomes

$$P[N(t) = y] = \frac{[m(t)]^y}{y!} e^{-m(t)}$$

Depending on the forms of $m(t)$, we get different Poisson processes called NHPP.

ANALYSIS OF LIVE DATA FILES

Table 1

Estimates of a, b, c& Specification values $b_0, b_1$for Time Domain

Data Set	No. of Samples	a	b	c	b_0	b_1
NTE	30	30.526286	0.055202	0.500320	0.0447802	0.067202
NTDS	26	26.613869	0.013832	0.125836	0.001322	0.026422
IBM	15	16.419633	0.045773	0.303497	0.013273	0.058273
AT&T	22	22.734515	0.063920	0.521470	0.051420	0.076420
SONATA	30	37.225335	0.033200	0.860121	0.020700	0.045700
LIVE	24	25.201579	0.034256	0.002983	0.021756	0.046756

Table 2

SPRT Analysis for Data Sets of Time Domain Data

t	N(t)	Acceptance Region	Rejection Region	Decision
0.3002	1	-2.826703	59.7372	
0.3146	2	-2.206270	52.4940	
0.5395	3	1.350998	36.1770	
0.5529	4	1.386293	15.1557	
0.5872	5	1.435542	33.4207	
0.7192	6	1.279444	27.3941	
0.7707	7	1.134477	25.5041	
0.8509	8	1.011414	24.1273	
1.019	9	0.262405	18.6939	Reject
1.1487	10	-0.176396	16.1740	Reject
1.1534	11	-0.191377	16.0924	Reject
1.2167	12	-0.382987	15.0663	Reject

(Similar calculations are presented in the paper for datasets NTDS, AT&T, IBM, LIVE and SONATA across the next pages.)

The developed SPRT methodology is for software failure data of the form $N(t)$where $N(t)$is the number of failures of the software system in units of time.

In this section, we evaluate decisions based on the estimated mean value function for five different datasets borrowed from previous studies. Parameter estimation is done using Maximum Likelihood Estimation.

Table 1 presents the estimates of parameters $a, b, c$and specification values b_0, b_1.

(Table 1 shows parameter estimates for datasets such as NTE, NTDS, IBM, AT&T, SONATA and LIVE — including number of samples, parameters a, b, c, and values $b_0$and b_1.)

Using these values, decision rules are calculated sequentially for each dataset.

Table 2 shows the sequential SPRT analysis for time domain data.

(Tables on pages 133–134 show sequential calculations for datasets Xie, NTDS, AT&T, IBM, LIVE and SONATA with acceptance and rejection regions and final decisions.)

CONCLUSION

The consolidated table of sequential likelihood ratio tests with Gompertz as an example for six data sets shows that the model performs well in decision making.

The model gave:

- Reject decision for 3 datasets (Xie, AT&T and IBM)
- Accept decision for 1 dataset (LIVE)
- Continue decision for datasets NTDS and SONATA

Thus we conclude that by applying SPRT to software data files, we can soon schedule the reliability or unreliability of the software.

REFERENCES

1. Frances, P.H. (1994). *Fitting a Gompertz Curve*. Journal of the Operational Research Society.
2. Garg, M.L., Rao, B.R., Redmond, C.K. (1970). Maximum Likelihood Estimation of the Parameters of the Gompertz Survival Function.
3. Gompertz, B. (1825). On the Nature of the Function Expressive of the Law of Human Mortality.
4. Gordon, N.E. (1990). Maximum Likelihood Estimation for Mixtures of Two Gompertz Distributions.

5. Johnson, N.L., Kotz, S., Balakrishnan, N. (1994). Continuous Univariate Distributions.
6. Kececioglu, D. (1991). Reliability Engineering Handbook.
7. Lyu, M.R. (1996). Handbook of Software Reliability Engineering.
8. Makany, R. (1991). Theoretical Basis of Gompertz Curve.
9. Pham, H. (2006). System Software Reliability.
10. Read, C.R. (1983). Gompertz Distribution.
11. Saleh, H.A. (1997). Statistical Quality Control: How to Detect Unreliable Software Components.
12. Wald, A. (1947). Sequential Analysis.
13. Wu, J.W., Lee, W.C. (1999). Characterization of the Structure of Gompertz Distributions.
14. Xie, M., Goh, T.N., Ranjan, P. (2002). Some Effective Control Chart Procedures for Reliability Monitoring.