Pro Mathematica Vol. XI, Nos. 21-22, 1997

MODELOS DE CONFIABILIDAD DE SOFTWARE: COMPARACION DE ENFOQUES DE ESTIMACION

Miguel Angel ***Díaz Martínez***
Luis Martín ***Aromí***
Domingo ***Galán Martínez***

Resumen

Se revisan diversos modelos de confiabilidad de software reportados en la literatura con los objetivos de:

1. Estudiar el comportamiento de la estimación máximo verosímil y bayesiana de los parámetros.

2. Dar a conocer estas técnicas con vistas a su generalización y futura aplicación en la práctica.

✏ *Departamento de Matemática General, I.S.P.J.A.E., Cuba.*

Introducción

El comportamiento de los fallos del software es determinista o estocástico? El Software ocasionalmente falla porque contiene defectos de diseño. Algunos han planteado que los fallos son sistemáticos, esto es, porque escribir un software es un ejercicio puramente lógico, no hay nada intrínsecamente incierto en esto. Si las entradas son suficientemente conocidas, el comportamiento del programa será completamente determinista.

Nosotros creemos que para describir la naturaleza de los fallos de software se requiere de un tratamiento probabilístico, así como usamos la estadística para describir cuán frecuente, en promedio, fallan componentes eléctricos o mecánicos. Esto se plantea en (1 y 2).

Para ver por qué, consideremos todas las posibles entradas, llamado espacio de entradas, que el software pudiera encontrarse en su vida. Una entrada para una operación del software es un conjunto de dígitos (números) leídos del mundo exterior y de información almacenada ya en la memoria de la computadora.

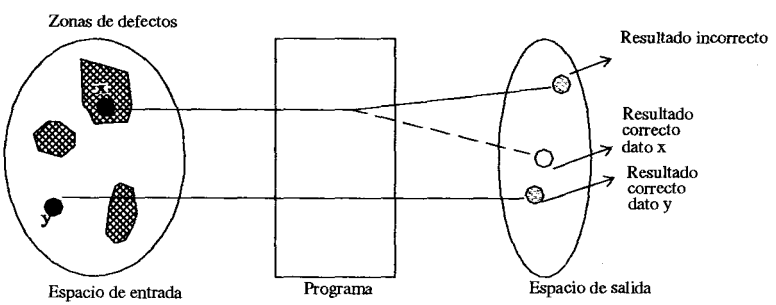

Figura 1

En la figura 1, se muestra el espacio de entradas en dos dimensiones, pero en la práctica el espacio pudiera usualmente ser multidimensional. En el espacio de entrada, las zonas de defectos están sombreadas. Aquí el dato x en la zona de defecto pudiera causar un resultado incorrecto. Por otro lado, el programa puede ser ejectuado exitosamente con el dato y, el cual no está en ninguna zona de fallos. En la fase de prueba del software para conocer si éste fallará o no, necesitaríamos billones de billones de años para mostrar todas las posibles entradas. De aquí la necesidad de inferir probabilidades de fallos a partir de una muestra de entradas. Pudiéramos desear conocer cuándo el programa fallará, pero esto no es posible debido a la incertidumbre inherente en el proceso.

Primero. La elevada incertidumbre del mecanismo físico que determina la sucesión de entradas, llamada trayectoria en el espacio de entradas. Nosotros nunca estaremos de cuál entrada será seleccionada en el futuro y diferentes entradas serán seleccionadas teniendo diferentes oportunidades de ser escogidas.

Segundo. Nosotros estamos inseguros del tamaño y localización de la región de defectos en el espacio de entradas. Aún si conociéramos la trayectoria, todavía no podremos conocer cuando el programa fallará.

Tercero. Aún cuando la estructura del programa sea un ejercicio puramente lógico, éste no está libre de los errores del programador.

Por estas razones debemos tratar el comportamiento futuro de los fallos de un programa en términos de probabilidades.

Algunos conceptos generales y notación

Aquí solamente revisaremos algunos conceptos y notaciones frecuentemente usados. Estamos interesados en el tiempo (T) hasta el próximo fallo y la distribución de probabilidad $F(t)$ de esta variable aleatoria, llamada distribución de vida, la que satisface la siguiente condición

$$F(t) = 0 \text{ para toda } t < 0 \quad \text{y} \quad F(t) = P(T \leq t) \text{ para } t \geq 0.$$

Como se define en (3), esto es la probabilidad de que el software se mantenga funcionando sin fallos hasta el instante t.

O sea, denotaremos por T el intervalo del tiempo libre de fallo aleatorio del software y sea $F(t)$ la función de distribución acumulativa de T. Entonces la confiabilidad está dada por $R(t) = 1\text{-}F(t)$; $t \geq 0$, que es la probabilidad de que el software esté funcionando sin fallos bajo condiciones ambientales dadas durante el intervalo $[0,t)$.

La razón de fallos o intensidad de fallos, $r(t)$ de $F(t)$ está definida como

$$r(t) = \frac{f(t)}{R(t)} = \frac{F'(t)}{R(t)}$$

donde $f(t)$ es la función de densidad de $F(t)$. A veces $r(t)$ es llamado ROCOF (Rate Of oCcurrence Of Failure: razón de ocurrencia de fallos) cuando hablamos sobre sistemas reparables.

La función razón de fallos es interesante desde el punto de vista de la confiabilidad, ya que la misma está conectada fuertemente con las propiedades de envejecimiento de la distribución de vida. Muchas clases de distribuciones de vida han sido definidas y estudiadas basadas en la forma de esta función razón de fallo, tal como distribuciones de RFC (Razón de fallo creciente) y distribuciones de RFD (Razón de fallo decreciente). La medida característica más importante de confiabilidad es la esperanza del tiempo hasta el fallo, definido como

$$ET = \int_0^\infty t f(t) dt = \int_0^\infty R(t) dt.$$

Dada la función de confiabilidad, esta es una medida del tiempo medio hasta el fallo para un componente con distribución de vida $F(t)$.

Modelo Jelinski-Moranda (JM)

Este es uno de los primeros modelos, el cual fue desarrollado por Kelinski y Moranda, ver (1) para más detalles. Este influyó fuertemente sobre muchos modelos posteriores, los cuales son modificaciones de este modelo. El modelo JM está basado en un proceso de Markov. Sus supuestos son:

1. El número inicial de defectos de software (N_0) es una constante desconocida.

2. Un defecto detectado es inmediatamente eliminado y no se introduce uno nuevo.
3. Los tiempos entre fallos (Ti) son independientes, cantidades aleatorias distribuidas exponencialmente;
4. Todos los defectos de software contribuyen con el mismo aumento a la intensidad de fallos del software.

Por las suposiciones 3 y 4 la intensidad de fallo inicial es entonces igual a $N_0\phi$ donde ϕ es una constante de proporcionalidad que denota la intensidad de fallo aportada por cada defecto. Tenemos de la suposición 2 que, después que un defecto nuevo es detectado y eliminado, el número de defectos restantes decrece en uno.

Denotemos por Ti, i=1,2,3...,N_0 el tiempo entre el (i-1)-ésimo y el i-ésimo fallo; Ti es así el i-ésimo intervalo de tiempo libre de fallos y su distribución es

$$F(t) = 1\text{-}e^{-\lambda(i)\,\xi} = P(T_i < t_i) \qquad (1)$$

donde

$$\lambda(I) = \phi[N_0 - (i\text{-}1)] \quad i=1,2,3,...,N_0 . \qquad (2)$$

La principal propiedad del modelo JM es que la intensidad de fallo es constante entre la detección de dos fallos consecutivos. Pero una crítica seria del modelo es que no todos los defectos de software son del mismo tamaño. Algunos defectos son más fácilmente detectados que otros. Teniendo en cuenta esto son presentadas algunas generalizaciones y modificaciones del modelo en (1).

Es razonable asumir que los primeros fallos son causados por defectos que tienen una alta probabilidad de ser detectados. Una generalización directa del modelo JM es usar una función tipo potencial para $\lambda(i)$ dada por:

$$\lambda(i) = \phi[N_0 - (i\text{-}1)]^{\alpha} \qquad i=1,2,3,...,N_0 . \qquad (3)$$

Debido a razones discutidas previamente $\lambda(i)$ debe decrecer rápido al principio y más lentamente después para cada i. De aquí que es razonable asumir que $\lambda(i)$ es una función convexa de i y α es probablemente mayor que uno, ya que en este caso, el decrecimiento de la intensidad de fallos es grande al inicio, ver figura 2.

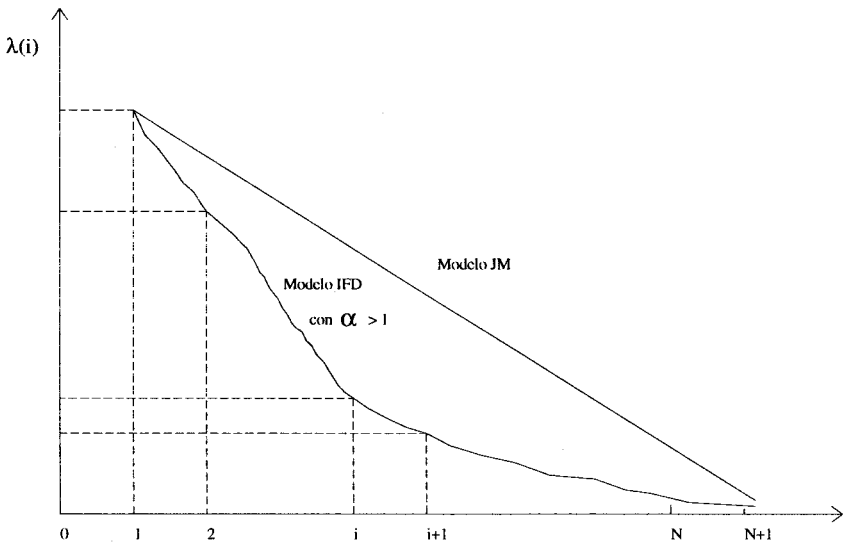

Figura 2

El modelo IFD (Intensidad de fallo decreciente) Markov exponencial supone que la intensidad de fallos es una función exponencial del número de defectos restantes. Está caracterizada por la función intensidad de fallos

$$\lambda(i) = \phi[e^{-\beta[N_0 - i+1]} - 1]; \qquad i=1,2,3,...,N_0 . \qquad (4)$$

Para el modelo con intensidad de fallo decreciente tipo exponencial, el decrecimiento de la función intensidad de fallos es mayor en la fase inicial que en la fase final.

Los parámetros del modelo IFD pueden ser estimados por el método de máxima verosimilitud. El número de defectos detectados se denota aquí por n (tamaño de la muestra). Supongamos que se conoce el conjunto de datos sobre los fallos $\{t1,t2,...tn\}$. Los parámetros ϕ y N_0 del modelo, pueden estimarse mediante la maximización de la función de verosimilitud. Luego tenemos para el modelo $\lambda(i)$, de manera análoga a (2)

$$\phi = n / \{\Sigma(N_0 - i+1)ti\} \qquad (5)$$

y el estimado de N_0 puede ser obtenido resolviendo la ecuación

$$\sum_{i=1}^{n} \frac{1}{N_0 - i + 1} = \frac{n \sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} (N_0 - i + 1)} \qquad (6)$$

Sustituyendo el N_0 estimado en la expresión de ϕ, podemos obtener la estimación máximo verosímil de ϕ. La función de verolimilitud posee, por lo general, un único máximo para un N_0 finito y un ϕ positivo, si y solo si, se satisface la siguiente desigualdad, ver (4).

$$\frac{\sum_{i=1}^{n} (i-1) t_i}{\sum_{i=1}^{n} (i-1)} > \frac{\sum_{i=1}^{n} t_i}{n} \qquad (7)$$

Realmente existen aquí varios problemas. De hecho, el estimado máximo verosímil de N_0 puede no existir para algún conjunto de datos y cuando existe, con frecuencia aparecen resultados ilógicos, tales como un número cercano a infinito de defectos, o que no hay más defectos remanentes. Otro problema es la intestabilidad de los resultados, lo que hace difícil confiar en ellos. una solución a estos problemas está en el uso de los métodos de la inferencia bayesiana.

Formulación Bayesiana del modeio JM

En el modelo bayesiano de Langberg-Sinpurwalla, presentado por estos autores en 1985. Los parámetros en el modelo JM son tratados como variables aleatorias. Sea Φ la intensidad de fallos por defecto y N el número de defectos iniciales en el software, ambos considerados vairables aleatorias. Dados Φ y N, los tiempos entre fallos están distribuidos exponencialmente con parámetros Φ y N, esto es

$$P(T_i > t_i / N, \phi) = e^{-\phi (N-i+1) t_i}, \; i \geq 1. \qquad (8)$$

Debido a la naturaleza discreta de N, la distribución a priori de N puede ser cualquier distribución discreta dada por $\pi k = P(N=k)$, $k=0,1,2,3,...$; y se asume que Φ tiene una distribución a priori Gamma con parámetros a y b, es decir

$$P(\phi) = \frac{a^b \phi^{b-1} e^{-a\phi}}{\Gamma(b)}; \quad \phi \geq 0. \tag{9}$$

Aquí la distribución posterior conjunta de N y Φ, para el conjunto de datos $(t1, t2,..., tn)$, es la siguiente

$$P(N = k, \phi = \varnothing) = \frac{\phi^{b+n-1} e^{-\phi(a+T_{n,k})}}{c} \pi_k, \quad k \geq n \tag{10}$$

siendo c la constante de normalización dada por

$$c = \Gamma(b+n) \sum_{j=n}^{\infty} \frac{j!}{(j-n)!} (a+T_{n,j})^{-b-n} \pi_j \tag{11}$$

y $T_{n,k}$ es el tiempo total sobre la prueba definido por

$$T_{n,k} = \sum_{i=1}^{n} (k-i+1) t_i, \quad k \geq n \tag{12}$$

La probabilidad posterior de Φ dado $N = k$ es también una distribución Gamma, Gamma(a',b'), con parámetros nuevos a' y b', dados por $a' = T_{n,k}$ y $b' = b+n$. La probabilidad marginal posterior de N es

$$P(N = k / \tilde{t}) = \frac{\frac{k!}{(n-k)!}(a+T_{n,k})^{-b-n} \pi_k}{\sum_{j=n}^{\infty} \frac{j!}{(j-n)!}(a+T_{n,j})^{-b-n} \pi_j}, \quad k \geq n. \tag{13}$$

Ya que N es una variable discreta (natural) el cálculo de esta distribución es directo y puede obtenerse de forma numérica mediante una truncación adecuada. El estimador bayesiano de N que minimiza la función de pérdida cuadrática puede entonces calcularse como sigue:

$$\hat{N} = \sum_{k=n}^{\infty} k\, P(N = k / \tilde{t}). \qquad (14)$$

Estudio de Simulación

Con el propósito de estudiar el comportamiento de los parámetros estimados del modelo de JM y compararlos con el enfoque bayesiano dado por Langberg y Sinpurwalla en 1985, consideremos un software con $N_0 = 50$ y $\phi = 10^9$. Mediante la transformación inversa de la distribución exponencial, es posible generar los tiempos *ti* entre fallos, y calcular la estimación máximo verosimil de N_0 y ϕ usando (5) y (6).

Para obtener la estimación bayesiana de los parámetros, consideramos que la distribución a priori de ϕ es Gamma con parámetros $a = 10^{11}$ y b=100 y la distribución a priori de N_0 viene dada por:

k	45	46	47	48	49	50	51	52	53	54	55
Π_k	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

Tabla 1. Distribución a priori de N_0.

Los valores de los parámetros de la distribución a priori fueron tomados de manera que garanticen que el valor esperado de ϕ esté tan cerca como sea posible del valor constante de ϕ dado en el modelo JM, pero con menor varianza. La misma idea es seguida para la distribución asumida de N_0.

Fueron generadas dos muestras de tamaños 22 y 30 y se calcularon las correspondientes estimaciones bayesianas y máximo verosimil. Los resultados se muestran en la tabla 2.

	Estimaciones			
tamaño	***máximo verosímil***		***Bayesiana***	
de muestra (n)	N_0	ϕ	N_0	ϕ
22	101	4.10^{-10}	49.844	$9.86\ 10^{-10}$
30	40	$6.95\ 10^{-11}$	50.568	$1.042\ 10^{-9}$

Tabla 2. Resultados de las estimaciones.

Conclusiones y Recomendaciones

Aunque este trabajo de investigación no brinda aún una respuesta definitiva, el número de muestras generadas no es suficiente, puede apreciarse que la estimación bayesiana del número de defectos iniciales N_0 está más cercano al valor asumido en las dos muestras.

Las estimación bayesiana de ϕ, sólo en la muestra de tamaño 30, es la más cercana al valor real de ϕ ; luego, se recomienda como conclusión general la repetición del proceso, incrementando el número de muestras generadas y cambiando sus tamaños. Así como variar las distribuciones a priori de los parámetros del modelo, esto es, de ϕ y de N_0 .

Referencias

[1] Xie, M. *Software Reliability Modelling*, Word Scientific, 1991.

[2] Littlewood. B. and Strigine, L. *The Risk of Software*. An article from Scientific American, Vol. 267, No.5, November 1992.

[3] Gnedenko, B. *Theory of probability*. Mir Publishers. Moscow, 1976.

[4] Littlewood, B. and Verrall, J.L *Likelihood Function of a Debugging. Model for Computer Software Reliability*. IEEE Transactions on Reliability, Vol. R-30, No.2, June 1981.

mgind@cujae.cu