1999年度日本オペレーションズ・リサーチ学会
春季研究発表会

ソフトウェア最適リリース問題に対する平滑化手法の適用について

土肥正† (01307065)，八並美文†，西尾泰彦† (02401815)，尾崎俊治† (01002265)
†広島大学工学部

1. はじめに

ソフトウェア製品のテスト工程を終了していつ運用段階に移行するべきかを決定する問題はソフトウェア最適リリース問題 [1, 2] と呼ばれており，ソフトウェア製品の開発管理において極めて重要な問題となっている．このような問題を定式化する際，ソフトウェア製品の信頼度成長現象を記述することが必要不可欠となる．従来までに報告されてきたソフトウェア最適リリース問題の解法では，ソフトウェア信頼度成長モデルと呼ばれる確率モデル (例えば [3]) を仮定することにより，解析的に最適なリリース時期を求めることがほとんどであった．しかしながら，このような解法の妥当性は仮定される信頼度成長モデルと観測データの性質に強く従属しており，常に最良の方法であるとはいえない．これに対して文献 [4] では，ニューラルネットワークを用いて将来におけるフォールト発見時刻を予測し，ノンパラメトリックに最適リリース時期を推定する方法を提案している．

しかしながら文献 [4] で提案された方法は，ソフトウェアのテスト工程において観測されたフォールト発見時刻のサンプルパスに基づいており，観測データ数が比較的少ない場合には高い精度で最適リリース時期を推定することが困難であった．そこで本稿では，フォールト発見時刻のサンプルパスに対してある種の平滑化技法を適用することにより，最適リリース時期をより高精度で推定することを目指す．具体的には，観測データならびに将来におけるフォールト発見時刻の予測値から，ソフトウェア故障率のノンパラメトリック推定量を導出する．この推定量はステップ関数で表現されるため，データ数が少ない場合にはかなり粗い関数となることに注意しなければならない．ここでは，この関数を平滑化する [5, 6] ことにより，より滑らかなソフトウェア故障率の推定値を求め，さらに総期待ソフトウェア費用を最小にする最適リリース時期を推定することを試みる．

2. ソフトウェア最適リリース問題

ソフトウェア製品をテスト段階から運用段階に移行させるために，経済性の観点からソフトウェアを出荷する時期を理論的に決定することは極めて重要な問題である．いま，時刻 $t=0$ でソフトウェアのテストを開始し，時刻 $t=t_0$ $(0 \leq t_0 \leq T_{LC})$ で製品をユーザに出荷するものとする．以下のような費用構造を仮定する．

c_1 (>0)：テスト工程において発見されるフォールト一個当たりの修正費用

c_2 $(>c_1)$：運用段階において発見されるフォールト一個当たりの修正費用

c_3 (>0)：ソフトウェアテストに要する単位時間当たりの費用．

ここで，T_{LC} (>0) はソフトウェア寿命であり，既知の定数とする．

時刻 t (≥ 0) において発見されたソフトウェアフォールト数 $\{N(t), t \geq 0\}$ は任意の計数過程であり，その期待値 $\mathrm{E}[N(t)] = M(t)$ は無限階微分可能とする．$\{N(t), t \geq 0\}$ が非定常ポアソン過程 (NHPP) ならば，関数 $M(t)$ は平均値関数と呼ばれ，特に $r(t) = dM(t)/dt$ は NHPP の強度関数もしくはソフトウェア故障率といわれる．このとき，総期待ソフトウェア費用は以下のように定式化される．

$$C(t_0) = c_1 M(t_0) + c_2\{M(T_{LC}) - M(t_0)\} + c_3 t_0. \tag{1}$$

よって，問題は総期待ソフトウェア費用を最小にする最適ソフトウェアリリース時刻 t_0^* を求めることである．すなわち，

$$\min_{0 \leq t_0 \leq T_{LC}} C(t_0). \tag{2}$$

関数 $C(t_0)$ が t_0 の狭義凸関数であるための必要かつ十分条件は，平均値関数 $M(t_0)$ が狭義凹関数であることである．これより，式 (2) で与えられる最小化問題に対する 1 階の最適性条件は，

$$r(t_0^*) = c_3/(c_2 - c_1) \tag{3}$$

を満足する $t_0^* \in [0, T_{LC}]$ を求めることである．

3. ソフトウェア故障率のノンパラメトリック

もしソフトウェア故障率の形状が既知であれば，式 (3) の非線形方程式を t_0^* に関して解けばよい．いま，時刻 T までに $N(T)$ (≥ 1) 個のフォールト発見時刻データ

$$0 < T_1 < T_2 < T_3 \cdots < T_{N(T)} < T$$

が得られているものとする．さらに，時刻 T から $N(T)+1$ 個目以降のフォールト発見時刻の推定値

$$T_{N(T)} < T < \hat{T}_{N(T)+1} < \cdots < \hat{T}_{N(T)+m}$$

が得られているものと仮定しよう．ここで，m は $T_{LC} \leq \hat{T}_{N(T)+m}$ を満足する十分大きい非負の整数であり，意志決定者によって予め設定されている定数とする．

上述のデータに対するソフトウェア故障率は完全単調 (complete monotonicity) である [5, 6] とする．すなわち，任意の時刻 $t \geq 0$ に対して

$$(-1)^n \frac{d^n r(t)}{dt^n} \geq 0, \quad (t \geq 0,\ n \geq 0) \tag{4}$$

である．いま，

$$\begin{aligned}(\tau_0, \tau_1, \cdots, \tau_{N(T)}, \quad &\cdots \quad, \tau_{N(T)+m}) \\ = \quad &(0, T_1, \cdots, T_{N(T)}, \cdots \hat{T}_{N(T)+m})\end{aligned}$$

と置き，二次元平面 $\mathcal{R}^2 \in [0,\infty) \times [0,\infty)$ 上に $(\tau_i, i),\ (i = 0, 1, 2, \cdots, N(T)+m)$ を打点し，各点を線分で結ぶことによって平均値関数 $M(t), (0 \leq t \leq \tau_{N(T)+m})$ のノンパラメトリック推定量

$$\hat{M}(t) = \begin{cases} i + (t-\tau_i)/(\tau_{i+1}-\tau_i), \\ (\tau_i \leq t \leq \tau_{i+1};\ i = 0, 1, \cdots, N(t)-1) \\ \\ n + 0.5(t-\tau_n)/(T-\tau_n), \\ (t_n \leq t \leq T) \end{cases} \tag{5}$$

を得る．

さらに，時間間隔 $[0, t]$, $(T \leq t \leq \tau_{N(T)+m})$ を $k\ (>1)$ 個に分割し，$\Delta s = t/k$ ならびに $s_i = i\Delta s\ (i = 0, 1, \cdots, k)$ を定義する．このとき，式 (5) に基づいたソフトウェア故障率のノンパラメトリック推定量を

$$\hat{r}(t) = \hat{r}_i, \ (s_{i-1} < t \leq s_i\ ; i = 1, 2, \cdots, k) \tag{6}$$

のように定義する．ここで，

$$\hat{r}_i = \{\hat{M}(s_i) - \hat{M}(s_{i-1})\}/\Delta s, \ (i = 1, 2, \cdots, k) \tag{7}$$

である．

いま，$\{\hat{r}_i;\ i = 1, 2, \ldots\}$ は完全単調数列であるので，

$$(-1)^j \Delta^j \hat{r}_i \geq 0, \ (j+1 \leq i;\ j = 0, 1, 2, \cdots) \tag{8}$$

であると仮定できる．ここで，Δ^j は j 番目の差分オペレータであり，

$$\begin{aligned}&\Delta^0 \hat{r}_i = \hat{r}_i, \\ &\Delta^1 \hat{r}_i = \hat{r}_i - \hat{r}_{i-1}, \\ &\Delta^j \hat{r}_i = \Delta^{j-1} \hat{r}_i - \Delta^{j-1} \hat{r}_{i-1}, \ (j > 1)\end{aligned} \tag{9}$$

である．

これより，ソフトウェア故障率の推定値

$$\mathbf{r} = (r_1^*, r_2^*, \cdots, r_{N(T)+m})$$

を求めるために，以下のような 2 次計画問題を任意の階数 $j\ (= 1, 2, \cdots)$ について解く．

$$\min_{\mathbf{r}} D(\mathbf{r}, \hat{\mathbf{r}}) = \sum_{i=1}^{k} w_i (r_i - \hat{r}_i)^2$$

$$\text{s.t. } (-1)^j \Delta^j r_i \geq 0,$$

$$(j+1 \leq i;\ \ j = 0, 1, 2, \cdots). \tag{10}$$

ここで，$\hat{\mathbf{r}} = (\hat{r}_1, \cdots, \hat{r}_{N(T)+m})$ であり，重み w_i は意志決定者によって任意に決定されるか，一様乱数によってランダムに設定される．

最終的に，式 (10) で求めたソフトウェア故障率の推定値 $\mathbf{r}$ を線分でつないだ区分的線形関数 $R(t)$ を用いて，$R(\hat{t_0}^*) = c_3/(c_2 - c_1)$ を満足するような最適リリース時刻 $\hat{t_0}^*$ $(< \tau_{N(T)+m})$ を求めればよい．時点 T から将来におけるフォールト発見時刻を予測するためには様々な統計的手法が考えられる．例えば，文献 [4] と同様にリカレント型ニューラルネットワークを適用したり，ソフトウェア信頼性評価において固有に開発された時系列モデルを適用することは有用であると考えられる．

4. まとめと今後の課題

本稿では，Miller and Sofer [5, 6] によって開発されたソフトウェア故障率のノンパラメトリック推定法をソフトウェア最適リリース問題に適用することを提案した．特に，最適リリース問題では将来におけるソフトウェアフォールト発見時刻を予測しなければならないため，ニューラルネットワークなどの時系列予測モデルを併用することが必要不可欠となる．

参考文献

[1] K. Okumoto and A. L. Goel, "Optimum release time for software system based on reliability and cost criteria", *J. Systems and Software*, vol. 1, no. 4, pp. 315-318 (1980).

[2] H. S. Koch and P. Kubat, "Optimal release time of computer software", *IEEE Software Eng.*, vol. SE-15, no. 3, pp. 323-327 (1983).

[3] 山田茂,「ソフトウェア信頼性モデル」, 日科技連, 東京 (1994).

[4] 土肥正, 西尾泰彦, 篠原康秀, 尾崎俊治, "ニューラルネットワークを用いたソフトウェア最適リリース問題の幾何学的解法", 電子情報通信学会論文誌 (A), vol. J81-A, no. 1, pp. 110-118 (1998).

[5] D. R. Miller and A. Sofer, "A non-parametric approach to sofrware reliability using complete monotonicity", *Software Reliability; State of the Art Report*, A. Bendell and P. Mellor (eds.), pp.185-195, Pergamon, Oxford (1986).

[6] D. R. Miller and A. Sofer, "A nonparametric software reliability growth model", *IEEE Trans. Reliab.*, vol. R-40, no. 3, pp. 329-337 (1991).