コピュラを適用した離散型２変量ソフトウェア信頼性モデル

坂上, 敏樹 / SAKAUE, Toshiki

(発行年 / Year)
2013-03-24
(学位授与年月日 / Date of Granted)
2013-03-24
(学位名 / Degree Name)
修士(工学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)

2012 年度　修士論文

コピュラを適用した
離散型２変量ソフトウェア信頼性モデル

A DISCRETE SOFTWARE RELIABILITY MODEL BASED ON COPULAS

法政大学大学院工学研究科

システム工学専攻修士課程

11R6209　　坂上 敏樹

Toshiki Sakaue

指導教員　木村 光宏

概要

本研究では，ソフトウェア開発におけるテスト実施日ごとに，発見フォールト数と消化テストケース数の 2 つの変量間の依存性をコピュラにより表現した離散周辺分布をもつモデルを構築し、その有効性について検証，考察する．ソフトウェア信頼性モデルは時系列データの発見フォールト数だけで構築されるモデルが主だが，例えば，消化テストケース数，テスト網羅度などもソフトウェアの信頼性品質に影響を与える要因だと直感的に感じられる．本研究では，非線形間の依存性を表現することができるコピュラをソフトウェア信頼性モデルに適用することで，発見フォールト数に加え，テストケース消化数を加えた 2 変量の離散型ソフトウェア信頼性モデルを構築し，より現実のソフトウェア開発環境に則した予測推定を提案する．また，これらを実測データによる信頼性評価例を示す．

Abstract

In this study, we propose a bivariate software reliability assessment model. The model has two variables which are the number of digested test cases and the number of detected software faults. On the modeling, we employ an FGM (Farlie-Gumbel-Morgenstern) copula which has two Poisson marginal distributions and a dependency parameter. By using our model, a real data set is analyzed and we show several software reliability assessment results as a numerical example. Also we introduce a change point factor into the model and discuss its results.

目 次

1　はじめに

1.1　研究背景と目的

現在のような高度情報化社会において，IT(情報技術）の役割はビジネスにとどまらず，個人の生活の場にも大きな役割を果たしている．近年では，人を中心に置いた IT 社会，ICT(Information and Communication Technology) とまで呼ばれ，スマートフォンやクラウドの登場や，今までの IT の領域を超えた農業や医療での IT 活用など，より IT と人との距離は近くなり，より豊かな社会を形成するための役割を果たしているといえる．それと同時に，製品，システムの不具合やトラブルが，多くの顧客やユーザーに損害を与えるだけでなく，場合によっては，企業の経営活動そのものにも大きなダメージを与えてしまうケースがある．近年の事例でいえば，自動車のシステム不具合によるリコール，ATM での取り扱いが行えなくなる大規模トラブルなどが目新しいだろう．これらのことから，ソフトウェアが大規模化や複雑化，多様化を実現することでより便利，生活や業務に必要になっていくと同時に，トラブルや不具合の発生というリスクをいかに抑えるかという企業側の考えから，ソフトウェアの信頼性がより求められている時代といえる．現在のソフトウェアの多くの開発はウォーターフォールモデルと呼ばれる図 1 のような，いくつかの工程に分け，順番に工程を実施していくことで，開発されている [1].

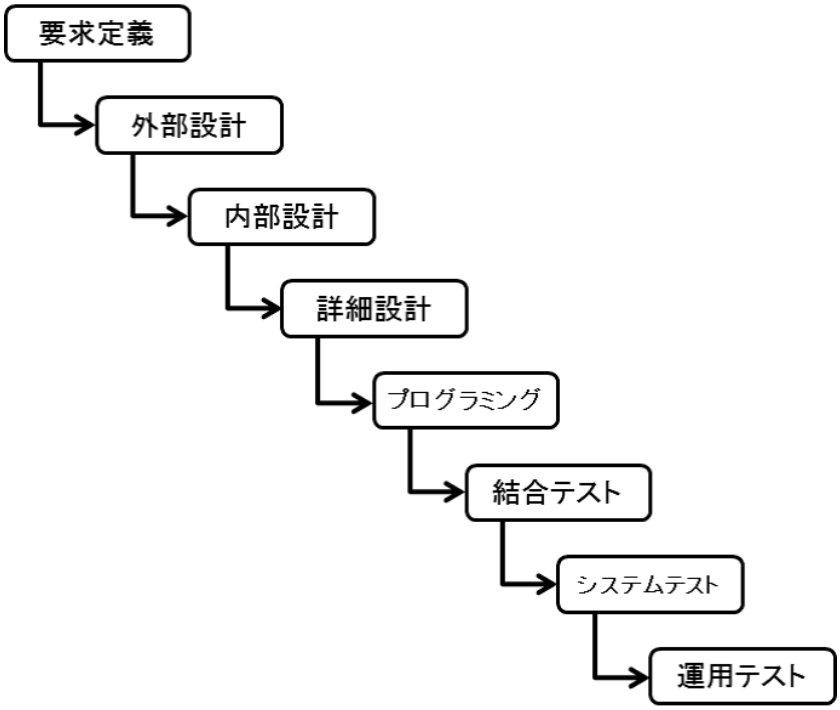

図 1: ウォーターフォールモデル.

各工程では以下の作業を行う.

- 要求定義
  開発するソフトウェアの役割や目的を定義する工程．解決すべき経営課題や，業務を分析することで業務課題を把握し，必要な機能や満たすべき性能を検討し明確にしていく．また，費用対効果分析や開発スケジュールを策定を行う.

- 外部設計
  利用者から見える部分の設計を行う工程．どのように利用者に対してソフトウェアがどのように振る舞うべきか考え，画面やユーザーインターフェースの構成，使用するデータベースの構造など機能面の設計を行う．

- 内部設計
  必要な機能をプログラムに分割し，処理構造を明確にする工程．詳細な機能を洗い出し，それぞれの内部構造について検討する．外部設計がインタフェースなどの入出力や機能面の設計に対し，内部設計ではそれを実現するための仕様を定めていく．

- 詳細設計
  各機能をより細分化しどのような処理を行い動作させるかを定める工程．プログラムの設計を行い，プログラミング工程においてスムーズに開発を行えるように設計する．

- プログラミング
  これまでの設計に基づきプログラミングしていく．チーム内でのプログラミングのための標準ルールを決め，個々の機能をモジュールと呼ばれるプログラム単位に分けプログラミングを行う．モジュール単体でのテスト（単体テスト）を実施後，各モジュールを組み合わせ一つのソフトウェアとして完成させる．

- 結合テスト
  モジュールを組み合わせて行うテスト．モジュール間の処理が正しく行えているかを確認する．

- システムテスト
  システム全体を対象としたテスト．実際の現場に近い環境で行われ，機能や要件を満たしているかどうかを確認する．

- 運用テスト
  実際の業務に即し，システムを利用してみて問題なく動作するか確認するテスト．応答時間や耐久性など業務遂行に支障がないかを確認する．

ウォーターフォールモデルはこのように明確に分けられた工程によりスケジュールの管理の容易さやいくつかのベンダー（協力企業）に別れて開発を行う場合も前工程の資料を基に確実に要件を組み込めるといったメリットがある．また，要求定義から外部設計を上流工程，内部設計以後を下流工程と呼び，上流工程での成果物を基に，下流工程で実装を行っていくために，上流工程での作りこみが品質や最終的なソフトウェアの出来に繋がっていくといえる．しかしながら，上流工程でいかに品質を考慮した設計をしたとしても，人が作成するがゆえに，ソフトウェアには多くの人為的誤りや欠陥，いわゆるフォールトが作り込まれ，のちに，ソフトウェアの障害や故障の発生をもたらしてしまう．したがって，フォールトを発見・修正・除去するために実施されるテスト工程は，ソフトウェアの品質・信頼性を高めるために最も重要な工程であるといえる．テスト工程ではあらかじめ仕様書から制御の流れや，機能要件をみたしているかどうかなど，それぞれの観点からあらかじめ入力データ（テストケース）を用意し，それらのプログラムパスに内在するフォールトを発見していく．しかしテストに時間をかければかけるほど品質は良くなるが，コストが増え，定められた納期に間に合わないことは企業にとって大きなジレンマといえる．また，特に大規模プロジェクトにおいては，開発管理者個人の能力によって管理することにも限界があり，適切な管理は不可能となってきている．このような状況下で，企業や管理者が品質，コスト，納期をそれぞれ確保しつつ，効率的に管理したいと

考えるのは当然である．そのために適切な進捗管理や品質評価のための定量化が必要である．正確な見積もり行うことで，ソフトウェア開発においてリソース，開発期間などの適切な配分を可能とすることが期待できる．このように，コスト・納期を考慮したうえでソフトウェアの性能や信頼性を高めることを目的に開発プロセスの進捗度を定量的に管理するために，ソフトウェア信頼度成長モデル (software reliability growth model，以下，SRGM と略す) がある [2].

SRGM は，確率統計に基づき，テスト期間中に発見されるフォールト数のデータから記述するモデルである．特に，フォールト発見数モデルは必要なデータがテスト経過時間のフォールト発見数だけであり，非常に低労力でデータを得られるため，実用的に広く利用されている．しかし，フォールトの発生がテスト時間のみに依存しているという前提を仮定に置き構築された SRGM は，フォールト発生現象そのものに影響を与えるであろうテスト労力やテスト網羅度については加味されておらず，実際には考慮すべき事柄である．この点に着目し，井上，山田ら [3][4] は，2 変量ワイブル型ソフトウェア信頼度成長モデルを考案し，テスト時間とフォールト発見数を変数として扱った．また，木村，山田らは [5][6]，あらかじめ用意されたテストケース K 個の消化されていく過程を死滅過程により表現し，発見されるフォールトの数とテスト終了時期を推定している．しかし，いずれのモデルでも，2 変量間の関連性については積極的には議論されては来なかった．

本研究では 2 つの変量間の関連性に着目し，離散周辺分布をもつコピュラによるモデル化について検討する．コピュラは非線形な依存関係を表現できる手法として，主に金融実務で活用されている [7]．コピュラを用いることにより，相互に依存した 2 つの周辺分布を同時分布として表現することが可能となるため，2 つの変量間の関連性や依存性に関して考慮したモデルを構築することができた．本モデルでは，時系列の 2 変量データである，発見フォールト数と消化テスト項目数を 2 つの周辺分布として定義し，コピュラを適用することで，例えば，テストケースを消化するほどフォールトの発見数が多くなるなどの事象を表現可能となった．また，実測データとの比較や依存性を考慮しない周辺分布との比較を通して，その有用性を検証し，最終的に本モデルがテスト工程の定量的進捗度管理とソフトウェア信頼性評価の精度向上に貢献できるかについて考察する．

1.2 論文構成

2 章にて本研究で扱うコピュラについて基本的な定義と，いくつか種類のあるコピュラの関数をそれぞれの特徴を踏まえ述べる．3 章にて，発見フォールト数，消化テストケース数を周辺分布として，コピュラを適用した 2 変量モデルを構築する．このモデルに対して，4 章では，実際のテスト開発で得られたデータを適用し，各パラメータの推定を行い，コピュラを適用したモデルと依存性を考慮しない周辺分布，実測データとの比較や管理図の作成などを行う．また，それらをグラフや表により結果を示す．それらに加え，第 i 日目までのデータを適用し，i 日目以降のテスト終了日時をモードにより推定を行い，その間の発見フォールト数の推定も行う．5 章では，結果を踏まえて考察を行い，本モデルの有効性を検証し，6 章にて結論と今後の課題について検討する．

2　コピュラについて

2.1　コピュラ

接合関数コピュラ（copula）とは多変量分布関数とそれらの 1 次元周辺分布関数に結合させ，関係を表せる関数である．相関を表す代表的な指標には相関係数があるが，一次の相関しかわからないため非線形な相関を表すことができない．コピュラは関数であることから，確率変数間の多様な依存関係を表すことができる．コピュラの基本的な定理としてスクラー（Sklar）の定理がある [8].
周辺分布関数 $F_1, \ldots, F_n$ を持つ連続な n 変量同時分布関数を F とすると，

$$\Pr(X_1 \leq x_1, \ldots, X_n \leq x_n) = F(x_1, \ldots, x_n) = C(F_1(x_1), \ldots, F_n(x_n)), \tag{1}$$

を満たす関数 C が一意に存在する．この C がコピュラと呼ばれる関数である．別の表現をとれば，コピュラ C に周辺分布関数 $F_1, \ldots, F_n$ から生成される $C(F_1(x_1), \ldots, F_n(x_n))$ は周辺分布を $[0, 1]$ 区間上で一様分布とする同時分布関数である．このことから，多次元分布をモデル化するときに周辺分布と確率変数間の従属構造を表すコピュラを別々に特定することが可能となる．

コピュラには複雑な依存関係を表現するために，様々な形でコピュラ関数の定式化が行われている．以下にいくつかの典型例を示す．ここで，u_1，u_2 は周辺分布を表しており，定義式は 2 変量の例を示している．

- 正規コピュラ

$$C(u_1, u_2) = \Phi_n((\Phi_1)^{-1}(u_1), (\Phi_1)^{-1}(u_2); \theta). \tag{2}$$

  $\Phi(\cdot)$ は n 変量正規分布関数である．θ は $-1 \leq \theta \leq 1$　を満たし，0 のとき 2 変量間には依存性がないことを表し，正で絶対値が大きいほど変量間の正の依存度が強いことを表し，負で絶対値が大きいほど負の依存度が強いことを表す.

- クレイトン・コピュラ

$$C(u_1, u_2) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}. \tag{3}$$

  θは $0 < \theta$ を満たし，0 のとき 2 変量間には依存性がないことであることを表し，正で絶対値が大きいほど正の依存度が強いことを表し，負で絶対値が大きいほど負の依存度が強いことを表す.

- ガンベル・コピュラ

$$C(u_1, u_2) = \exp -((-\ln u_1)^{\theta} + (-\ln u_2)^{\theta})^{1/\theta}. \tag{4}$$

  θは $1 \leq \theta$ を満たし，1 のとき 2 変量間には依存性がないことを表し，正で絶対値が大きいほど正の依存度が強いことを表すが，負の依存関係を表すことはできない.

- フランク・コピュラ

$$C(u_1, u_2) = -\frac{1}{\theta} \ln(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1}) \tag{5}$$

  θは$\theta \neq 0$ を満たし，0 のとき 2 変量間には依存性がないことを，正で絶対値が大きいほど正の依存度が強いことを表し，負で絶対値が大きいほど負の依存度が強いことを表す.

このようにいくつかの種類のコピュラの中からを選択し，モデル化することとなる.

2.2 FGM コピュラについて

本研究では，Farlie-Gumbel-Morgenstern コピュラ（以下，FGM コピュラと略す）を適用し，モデルを構築する．

$$C(u_1, u_2) = u_1 u_2 (1 + \theta(1 - u_1)(1 - u_2)) \tag{6}$$

FGM コピュラは変数間のわずかな依存性を表現する場合に有効である．また，パラメータ数も 1 つであり，シンプルな構造を持つためにとても取扱いやすい．θ は $-1 \leq \theta \leq 1$ を満たし，0 のとき 2 変量間には依存性がないことを表し，正で絶対値が大きいほど変量間の正の依存度が強いことを表し，負で絶対値が大きいほど負の依存度が強いことを表す．

3　モデルの構築

3.1　モデルの概要

従来の 1 変量 SRGM はテスト時間要因にのみ依存したモデルであり，広く利用されているが，現実的ではない前提条件が多いモデルであり，テストの網羅度やテストケースによりフォールトが内在するプログラムパスが実行された場合にしかフォールト発見できないことなどを考えると，信頼度の評価モデルとしては十分なものとは言い難い．本研究で提案するモデルは，テスト時間要因に加えて，テスト項目数の消化量による要因の 2 つの要因を変量として扱い，2 変量間の依存性をコピュラによって表現しモデル化を行う．テスト時間要因は，従来の SRGM 同様に，発見フォールト数データを扱う．後者の要因には，実際のテスト工程においても容易にデータの収集が可能であることから消化テストケース数を扱う．

3.2　諸量の定義

X_i：第 i 日のテストにおいて発見・修正されるフォールト数（確率変数，$i = 1, 2, \ldots, n$）
Y_i：第 i 日のテストにおいて消化されるテスト項目数（確率変数，$i = 1, 2, \ldots, n$）
λ_i：X_i の期待値（$\mathrm{E}[X_i]$）
μ_i：Y_i の期待値（$\mathrm{E}[Y_i]$）
$C_i(u_{X_i}, u_{Y_i}\,;\theta\,)$：第 i 日を記述する 2 変量 (u_{X_i}, u_{Y_i}) の Farlie-Gumbel-Morgenstern（以下，FGM と略す）コピュラ．ここで，θ は 2 つの変量の依存度を表す定数パラメータである．

3.3　フォールト数の分布

X_i に関する実測データは，1 日毎に収集された発見・修正フォールト数であり，各フォールトは発見された時点でその修正・除去がなされている．X_i の従う分布モデルとして，第 i 日ごとに期待値が変化するポアソン分布としてモデル化する．すなわち，

$$p_{X_i}(x) = \Pr[X_i = x] = \frac{\lambda_i^x e^{-\lambda_i}}{x!}(x = 0, 1, 2, \ldots), \tag{7}$$

$$\lambda_i = d \cdot e^{-b \times i}(b \times i)^{r-1}(d, b > 0, r \geq 1), \tag{8}$$

を仮定する．ここで，b，d，および r は未知の定数パラメータであり，特に r は形状パラメータとして機能する．図 2 に r を $r = 1$，および $r = 2$ とした場合の λ_i の概形の例を示す（$f = 100, m = 0.5$）．

3.4　テストケース数の分布

Y_i に関する実測データは，1 日毎に収集された消化テストケース数であり，この量 Y_i は以下のポアソン分布に従うとする．

$$p_{Y_i} = \Pr[Y_i = y] = \frac{{\mu_i}^y e^{-\mu_i}}{y!}(y = 0, 1, 2, \ldots). \tag{9}$$

ここで，μ_i に関して以下の 2 つのモデルを考える．

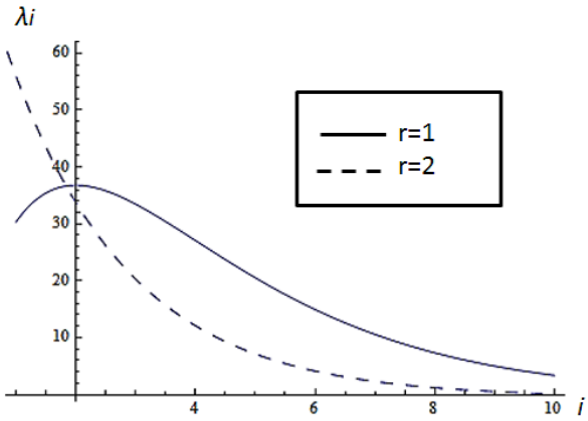

図 2: パラメータ r による λ_i の振る舞いの概形.

- モデルA：$\mu_i = \mu\ (> 0)$，すなわち，期待値はテスト工程を通じて一定とするポアソン分布を仮定する．1 日に消化するテストケース数に大きな変化が見られない場合やテスト日数が浅く，推定できるほどデータが蓄積されていない場合などに有用であると考えた．推定すべきパラメータが 1 つであることから，計算が容易であるというメリットを持つ．
- モデルB：$\mu_i = f \cdot e^{-m \times i}(m \times i)^{h-1}(f, m > 0, h \geq 1)$. すなわち，ある日に消化されるテストケース数は，その期待値がテスト日によって変化するポアソン分布と仮定する．期間を通じて消化するテストケースにバラつきがある場合に有用であると仮定した．推定すべきパラメータ数が 3 つであることから計算が複雑であり，推定しづらい点はあるものの，モデルAよりも柔軟にデータに適応することができる．

3.5 コピュラ関数を用いた信頼性モデル

本研究では，取扱いの容易さにより，FGM コピュラを採用した．上記の 2 つの変量 X_i および Y_i を周辺分布として定式化すると，同時分布関数は

$$C_i(P_{X_i(x)}, P_{Y_i(y)}; \theta) = P_{X_i}(x)P_{Y_i}(y) \times (1 + \theta(1 - P_{X_i}(x))(1 - P_{Y_i}(y))), \tag{10}$$

となる．ここで，

$$P_{X_i}(x) = \sum_{j=0}^{x} p_{X_i}(j),\ P_{Y_i}(y) = \sum_{j=0}^{y} p_{Y_i}, \tag{11}$$

である．また，2 つの周辺分布の依存性を表す θ は $-1 \leq \theta \leq 1$ を満たす．

3.6 パラメータ推定

構築したモデルの推定すべき未知パラメータ数はそれぞれ 5 つ (モデルA)，および 7 つ (モデルB) である．最尤法によりコピュラのパラメータの推定を行う．ここで，式 (10) に対し，最尤法を用い

ることは，全パラメータの推定を同時に行うことになり，計算が非常に複雑になり，推定が困難になる．そこで，まず，文献 [7] に基づき，それぞれのデータによって周辺分布のパラメータの推定を行い，その後，式 (10) に与えてから θ を最尤法により推定することとする．

4　適用例

本章では，前節までで構築してきたモデルに実測データを適用し，その結果を示す．

4.1　使用データ

前節までで構築したモデルに対し，表 1 に示した，ソフトウェア開発工程のうち 27 日間のテスト期間のデータを用いる．フォールト発見数と共に，消化されたテストケース数が記録されている．フォールトは発見された時点で，修正・除去が行われている．なお，本研究では用いるデータが 1 日毎に収集された離散データであるため，離散型のモデルとなる．

表 1: 実測データ.

i	フォールト数 X_i	テストケース数 Y_i
1	3	214
2	3	445
3	13	148
4	16	171
5	12	59
6	13	480
7	12	536
8	8	209
9	19	152
10	3	19
11	1	507
12	13	292
13	2	206
14	7	316
15	4	46
16	4	25
17	8	39
18	4	54
19	2	54
20	0	49
21	4	72
22	3	75
23	3	65
24	3	95
25	0	13
26	0	59
27	1	43
合計	161	4443

4.2 フォールト数の分布の推定

表 1 の発見フォールト数のデータを式 (7)，式 (8) に適用し，未知パラメータ $\hat{d}$,$\hat{r}$,$\hat{b}$ を最尤法により求める．その結果以下の表 2 のように求められた．以後，どのモデルにおいても共通してこれらのパラメータを用いる．ここで，推定日時とは，例えば i=22 日目までのデータを用いて，その時点におけるパラメータの推定を行ったことを表す．テスト終了 6 日前からのモデルのあてはまりを調べる．

表 2: フォールト数のパラメータ推定結果.

推定日時 i	フォールト数		
	$\hat{d}$	$\hat{r}$	$\hat{b}$
22	31.509	2.3179	0.22770
23	31.549	2.2379	0.21347
24	31.405	2.1583	0.19963
25	31.546	2.2284	0.21159
26	31.560	2.2861	0.22124
27	31.564	2.2745	0.21934

また，全データを使って求めたパラメータから，算出した期待値 λ_i の振る舞いは以下の図 1 のようになる．プロットは実測値，中央の曲線は λ_i，上下の曲線は 3 シグマである．この結果から，λ_i は 3 シグマに収まっており，このモデルの下では，統計的に異常なデータ (外れ値) は生じていないといえる．

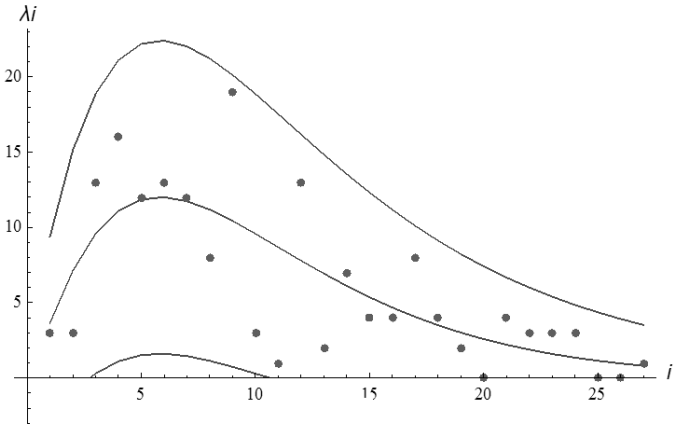

図 3: 3 シグマの信頼区間による期待値 λ_i の振る舞い.

4.3 テストケース数の分布モデルＡの推定と結果

表 1 の消化テストケース数のデータを式 (9) に適用し，未知パラメータ $\hat{\mu}$ を最尤法により求める．モデルＡと実測値の振る舞いを図 4 に示す．図 4 から 14 日目より以前のデータにはある程度適合しているが，15 日目以降のデータに関しては大きく外れてしまっていることがわかる．推定結果と表 2

のフォールト数の分布の推定パラメータを式 (10) に適用し，$\hat{\theta}$ を求めた結果を表 3 に示す．$\hat{\theta}$ の値を見ると，ほぼ全ての日に-0.35 程度であり，わずかながらではあるが負の相関を持たせていることがわかる．また，27 日間全データを用いて推定したパラメータによる同時確率関数の一部を図 5，図 6，図 7，図 8 に示す．これらの図から，モデルAは，テストケース数に関して期待値が一定の分布を用いていることから，発見フォールト数 (X_i 軸) 方向のみ分布が変化していることがわかる．なお，推定日時は表 2 と同様な意味付けである．

表 3: モデルAの各パラメータ推定結果.

推定日時 i	テストケース $\hat{\mu}$	コピュラ $\hat{\theta}$
22	189.455	-0.41294
23	184.043	-0.36038
24	180.333	-0.43664
25	173.64	-0.37300
26	169.231	-0.32437
27	164.556	-0.34117

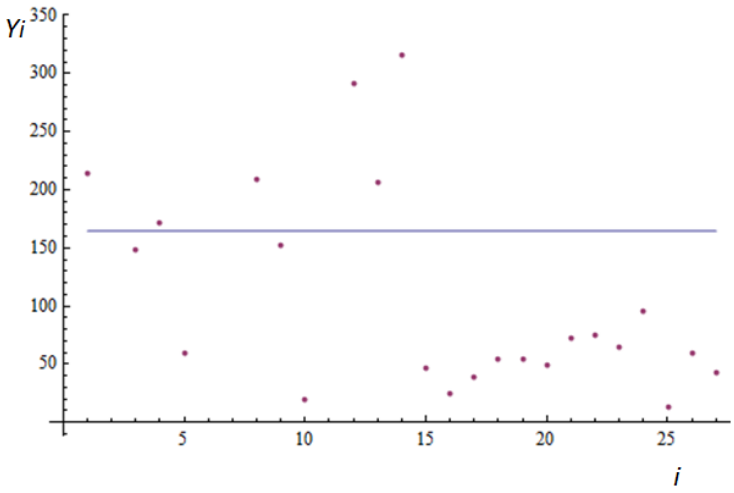

図 4: モデルAと実測値の振る舞いと推定した期待値.

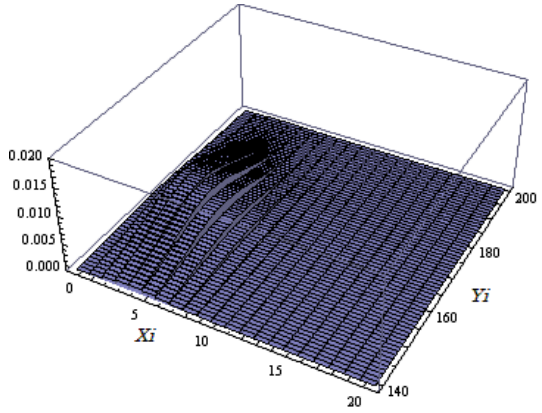

図 5: モデルＡ：1 日目の同時確率関数.

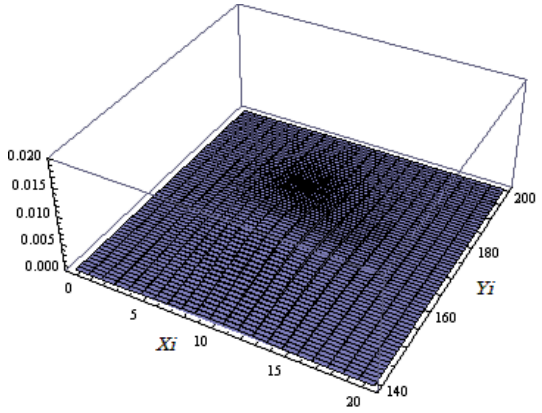

図 6: モデルＡ：10 日目の同時確率関数.

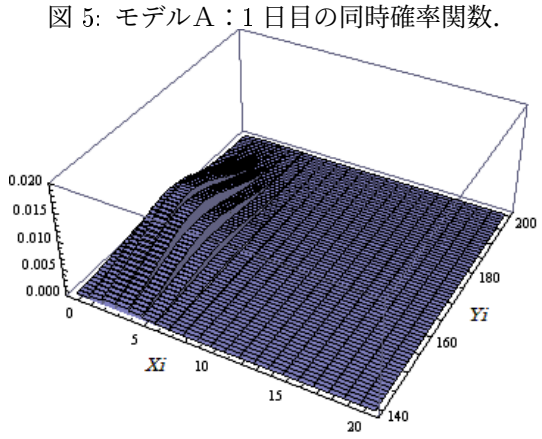

図 7: モデルＡ：20 日目の同時確率関数.

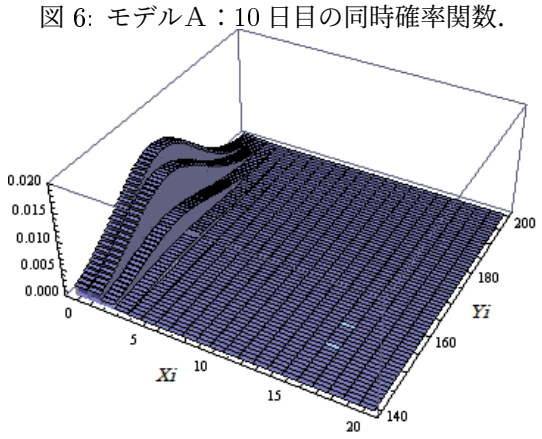

図 8: モデルＡ：27 日目の同時確率関数.

4.4　テストケース数の分布モデルＢと推定，結果

前節までと同様に表 1 のテストケース数のデータを式 (9) に適用し，未知パラメータ $\hat{f}$,$\hat{h}$,$\hat{m}$ を最尤法により求める．推定により求められたパラメータを用いたモデルＢと実測値の振る舞いを図 9 に示す．モデルＡと比べ，14 日目以降のデータに関してはやや当てはまりがよくないように思われるが，15 日目以降に関しては概ね適合した．推定結果を用いて，$\hat{\theta}$ を求めた結果，表 4 のようになった．$\hat{\theta}$ の値を見ると，モデルＡと変わり，ほぼ全ての日において 0 に近い値をとっていることから，依存性はなく，2 変量はそれぞれ独立であるという結果になった．また，27 日間全データを用いて推定したパラメータによる同時確率関数の一部を図 10，図 11，図 12，図 13 に示す．図 13 から，モデルＢは，日によって X_i，Y_i 軸方向にそれぞれ分布が変化していることがわかるが，特にテスト終了間近である 20 日目以降において，日数が経つにつれてフォールト数 (X_i) が 0 に大きな確率を持っていることが見て取れる．

表 4: モデルＢのパラメータ推定結果.

推定日時 i	テストケース数			コピュラ
	$\hat{f}$	$\hat{h}$	$\hat{m}$	$\hat{\theta}$
22	835.507	1.72552	0.16763	-0.03921
23	821.332	1.69724	0.16214	0.04311
24	778.451	1.61898	0.14730	0.07027
25	804.406	1.66526	0.15588	0.02222
26	784.54	1.62947	0.14938	0.08987
27	776.32	1.61516	0.14682	0.09929

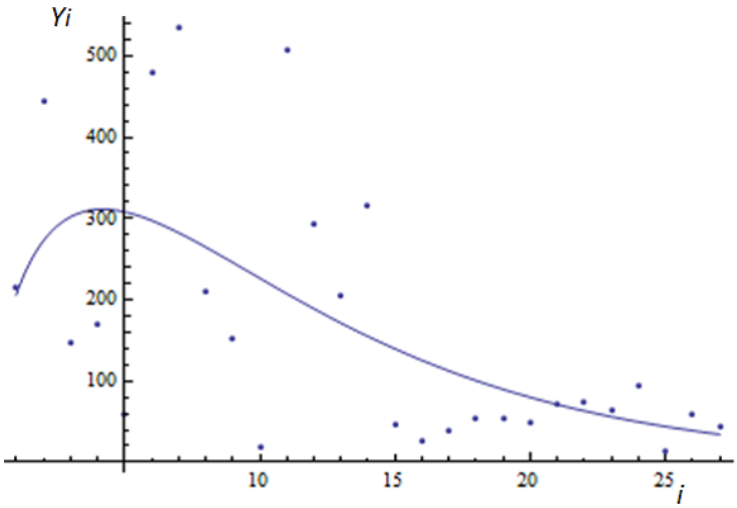

図 9: モデルＢと実測値の振る舞いと推定した期待値.

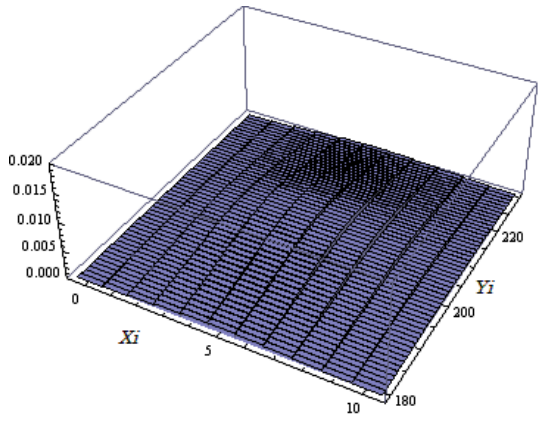

図 10: モデルＢ：1 日目の同時確率関数.

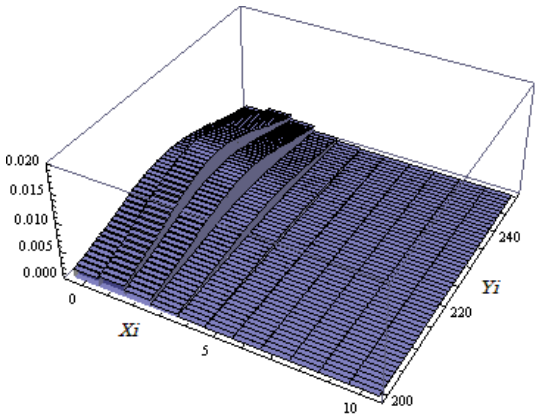

図 11: モデルＢ：10 日目の同時確率関数.

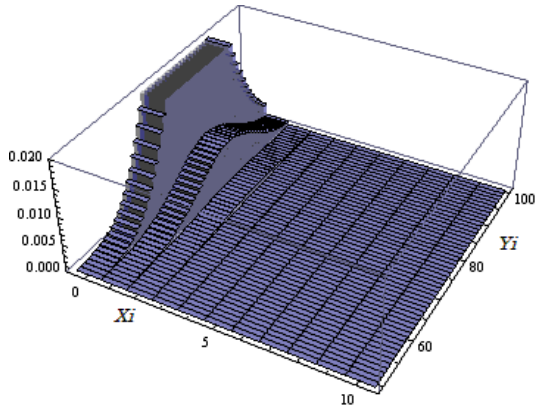

図 12: モデルＢ：20 日目の同時確率関数.

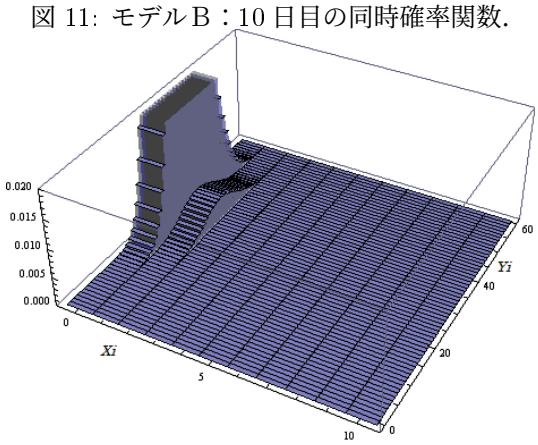

図 13: モデルＢ：27 日目の同時確率関数.

ここまでの結果を踏まえて，モデルＡ，Ｂの比較を行い，信頼性評価を求めていく．まず，両モデルのコピュラ推定時の最大対数尤度から赤池情報基準（以下，AIC と略す）を算出した．AIC は $-2\ln L + 2k$ により求められる．L は最大対数尤度，k は自由パラメータ数である．なお，最大対数尤度は 27 日間全データの $\hat{\theta}$ を最尤推定した際のパラメータで行っている．

モデルＡの最大対数尤度は-1141.77 でありパラメータ数は 5 つであるから

$$\text{AIC} = -2 \times (-1141.77) + 2 \times 5 = 2291.54$$

同様にモデルＢの最大対数尤度は-743.21 でありパラメータ数は 7 つであるから

$$\text{AIC} = -2 \times (-743.21) + 2 \times 7 = 1500.42$$

AIC は値の小さなモデルの方が当てはまりの良いことを表すので，両モデルの AIC を比較して，パラメータの数は多いもののモデルＢの方が当てはまりが良い結果となった．

よって，モデルＢを採用し，まず，周辺分布と実測値の比較によってコピュラが有用であるかを検証するために，27 日目までのデータから推定したパラメータを適用した式 10 からモードによる発見フォールト数と消化テストケース数の $i = 1, \ldots, 27$ のそれぞれの推定を行い，図 14，図 15 のようにそれぞれ発見フォールト数と消化テストケース数の累積値の振る舞いを見てみた．図 14，図 15 を見ると，モデルＢと周辺分布が重なりほぼ同じ振る舞いをする結果となった．フォールト数は実測値よりも大きく推定してしまい当てはまりがあまり良くなく大きく外れてしまう振る舞いをとってしまった．また，周辺分布と重なってしまう理由としては，モデルＢの $\hat{\theta}$ の値がほぼ 0 に近いを取ったため

に依存性がモデルに影響されなかったためである．つまり，モデルＢにおいては，コピュラを適用することでの有用性が見られないということである．なお，推定にモードを用いた理由は，期待値による算出ではコピュラの影響が反映されていないためである．これは，未知パラメータ推定時に周辺分布のパラメータを推定してから，コピュラのパラメータを推定しているためだと思われる．

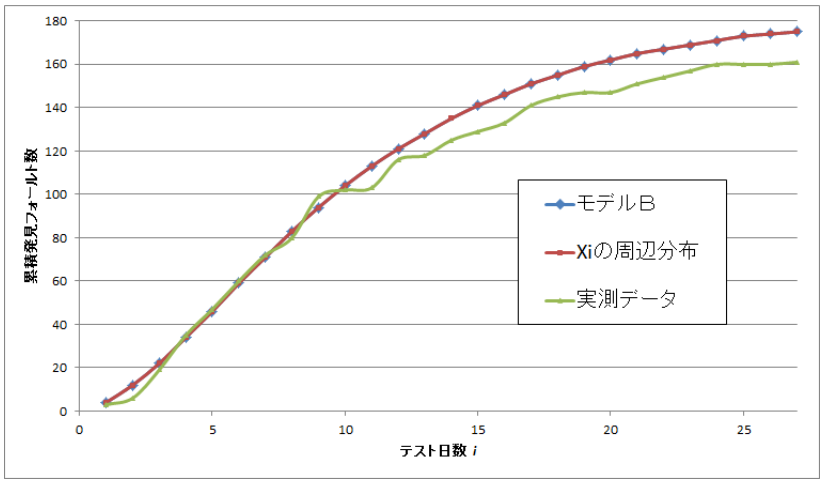

図 14: 累積発見フォールト数の振る舞い．

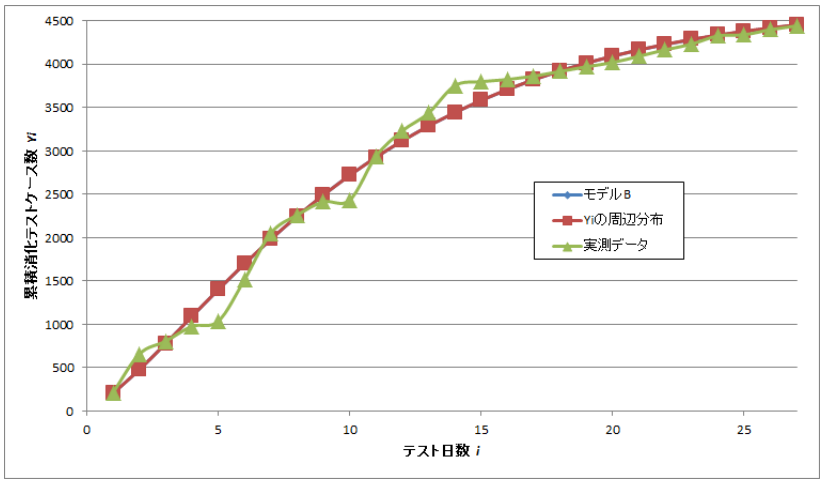

図 15: 累積消化テストケース数の振る舞い．

また，図 16，図 17，図 18，図 19，図 20 に示すような 95 %管理図を作成した [9]．第 i 日目までのデータを適用し，第 $i+1$ 日目の管理図を作成し実測値との比較を行った．図中の☆印は $i+1$ 日での実測値であり，これが網掛けで示された信頼領域に入っていれば，ほぼモデルの予測通りにテストが進捗しているものと判定できる．本研究では，22 日目から 1 日毎に推定し，27 日目までの 5 日間の管理図の作成，および実測値との比較を行った．これにより，途中データでの推定の有用性を検証するためである．しかし，その結果，5 日間のうち 2 日間しか実測値が信頼領域に含まれないという結果となった．注意深く見ると，消化テストケース数に関して外れていることがわかるが，信頼領域

が消化テストケース数が狭い範囲であり，かつ徐々に狭まっていることがわかる．また，発見フォールト数に関しても，0 あるいは 1 に大きく偏った確率を持っていることから，結果として 95%区間の範囲が狭い範囲に集中してしまったと思われる．なお，図 16，図 17，図 18，図 19，図 20 において信頼領域は色のついている領域であり，実測値は☆の印があるセルとなる．

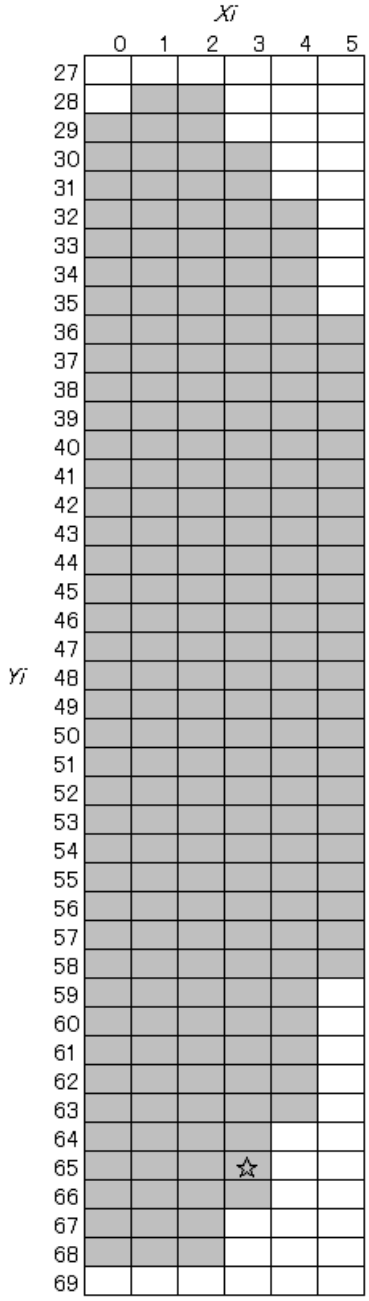

図 16: モデルＢ：23 日目の管理図推定.

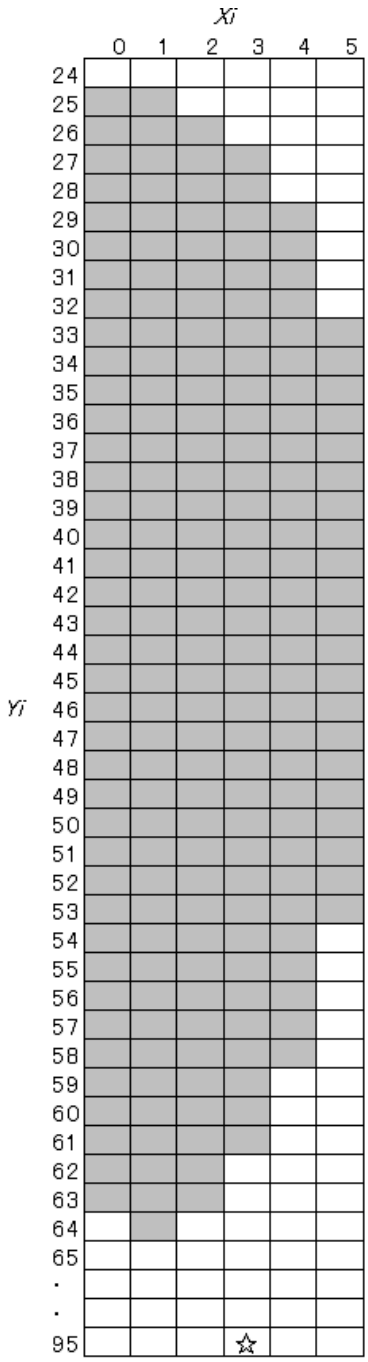

図 17: モデルＢ：24 日目の管理図推定.

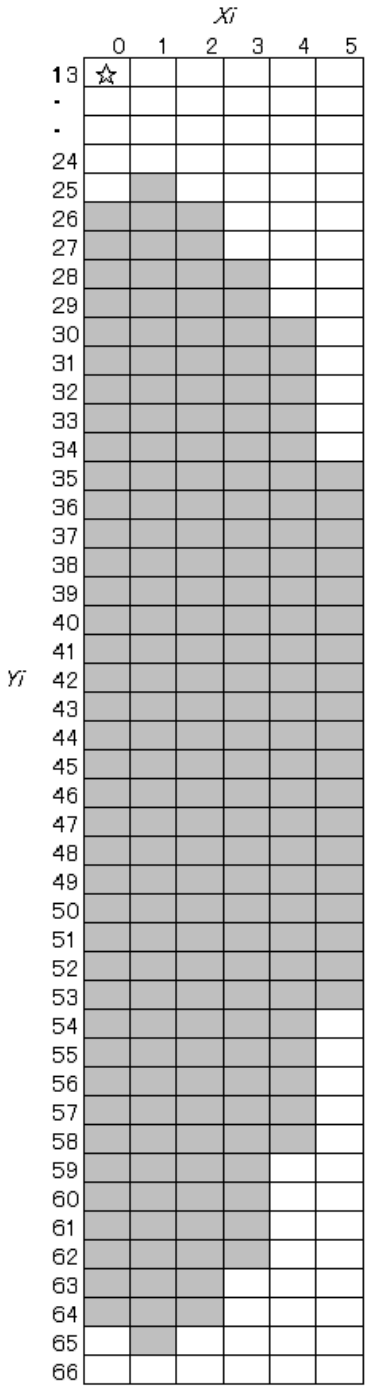

図 18: モデルＢ：25 日目の管理図推定.

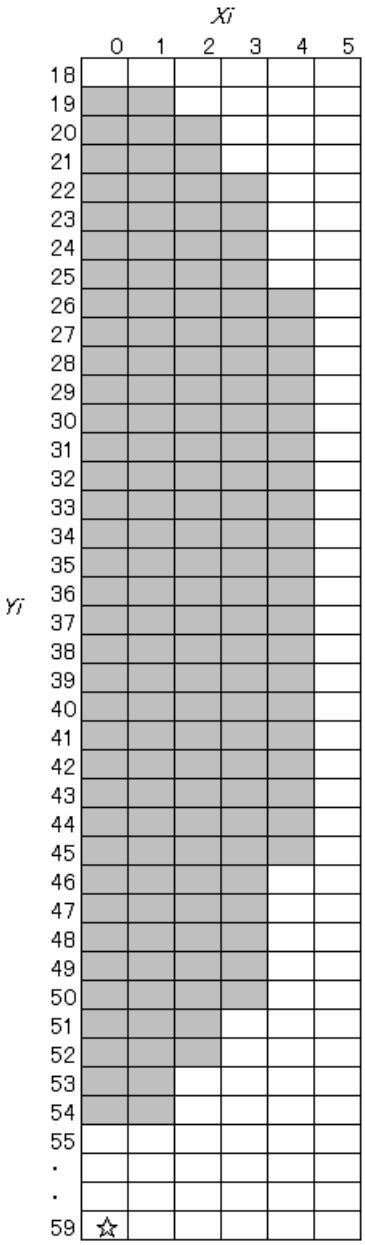

図 19: モデルＢ：26 日目の管理図推定.

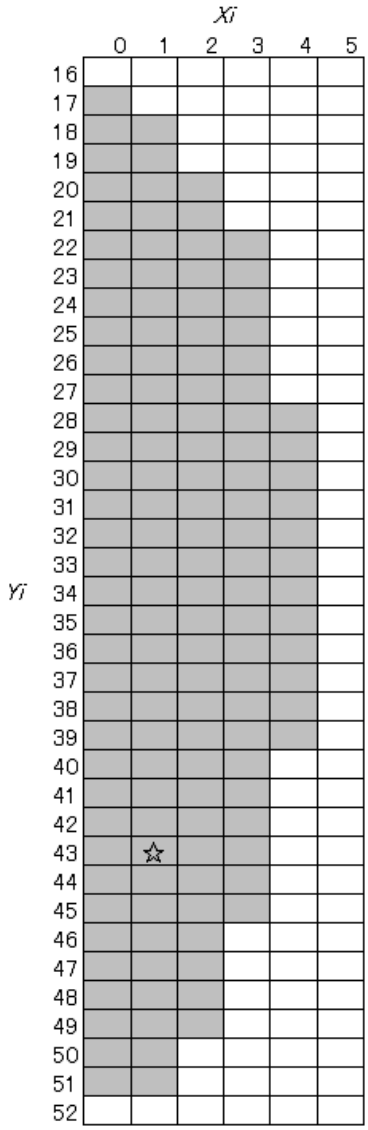

図 20: モデルＢ：27 日目の管理図推定.

ここまでの結果を踏まえ，期待する結果が得られないため，実測データに対してモデルが適合しきれていないと判断し，改めて我々のデータの振る舞いに適ったモデルを再構築する．そこで，テストケースのデータについて着目すると，15 日目を境に大幅に減っていることが見てとれる．この日をチェンジポイント [10] として，15 日目以降のデータをテストケースの分布，モデルA，モデルBに当てはめ改めてモデルC（期待値が一定のポアソン分布），モデルD（期待値が変化するポアソン分布）と再定義し，再度推定することにした．なお，フォールト数の分布に関しては，図 3 の通り，15 日目以降も当てはまりが良いことからそのまま周辺分布として扱った．

4.5 モデルCによる推定

前節までと同様に表 1 のテストケース数の 15 日目以降のデータを式 (9) に適用し，未知パラメータ $\hat{\mu}$ を最尤法により求めた結果の分布の振る舞いを図 21 に示す．これまでのモデルと比べ，ある程度当てはまりが良く見える振る舞いをしている．なお，図にはデータに用いた 15 日目以降の振る舞いだけを示している．推定結果と表 2 のフォールト数の分布の推定パラメータを式 (10) に適用し，$\hat{\theta}$ を求める．その結果以下の表 5 のように求められた．$\hat{\theta}$ を見ると，モデルA，Bと比較して 0.5 から 0.7 程度の値から，2 変量間が正の依存性を持つ結果となった．また，図 22，図 23 にモデルCによる同時確率関数の一部を示す．モデルA，Bに比べフォールト数に関して (X_i 軸) 分布の広がりが見られる．これは，フォールト数の分布に関してはモデルA，Bと同一のものを適用しているため，コピュラによって 2 変量間に正の依存性を持たせたことによるものと考えられる．

表 5: モデルCのパラメータ推定結果.

推定日時 i	テストケース数 $\hat{\mu}$	コピュラ $\hat{\theta}$
22	51.75	0.51080
23	53.22	0.65238
24	57.4	0.74990
25	53.36	0.79612
26	53.83	0.73384
27	53.0	0.69364

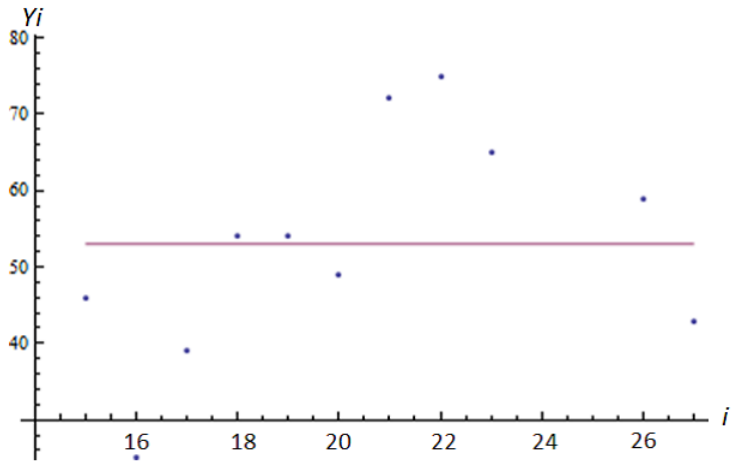

図 21: モデルCの振る舞いと推定した期待値.

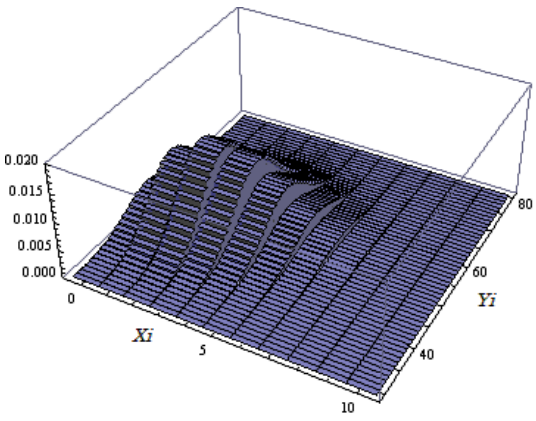

図 22: モデルC：20 日目の同時確率関数.

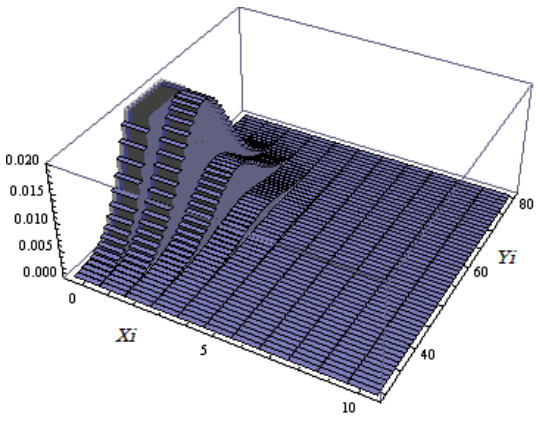

図 23: モデルC：27 日目の同時確率関数.

4.6　モデルDによる推定

前節までと同様に表 1 のテストケース数の 15 日目以降のデータを式 (9) に適用し，未知パラメータ $\hat{f}$,$\hat{h}$,$\hat{m}$ を求めた結果の分布の振る舞いを図 24 に示す．モデルC同様にうまく適合しているように見える．推定結果と表 2 のフォールト数の分布の推定パラメータを式 (10) に適用し，$\hat{\theta}$ を求める．その結果以下の表 6 のように求められた．$\hat{\theta}$ を見ると，モデルCと同様に，0.5 から 0.8 の値をとり，2 変量間が正の依存性を持つ結果となった．また，図 25，図 26 にモデルDによる同時確率関数を示す．モデルCとほぼ同様ではあるが，テストケース数に関して (Y_i 軸) 変化している．

表 6: モデルDのパラメータ推定結果.

推定日時 i	テストケース数 $\hat{f}$	$\hat{h}$	$\hat{m}$	コピュラ $\hat{\theta}$
22	191.6571	1.42136	0.01256	0.57129
23	189.1423	1.46648	0.01753	0.52202
24	210.1346	1.50573	0.01899	0.50123
25	135.53	1.46161	0.05032	0.81764
26	135.395	1.45918	0.04965	0.60806
27	144.179	1.5738	0.07924	0.67004

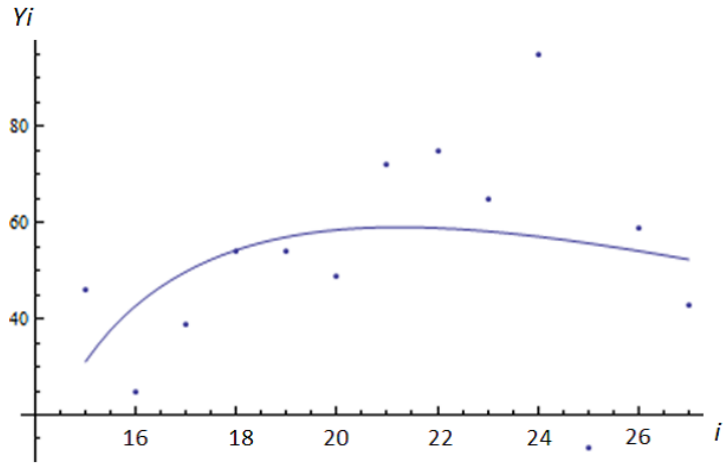

図 24: モデルDの振る舞いと推定した期待値.

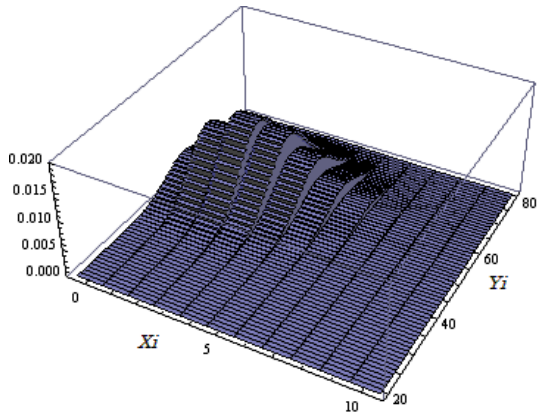

図 25: モデルD：20 日目の同時確率関数.

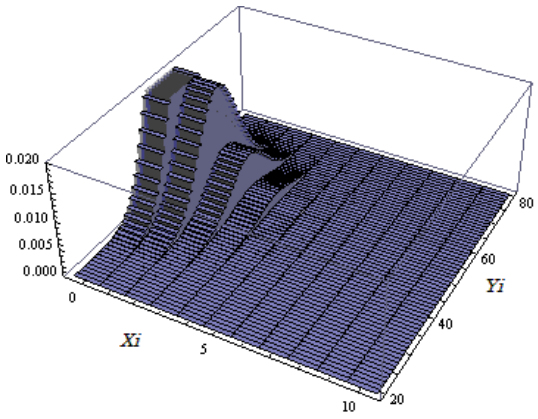

図 26: モデルD：27 日目の同時確率関数.

前節と同様にモデルC，モデルDを比較し，信頼性評価を行う．モデルCの最大対数尤度は-116.907であり，パラメータ数は 5 つであるから

$\mathrm{AIC}=-2\times(-116.907)+2\times 5=243.81$

モデルDの最大対数尤度は-110.565 であり，パラメータ数は 7 つであるから

$\mathrm{AIC}=-2\times(-110.565)+2\times 7=235.13$

この結果からモデルDを採用し，モデルBと同様に信頼性評価例を示していく．まず，図 27，図 28 にモードの推定による累積発見フォールト数，累積消化テストケース数の振る舞いを示す．同様に周辺分布とモデルBの 15 日目以降の結果，実測値の振る舞いも共に示している．ここで，モデルDがフォールト数，テストケース数共に，最も実測値に近い振る舞いをとっていることから，コピュラを適用したことによる有用性が認められる結果といえる．なお，図 27 について，モデルBと周辺分布はほぼ同じ振る舞いをとっている．

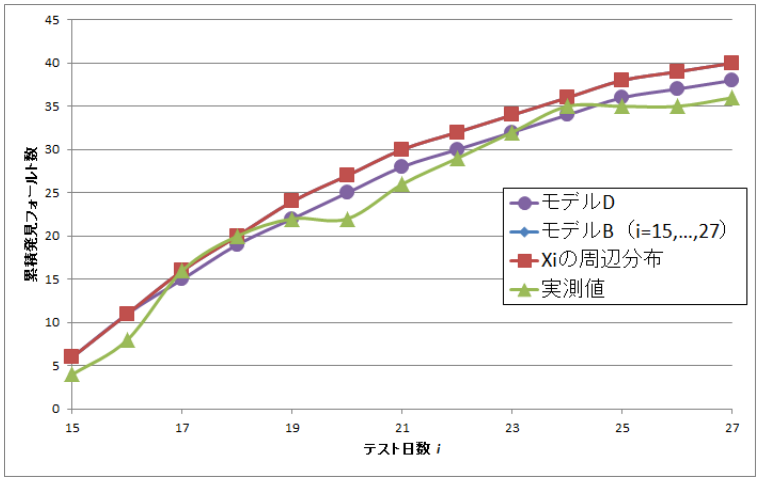

図 27: 累積発見フォールト数の振る舞い.

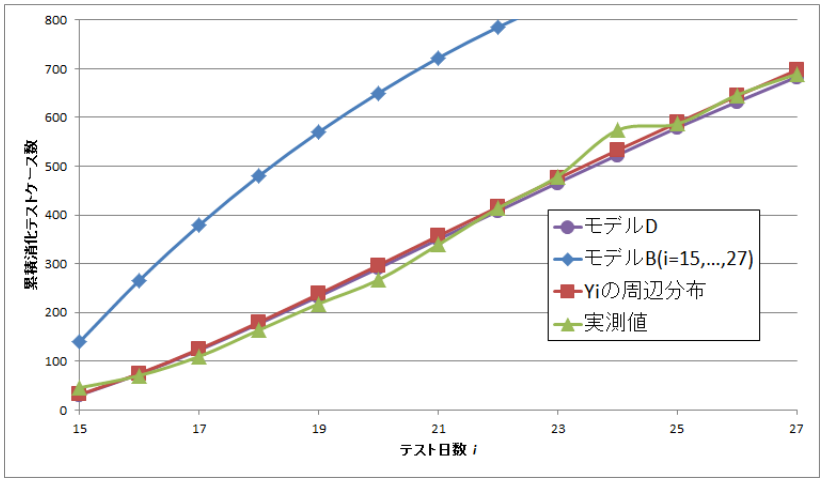

図 28: 累積消化テストケース数の振る舞い.

また，モデルB同様に図 29，図 30，図 31，図 32，図 33 に第 i 日目までのパラメータを用いた第 $i+1$ 目の 95 %管理図を作成し，実測値との比較を行い，途中データでの推定の有用性を検証した．その結果，5 日間のうち 4 日間，実測値が信頼領域に含まれ，精度の高い結果となった．フォールト数に関して，モデルBのように 0 や 1 に偏った領域を持つのではなく，0 から 5 まで広い範囲に領域が存在している．テストケース数に関しても同様に大きな広がりを見せている．

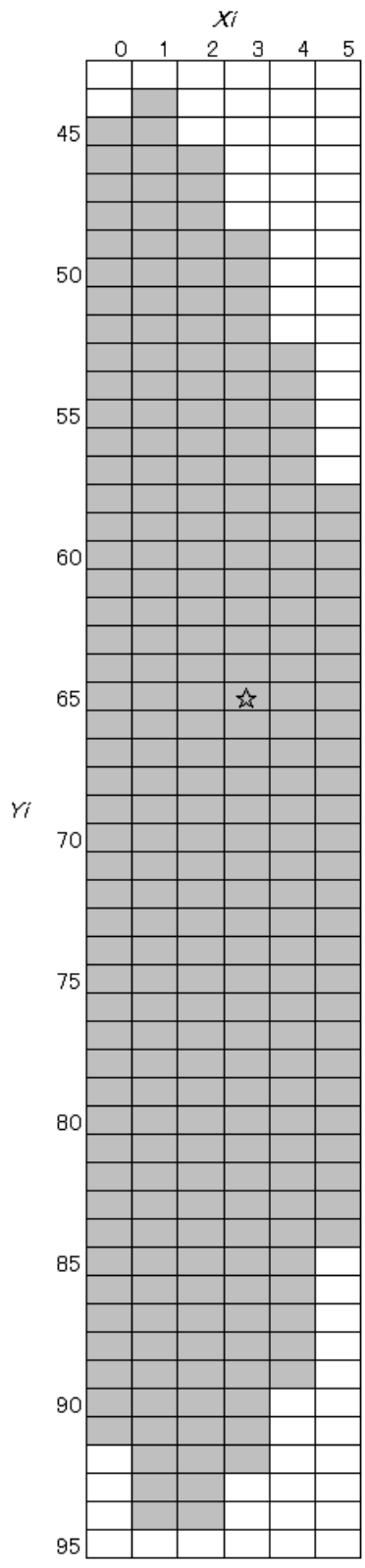

図 29: モデルD：23 日目の管理図推定.

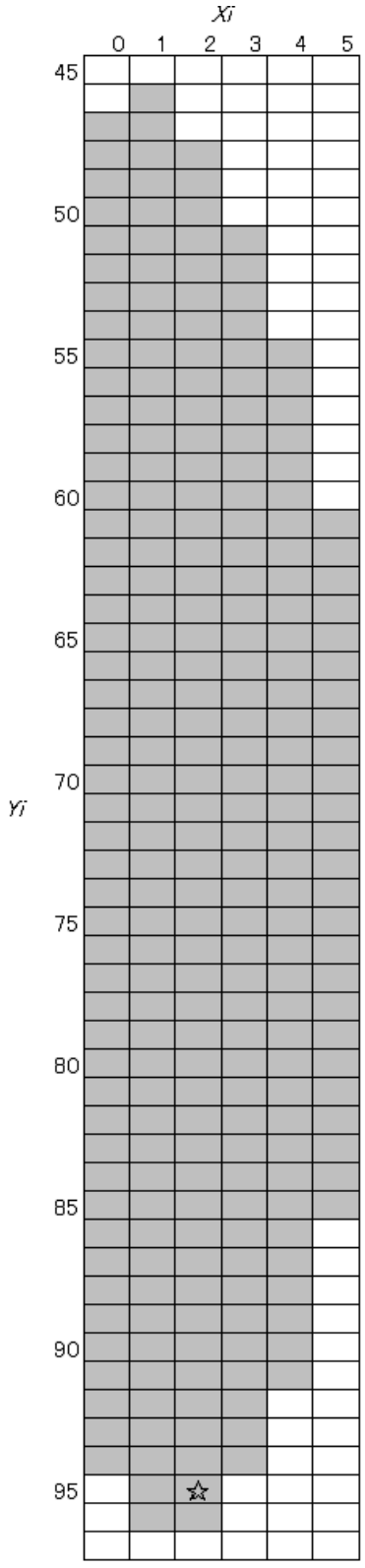

図 30: モデルD：24 日目の管理図推定.

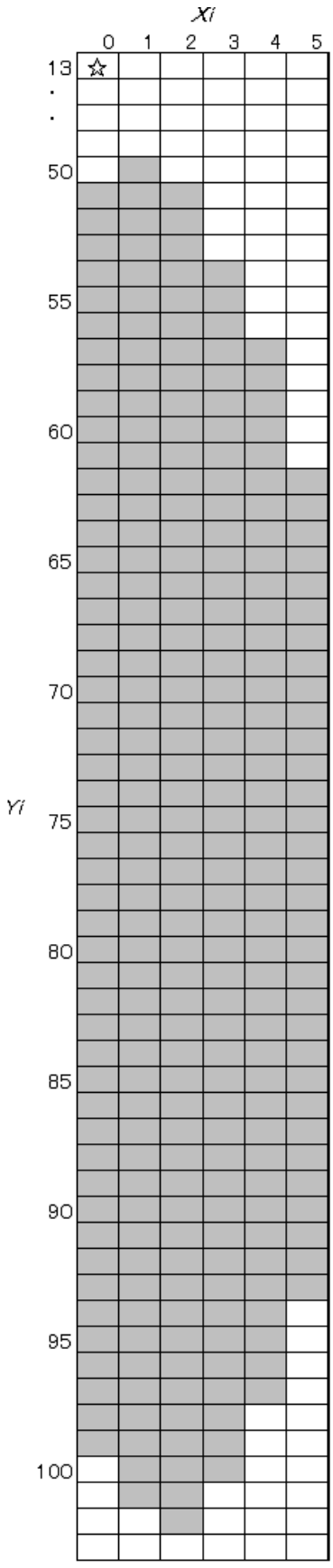

図 31: モデルD：25 日目の管理図推定.

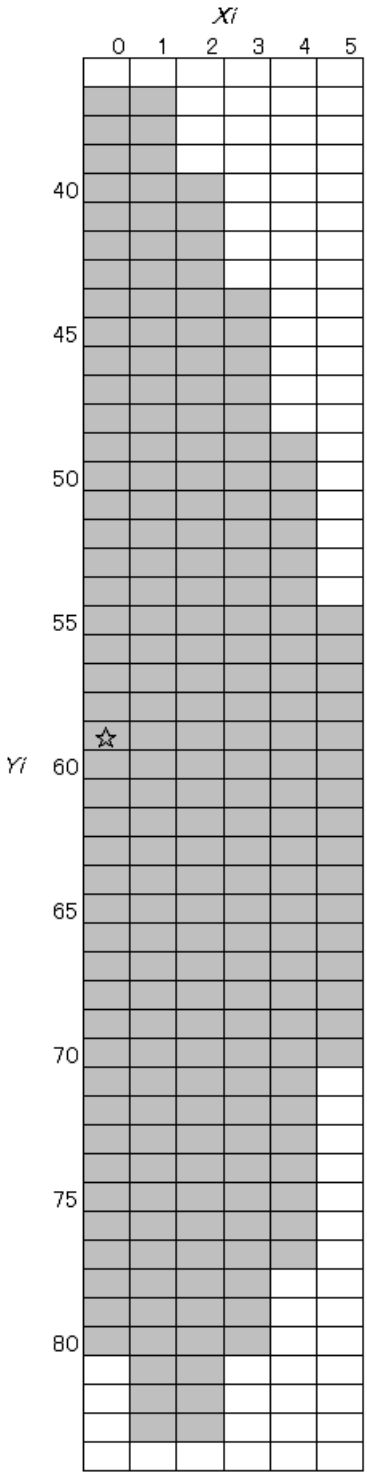

図 32: モデルD：26 日目の管理図推定.

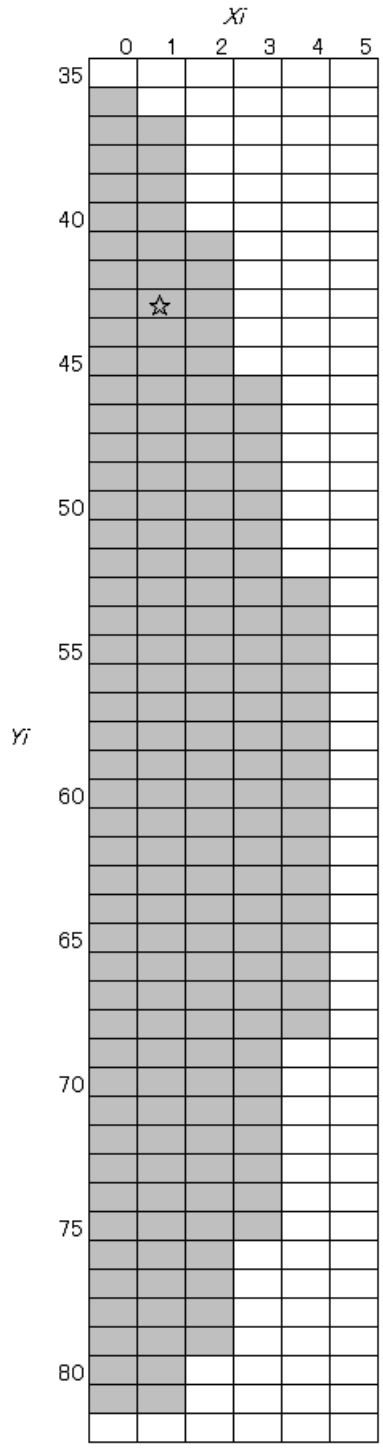

図 33: モデルD：27 日目の管理図推定.

これらの結果から，途中データでの推定も有用であると考えたため，もう 1 つ信頼性評価を行った．テストケースは上流工程であらかじめプログラムやシステムの機能内容などからあらかじめ用意されているためその数は既知数である．このことから，第 i 日目時点のデータを適用し，$i+1, i+2, \ldots, i+n$(n は推定残りテスト期間) とモードにより消化テストケース数を逐次推定し，推定消化テストケース数の累積が残りテストケース数を超えた日をテスト終了日と推定することができる．また，その間の発見フォールト数を推定する．本研究で適用したデータの場合はテストケース数は 4443 個であるから，22 日目から 26 日目までの推定結果を表 7 に示す．22 日間から 24 日間のデータを適用した場合の結果は 1 日早くテストの終了を推定してしまっているが，これは，25 日目の消化テストケースの実測値が 13 と，全テスト期間で最も少ない値を取っており，これが誤差となってしまっていると考えられる．フォールト数に関しては，ある程度推定値と実測値が近い結果となった．

表 7: 第 i 日目までのデータによるモード逐次テスト終了日推定.

22 日	推定値 X_i	実測値 X_i	推定値 Y_i	実測値 Y_i
23	2	3	68	65
24	2	3	71	95
25	1	0	71	13
26	1	0	73	59
27		1		43
合計	6	7	283	275

23 日	推定値 X_i	実測値 X_i	推定値 Y_i	実測値 Y_i
24	2	3	70	95
25	2	0	73	13
26	1	0	73	59
27		1		43
合計	5	4	216	210

24 日	推定値 X_i	実測値 X_i	推定値 Y_i	実測値 Y_i
25	2	0	77	13
26	2	0	80	59
27		1		43
合計	4	1	157	115

25 日	推定値 X_i	実測値 X_i	推定値 Y_i	実測値 Y_i
26	1	0	57	59
27	1	1	56	43
合計	2	1	113	102

26 日	推定値 X_i	実測値 X_i	推定値 Y_i	実測値 Y_i
27	1	1	56	43
合計	1	1	56	43

5　結果と考察

本研究では，コピュラを適用することによって 2 変量の依存関係を表現したモデルを構築し，ソフトウェア信頼性評価に対する有効性を実際の観測データを用いて示した．実際の適用例として，発見フォールト数と消化テストケース数の推定，およびその 95 %信頼区間の推定を行い，その推定結果を示した．最終的な結果として強い正の依存性を持たせ，チェンジポイントに考慮したモデルDの精度が高い結果となったが，テストケースを消化するほどフォールトを発見する数が増え，逆に，消化数が少ない日は発見数が少ないという現実に近い事象を再現していると考えられる．これは，同時確率関数などがフォールト数に関して大きな偏りを見せた依存性を持たないモデルBの信頼性評価例からも捉えることができ，正の依存によって信頼領域が広がるなど，現実面を帯びた事象を如実に表現できたことにより，予測精度を高めることができた．また，適用データによっては，テストの修正に時間が掛かり，テストケース数を消化できないといった事象もあると考えられるが，その場合は負の依存性を持つことになるものと考えられる．いずれにしても θ によってモデルの記述力は増すものと期待できる．また，累積発見フォールト数の振る舞い，累積消化テストケース数の振る舞いから，周辺分布，すなわち 1 変量のモデルとの比較を行うことにより，コピュラがソフトウェア信頼性にも有用であるということを実証できた．加えて，95 %管理図の逐日推定，モードによりテスト終了日推定により，途中データを用いた場合の予測精度も高い結果を示せた．しかしながら，今回適用したデータではチェンジポイントが存在したため，どのテスト段階で精度があるかというところまでは求めることができなかったが，15 日目以降のデータで良い結果が得られたことから，推定するに十分なデータを蓄積した 7，8 割程度であれば精度の高い結果が得られると考えられる．本研究では，テスト時間要因とテスト項目消化数による要因の 2 つに依存性を考慮したモデルとなったが，今後の課題としてよりモデルの精度を上げることが挙げられる．そのために，例えば，本モデルは FGM コピュラを用いたが他のコピュラも含めて取捨選択を行うことや，本研究のようにチェンジポイントがある場合の周辺分布の扱いを考慮し，恐らくテスト人員数の変化などの他の要因をモデルに組み込むことなどが挙げられる．これらの課題に取り組んでいくことにより，安定したソフトウェア開発，ひいてはこれからの情報化社会のよりよい発展に貢献できると考えている．

参考文献

[1] Mint(経営情報研究会)：ソフトウェア開発のすべて, 日本実業出版社 (2000).

[2] 山田 茂：ソフトウェア信頼性の基礎—モデリングアプローチ，共立出版 (2011).

[3] 井上 真二，山田 茂：「高信頼性ソフトウェア開発のための最適テスト労力投入問題」，REAJ 誌, Vol. 32，No. 1 (2010).

[4] 井上 真二，山田 茂：「2 変量ワイブル型ソフトウェア信頼度成長モデルとその適合性評価」，情報処理学会論文誌，Vol. 49，No. 8，pp. 2851-2861 (2008).

[5] 木村 光宏，山田 茂：「テスト項目の消化過程に基づくソフトウェア進ちょく度評価モデルに関する考察」，電子情報通信学会，Vol．J79-D-1，No. 12，pp. 1211-1217(1996).

[6] 安信 那有子，木村 光宏：「死滅過程モデルのソフトウェア信頼性評価への応用」，法政大学大学院工学研究科，平成 23 年度修士論文 (2012).

[7] 戸坂 凡展，吉羽 要直：「コピュラの金融実務での具体的な活用方法の解説」，日本銀行金融研究所，金融研究，pp. 115-162 (2005).

[8] A. J. McNeil, R. Frey, P. Embrechts：定量的リスク管理ー基礎概念と数理技法ー，共立出版 (2008).

[9] 坂上 敏樹，木村 光宏，藤原隆次：「離散 FGM コピュラを用いたソフトウェアテスト管理図」，電子情報通信学会総合大会報文集 (2013) (発表予定).

[10] S. Inoue, K. Fukuma, and S. Yamada, “Multiple change-point modeling for software reliability assessment and goodness-of-fit comparisons”, Information: An International Interdisciplinary Journal, Vol. 16, No. 1(B), pp, 765-770 (2013).

謝辞

本研究を進めるにあたり終始多大なご助言，ご指導をいただいた本学院工学研究科システム工学専攻，木村光宏教授に心より感謝の意を表します．貴重なデータを提供して頂いた，株式会社 SRATECH Lab の藤原隆次博士に感謝申し上げます．また，多くのご指摘や助力していただいた信頼性工学研究室の諸先輩，同期，後輩諸氏に感謝いたします．

著者の文献リスト

[1] 坂上 敏樹，木村 光宏，藤原隆次：「離散 FGM コピュラを用いたソフトウェアテスト管理図」，電子情報通信学会総合大会報文集 (2013) (発表予定).