

ニューラルネットを用いたソフトウェア信頼性予測モデル

正員 高田 義広[†] 正員 松本 健一[†] 正員 鳥居 宏次[†]

A Software Reliability Prediction Model Using a Neural Network

Yoshihiro TAKADA[†], Ken'ichi MATSUMOTO[†] and Koji TORII[†], Members

あらまし 開発中のソフトウェアの信頼性を予測するためには、複数の説明変数をもつ予測モデルを、開発組織やソフトウェアの種類に応じて構築する必要がある。本論文では、測定可能な多くの変数の中から信頼性に強く影響する変数の組合せを選択し、それらを説明変数とする予測モデルを構築するための系統的な方法を提案する。説明変数と目的変数の間の依存関係については、多くの場合に線形関係が見出せないため、従来の線形モデルでは必ずしも現実世界を表現しきれていない。本論文では、ニューラルネットによる非線形モデルを導入し、その有効性を線形モデルと比較する。説明変数の選択には、AICを基準とする変数増加法を使用する。評価実験では、100件のソフトウェア開発から収集した21変数のデータを使用してモデルを構築した。構築したモデルを他の30件のソフトウェア開発に適用して予測精度を評価した。主な結果として次のことがわかった。(1)線形モデルと比較して、ニューラルネットによる非線形モデルの予測精度が優れている。(2)AICを基準とした説明変数の選択が非線形モデルに対しても有効に働く。

キーワード ソフトウェア信頼性, ニューラルネット, 重回帰分析, 変数増加法, AIC

1. まえがき

ソフトウェア開発管理における重要な問題は、ソフトウェア信頼性の予せぬ低下により、テスト、修正、保守の労力や期間が極端に増大することである。このような問題を未然に防止するためには、開発中のソフトウェアの信頼性を監視する必要がある。そのために、信頼性に影響する要因を測定し、そのデータに基づく信頼性予測モデルを構築する試みが行われている^{(2),(4)}。

しかし、開発するソフトウェアの種類、開発組織などに依存しない汎用的な予測モデルを構築することは容易でない。特に、予測モデルに使用する説明変数を決定することは容易ではない。ソフトウェアの信頼性には、非常に多くの要因が複雑に影響するためである。信頼性に影響する要因としては、開発するソフトウェアの規模や複雑さのほかに、開発者の能力や経験、開発方法や環境上の制約、各工程に費やした労力などが挙げられる⁽²⁾。また、開発組織の制約や慣習に依存して、測定されている(または、測定可能である)変数が

異なる場合や、同種類の変数でも測定の方法が異なる場合が多い。従って、実用的な予測モデルを構築するためには、開発組織に応じて、あるいは開発するソフトウェアの種類に応じて、測定可能な変数の中から説明変数の組合せを適切に選択する必要があると考えられる⁽⁸⁾。

説明変数と目的変数の間の関係の表現についても問題が残されている。多変量解析の一つの重回帰分析⁽⁹⁾を適用する場合、その依存関係は次のような線形式によって表現される。

$$Y = a_0 + a_1X_1 + \dots + a_nX_n + E \tag{1}$$

ここで、 Y は目的変数、 $X_i (i=1, \dots, n)$ は説明変数、 $a_i (i=0, \dots, n)$ は係数、 E は予測値と実測値の間の残差である。上式のような線形モデルでは、ソフトウェアの信頼性に対する各要因の影響は独立であり、加法的に現れることを前提としている。しかし、各要因は相互に影響するため、このような線形モデルが妥当であるとは限らない。変数間の非線形な関係を測定データより自動的に抽出する分析方法が要求される。

本論文では、測定可能な多くの変数の中から信頼性に強く影響する変数の組合せを選択し、それらを説明変数とする予測モデルを構築するための系統的な方法

[†] 奈良先端科学技術大学院大学情報科学研究科, 生駒市
Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-01 Japan

を提案する。提案する方法では、目的変数と説明変数の非線形な依存関係を表現するために、階層型ニューラルネットを使用する。階層型ニューラルネットとは、入力層と出力層の間の多数の処理ユニットとユニット間を結びリンクの網によって非線形な依存関係を表現するモデルである。また、多数の候補の中から説明変数の組合せを選択するには、収集したデータに基づき必要な変数を1個ずつ付加する方法(変数増加法)を使用する⁽⁶⁾。異なる説明変数をもつ予測モデルの評価には、統計的な情報基準 AIC (Akaike's Information Criterion)^{(1),(7)}を使用する。

評価実験として、実際のソフトウェア開発から収集したデータに対して提案する方法を適用し、非線形予測モデルを構築した。また、重回帰分析により線形予測モデルも構築し、両者の比較を行った。以降、2.では、使用したニューラルネットを示す。3.では、線形モデルの構築方法と対比しながら、非線形予測モデルの構築方法を説明する。4.では、実験に使用したデータについて述べ、5.では、そのデータを使用した線形予測モデルと非線形予測モデルの構築について述べる。6.では、構築した線形予測モデルと非線形予測モデルとの比較、および、選択された説明変数の特徴について考察する。7.では、結果についてまとめる。

2. ニューラルネットモデル

使用したニューラルネットモデルは、図1に示すように、入力層、中間層、出力層からなる。各層に含まれる処理ユニットは隣接する層の処理ユニットと重み付きリンクで結合している。入力層から説明変数の値が入力され、出力層から目的変数に対する予測値が出力される。入力層の各処理ユニットに入力された説明

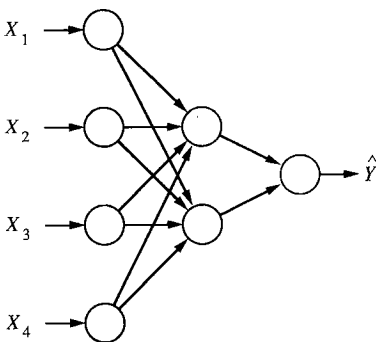


図1 ニューラルネット
Fig. 1 Neural net.

変数の値は、そのまま中間層の処理ユニットに送られる。中間層に送られた説明変数の値はリンクの重みが乗算された後に和が計算される。そして、シグモイド関数 ($g(\theta) = 2 / (1 + \exp(-\theta)) - 1$) によって非線形変換を受けて、出力層の処理ユニットへ送られる。出力層に送られた値は、再びリンクの重みが乗算された後に和が計算され、その和がニューラルネットの出力となる。各ユニットは単純な機能しかもたないが、複数の処理ユニットが結合したニューラルネットの全体は説明変数と目的変数との間の複雑な関係を表現することが可能である。

リンクの重みの決定には、さまざまな学習アルゴリズムが提案されている。本論文では、標準的な誤差逆伝搬アルゴリズム⁽⁶⁾を使用した。この方法では、測定データを使用して、次のようなリンクの重みの更新を収束するまで繰り返す。まず、ニューラルネットの出力と目的変数の値との間の残差平方和を計算する。次に、リンクの重みに対する残差平方和のこう配を計算し、残差平方和を減少させる方向にリンクの重みを少量だけ変化させる。

入力層と出力層に含まれる処理ユニットの数は、それぞれ、説明変数と目的変数の個数に同じであるが、中間層に含まれる処理ユニットの個数には自由度がある。中間層の処理ユニットを多くすると、より複雑な関係を表現できる。しかし、パラメータ(リンクの重み)の個数が多くなるため、学習に使用されるデータの量が少なければ、パラメータの推定の精度が低くなる⁽⁸⁾。逆に、処理ユニットの個数を少なくすれば、データの量が少なくても、パラメータの推定の精度が高くなる。中間層の処理ユニットの個数を決定する方法については3.で説明する。

3. モデル構築方法

提案する非線形予測モデルの構築方法は、線形予測モデル⁽⁵⁾の構築方法の拡張である。本章では、まず線形予測モデルの構築方法について述べ、次に提案する非線形予測モデルの構築方法について述べる。

3.1 線形予測モデルの構築方法

説明変数の候補となる変数と目的変数についての観測データが用意されているものとする。説明変数の組合せを適切に選択して、式(1)のような線形予測モデルを構築する手順は、以下のとおりである。

(ステップ1) 説明変数が1個だけの予測モデルを構築する。説明変数の候補の個数と同じ個数の予測モデル

が構築されることになる。モデルのパラメータ(回帰係数 a_i)の推定には、最小2乗法を使用する。

(ステップ2) 構築した予測モデルの中から、統計的基準により、最良の予測モデルを選択する。

(ステップ3) 選択した予測モデルに対して、残りの説明変数の候補に着目し、任意の1個の説明変数を追加した予測モデルを構築する。説明変数の候補と同じ個数の予測モデルが構築されるため、ステップ2に戻る。追加すべき説明変数の候補がなくなるまで、あるいは、統計的基準による評価値が極端に悪くなるまで、ステップ2, 3を繰り返す。構築された予測モデルの中で、統計的基準による評価値が最小の予測モデルを最終的に選択する。

なお、最終的に選択された予測モデルがもつ説明変数の組合せは、必ずしも最適な組合せではない。説明変数の間に強い関連が存在するとき、ある説明変数を追加することによって、既に追加している説明変数の(目的変数の値を計算する上での)重要性が減少する可能性があるためである。最適な組合せを選択するためには、すべての組合せについて予測モデルを構築し、それらの中から最良のものを選択する必要がある。しかし、そのような総当り的な方法は、膨大な計算を必要とするので現実的でない(n 個の説明変数の候補が存在する場合、 n^2-1 個の予測モデルを構築する必要がある)。上述の方法は、実施が容易な程度に少ない計算で、最適に近い組合せを選択するための方法である($n \cdot (n+1)/2$ 個以下の予測モデルを構築するだけでよい)。

統計的基準としては、AIC^{(1),(7)}を使用する。AICとは、説明変数に対するモデルのゆう度(対数ゆう度)の期待値を(-2)倍した量である。いくつかのモデルを比較する場合、ゆう度の値が大きいほど、良いモデルであると言えるため、AICの値が小さいほど、良いモデルであると考えられる((-2)倍する理由は歴史的経緯に由来する。モデルの評価には影響しない)。一般に、AICは次式によって表される。

$$\text{AIC} = (\text{データ数}) \log(\text{残差平方和}) + 2(\text{パラメータ数}) \quad (2)$$

上式において、第1項はゆう度の一つの推定値の(-2)倍を表し、第2項はその推定値と実際のゆう度との差(-2)倍を表している。第1項の推定値は、文献に述べられているように、不偏推定量ではなく、パラメータ数が多いほど、実際のゆう度の期待値より大きくなる偏りがある。第2項は、その偏りを補正する役割もっている。この特徴により、AICは説明変数の個数の異

なるモデルやパラメータ数の異なるモデルの比較に使用できる。

3.2 非線形予測モデルの構築方法

ニューラルネットによる非線形予測モデルの構築について述べる。3.1の線形予測モデルの構築方法と異なる主な点は、(1)パラメータ(リンクの重み)の推定に学習アルゴリズムを使用すること、(2)中間層の処理ユニットの個数を決定する必要があることである。

中間層の処理ユニットの個数の決定には、説明変数の追加の際に、中間層に処理ユニットも追加するべきかどうかを判定する方法を採用した。説明変数の追加と処理ユニットの追加を合わせた手順は、次のとおりである。

[ステップ1](初期モデルの構築) 説明変数の個数が1個で、中間層の処理ユニットの個数も1個である予測モデルを学習させる。説明変数の候補の数と同じ個数のモデルが構築されることになる。

[ステップ2](処理ユニットの追加) 構築した各モデルに対して、中間層に処理ユニットを1個だけ追加して再び学習させる。処理ユニットの追加前と追加後のすべてのモデルをAICを基準として評価し、1個の最良のモデルを選択する。最良のモデルが処理ユニットの追加後のモデルであれば、ステップ2を再び繰り返し、更に処理ユニットを追加する。

[ステップ3](説明変数の追加) ステップ2で選択したモデルに対して、残りの説明変数の候補に着目し、任意の1個の説明変数をモデルに追加して学習させる。このようにして、説明変数の候補と同じ個数の予測モデルを構築し、ステップ2に戻る。

追加すべき説明変数の候補がなくなるまで、あるいはAICの値が極端に大きくなるまで、ステップ2, 3を繰り返す。AICの値が最小のときの予測モデルを最終的に選択する。

4. 実験用データ

評価実験に使用したデータは、企業ソフトウェア開発から収集した130件分である。作成されたソフトウェアは、開発に使用されるソフトウェアツールである。各ソフトウェア開発については、次の21変数の値が含まれている。

X_{sn} : 新規開発されたモジュールの行数

X_{sr} : 再利用ライブラリーから使用したモジュールの行数

X_{sm} : 他のプログラムの一部を修正して使用したモ

ジュールの行数

- X_{so} : 他のプログラムの一部を流用したモジュールの行数
- X_{df} : 基本設計ドキュメントのページ数
- X_{dd} : 詳細設計ドキュメントのページ数
- X_{do} : マニュアルのページ数
- X_{pf} : 基本設計に費やした日数
- X_{pd} : 詳細設計に費やした日数
- X_{pm} : コーディングに費やした日数
- X_{ef} : 基本設計に費やした人月
- X_{ed} : 詳細設計に費やした人月
- X_{em} : コーディングに費やした人月
- X_{rf} : 基本設計レビューに費やした人月
- X_{rm} : 詳細設計レビューに費やした人月
- X_{rc} : コードレビューに費やした人月
- X_{bf} : 基本設計レビューで発見された欠陥の数
- X_{bd} : 詳細設計レビューで発見された欠陥の数
- X_{bm} : コードレビューで発見された欠陥の数
- Y_b : テスト工程で発見された欠陥の総数
- Y_e : テスト工程に要した人月

データの収集は開発者からの報告書に基づいている。

評価実験では、テスト工程で発見される欠陥 (software fault) の数 Y_b とテスト工程に要する人月 Y_e とを予測する 2 種類のモデルを構築した。これらの変数は、テスト開始時のソフトウェアの信頼性に関連の深い変数である。例えば、 Y_b が大きければ、テスト開始時のソフトウェアは多くの欠陥を含む。 Y_e が大きければ、テスト開始時のソフトウェアは多くの欠陥を含む、または、発見や修正が困難な欠陥を含む。このように発見や修正が困難な欠陥が多いほど、信頼性の低いソフトウェアであると言える。

説明変数の候補としては、テスト工程以前に測定される残りの 19 個の変数を対象とした。モデルのパラメータ (ニューラルネットのリンクの重みや式(1)の a_i など) の推定には、130 件から無作為に抽出した 100 件分のデータを使用した。構築されたモデルの予測精度の評価には、残りの 30 件分のデータを使用した。

5. 評価実験

5.1 線形予測モデル

4. で述べた実験データを使用して Y_b と Y_e の線形予測モデルをそれぞれ構築した。3. で述べた線形予測モデルの構築方法では、残差 E の分布が目的変数の値に依存せずに均一であることを前提として、 E の分散が最

表 1 線形モデルの説明変数の選択

(a) Y_b のモデル		
説明変数	残差分散	AIC
X_{bm}	0.08435	217.2
X_{bm}, X_{sn}	0.06976	200.3
X_{bm}, X_{sn}, X_{sm}	0.06534	195.7
$X_{bm}, X_{sn}, X_{sm}, X_{rf}$	0.06240	193.1
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}$	0.06094	192.7
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}, X_{pf}$	0.05951	<u>192.4</u>
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}, X_{pf}, X_{sr}$	0.05862	192.8
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}, X_{pf}, X_{sr}, X_{so}$	0.05774	193.3
(b) Y_e のモデル		
説明変数	残差分散	AIC
X_{bm}	0.13285	262.7
X_{bm}, X_{ef}	0.09720	233.4
X_{bm}, X_{ef}, X_{em}	0.08005	216.0
$X_{bm}, X_{ef}, X_{em}, X_{sm}$	0.07512	211.6
$X_{bm}, X_{ef}, X_{em}, X_{sm}, X_{do}$	0.07265	<u>210.3</u>
$X_{bm}, X_{ef}, X_{em}, X_{sm}, X_{do}, X_{pd}$	0.07128	210.4
$X_{bm}, X_{ef}, X_{em}, X_{sm}, X_{do}, X_{pd}, X_{df}$	0.07063	211.5

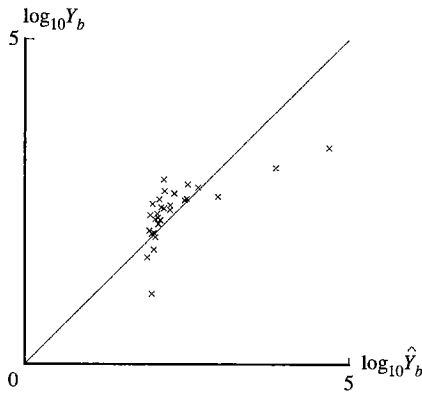
小になるようにモデルのパラメータを推定する。しかし、4. で述べた実験データでは、目的変数の値が大きいくほど、残差も大きくなる傾向があった。そこで、目的変数を対数変換することにより、残差分散の均一化を行った。結果として、次式で表される予測モデルを構築することになった。

$$\log Y = a_0 + a_1 X_1 + \dots + a_n X_n + E \tag{3}$$

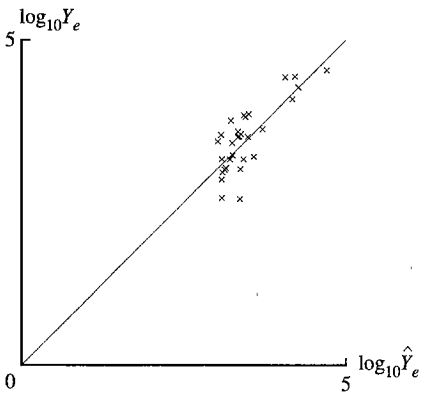
ここで、 Y は Y_b または Y_e を表し、各 $X_i (i=1, \dots, n)$ は 19 個の変数の中の 1 個を表す。変数 E は Y と独立な残差である。

変数 Y_b, Y_e のそれぞれに対する説明変数の選択の過程を表 1 に示す。各行は 1 個の説明変数の追加に対応している。下線で示すように、 Y_b については 6 個の説明変数をもつときに AIC の値が最小であった。 Y_e については 5 個の説明変数をもつときに AIC の値が最小であった。そこで、それらの 6 変数モデルと 5 変数モデルを、それぞれ、最終的に選択した。

選択した予測モデルを、評価用の 30 件のソフトウェア開発にそれぞれ適用した結果を図 2 に示す。横軸はモデルの出力した予測値、縦軸は実測値を示している。図の中の各点は 1 件のソフトウェア開発に対応している。各点が、原点を通る斜線に近いほど、予測誤差が小さいことを意味する。 Y_b については、説明変数を選



(a) 6-variable model for Y_b



(b) 5-variable model for Y_e

図2 線形モデルの予測精度

Fig. 2 Prediction accuracies of linear models.

択したにもかかわらず、斜線から離れた点が存在し、予測誤差が大きいことがわかる。 Y_e については、大きく斜線から離れた点は少なく、予測誤差は比較的小さい。

AICによる予測モデルの評価の結果と30件のソフトウェア開発に適用した結果を比較しよう。説明変数の選択の過程で得られた表1の予測モデルについて、30件のソフトウェア開発にそれぞれ適用した結果を表2に示す。比較的良好にデータと適合していた Y_e の予測モデルについては、表1(b)と表2(b)の最右列を比較するとわかるように、AICの評価による予測モデルの順位と実際の予測誤差を基準とした順位が一致している。 Y_b の予測モデルについては、表1(a)と表2(a)を比較するとわかるように、AICの評価による予測モデルの順位と実際の予測誤差を基準とした順位は一致していない。 Y_b のモデルがデータとあまり適合していない

表2 線形モデルの予測精度の推移

(a) Y_b のモデル

説明変数	誤差分散
X_{bm}	0.1871
X_{bm}, X_{sn}	0.1997
X_{bm}, X_{sn}, X_{sm}	0.1611
$X_{bm}, X_{sn}, X_{sm}, X_{rf}$	0.1960
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}$	0.1825
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}, X_{pf}$	0.1880
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}, X_{pf}, X_{sr}$	0.2132
$X_{bm}, X_{sn}, X_{sm}, X_{rf}, X_{dd}, X_{pf}, X_{sr}, X_{so}$	0.2148

(b) Y_e のモデル

説明変数	誤差分散
X_{bm}	0.2052
X_{bm}, X_{ef}	0.1327
X_{bm}, X_{ef}, X_{em}	0.1307
$X_{bm}, X_{ef}, X_{em}, X_{sm}$	0.1096
$X_{bm}, X_{ef}, X_{em}, X_{sm}, X_{do}$	0.0984
$X_{bm}, X_{ef}, X_{em}, X_{sm}, X_{do}, X_{pd}$	0.0987
$X_{bm}, X_{ef}, X_{em}, X_{sm}, X_{do}, X_{pd}, X_{df}$	0.1117

ために、AICの値や実際の予測誤差が比較的大きく、ばらつきも大きいためであると考えられる。

5.2 非線形モデル

前節の線形予測モデルと同様に変数変換を施した目的変数に対して、ニューラルネットによる非線形モデルを構築した。構築した予測モデルは次式のように表される。

$$\log Y = f(X_1, \dots, X_n) + E \quad (4)$$

ここで、 f はニューラルネットによって表現される非線形関数である。

ニューラルネットの学習(パラメータの推定)には、2.で述べた誤差逆伝搬法を使用した。学習のパラメータとして、学習率を0.001、慣性項の係数を0.9と設定した⁽⁶⁾。一般に、リンクの重みが収束して学習が完了するまでには多くの時間を必要とする。但し、提案する方法のステップ2, 3での予測モデル構築では、前ステップで構築されたニューラルネットの一部を変更するだけであるため、学習に必要な時間も比較的短い。新規にニューラルネットの学習を始める場合と、前ステップで構築したニューラルネットの一部を変更して学習を始める場合の学習の収束の様子を図3に示す。ここでは、 X_{bm} , X_{sn} , X_{ef} の3変数を入力し、 Y_b の予測値を出力する2個のニューラルネットを学習させてい

る。横軸は学習時間(リンクの重みの更新回数), 縦軸は残差平方和を示している。実線は, 新規に学習を始めた場合の収束の様子を表している。すべてのリンクの重みをランダムに与えた後に学習を開始した。破線は, 前ステップで構築したニューラルネットの一部を変更して学習を始めた場合の学習の収束の様子を表している。 X_{bm} と X_{sn} の2変数を入力とする学習済みのニューラルネットの入力層に X_{sn} の処理ユニットを追加した後に学習を開始した。処理ユニットの追加前から存在するリンクの重みは変更せずに, 新たに追加し

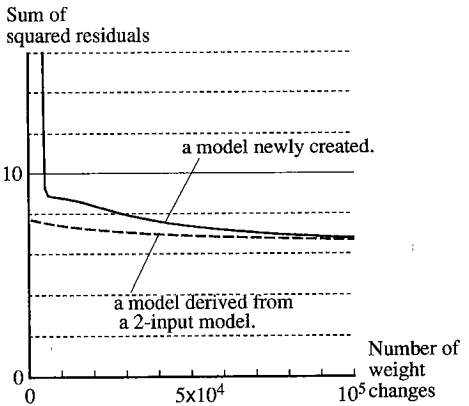
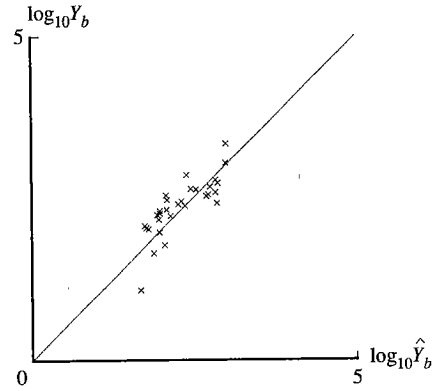
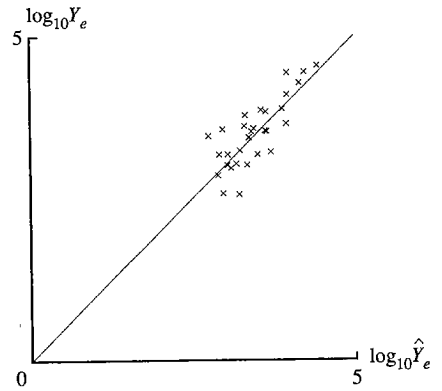


図3 ニューラルネット学習の収束(3入力モデルの例)
Fig. 3 Example of convergence of neural net learning.
(example of a 3-input model)



(a) 5-variable model for Y_b



(b) 4-variable model for Y_e

図4 非線形モデルの予測精度
Fig. 4 Prediction accuracies of nonlinear models.

表3 非線形モデルの説明変数の選択過程

(a) Y_b のモデル

説明変数	中間ユニット	残差分散	AIC
X_{bm}	1	0.05771	183.3
X_{bm}, X_{sn}	1	0.03122	123.8
X_{bm}, X_{sn}, X_{sm}	1	0.02564	106.1
$X_{bm}, X_{sn}, X_{sm}, X_{ed}$	1	0.02481	104.9
$X_{bm}, X_{sn}, X_{sm}, X_{ed}, X_{df}$	1	0.02323	<u>100.3</u>
$X_{bm}, X_{sn}, X_{sm}, X_{ed}, X_{df}, X_{bd}$	1	0.02319	102.1
$X_{bm}, X_{sn}, X_{sm}, X_{ed}, X_{df}, X_{bd}, X_{pf}$	1	0.02276	102.3

(b) Y_e のモデル

説明変数	中間ユニット	残差分散	AIC
X_{bm}	1	0.13198	266.0
X_{bm}, X_{ed}	1	0.07739	214.6
X_{bm}, X_{ed}, X_{ef}	1	0.06427	198.1
$X_{bm}, X_{ed}, X_{ef}, X_{sm}$	2	0.05294	<u>192.6</u>
$X_{bm}, X_{ed}, X_{ef}, X_{sm}, X_{pf}$	2	0.05235	195.7
$X_{bm}, X_{ed}, X_{ef}, X_{sm}, X_{pf}, X_{dd}$	2	0.05031	195.6

表4 非線形モデルの予測精度の推移

(a) Y_b のモデル	
説明変数	誤差分散
X_{bm}	0.1245
X_{bm}, X_{sn}	0.0880
X_{bm}, X_{sm}, X_{sr}	0.0701
$X_{bm}, X_{sn}, X_{sm}, X_{ed}$	0.0718
$X_{bm}, X_{sn}, X_{sm}, X_{ed}, X_{df}$	0.0714
$X_{bm}, X_{sn}, X_{sm}, X_{ed}, X_{df}, X_{bd}$	0.0724
$X_{bm}, X_{sn}, X_{sm}, X_{ed}, X_{df}, X_{bd}, X_{pf}$	0.0733

(b) Y_e のモデル	
説明変数	誤差分散
X_{bm}	0.1805
X_{bm}, X_{ed}	0.1774
X_{bm}, X_{ed}, X_{ef}	0.1184
$X_{bm}, X_{ed}, X_{ef}, X_{sm}$	0.0971
$X_{bm}, X_{ed}, X_{ef}, X_{sm}, X_{pf}$	0.1133
$X_{bm}, X_{ed}, X_{ef}, X_{sm}, X_{pf}, X_{dd}$	0.1176

表5 他の予測モデル

種類	目的変数	説明変数	残差分散	AIC
線形	Y_b	$X_{sn}, X_{sm}, X_{sr}, X_{so}$	0.08491	223.9
非線形	Y_b	$X_{sn}, X_{sm}, X_{sr}, X_{so}$	0.06459	200.5
線形	Y_e	$X_{sn}, X_{sm}, X_{sr}, X_{so}$	0.09054	230.3
非線形	Y_e	$X_{sn}, X_{sm}, X_{sr}, X_{so}$	0.08672	230.0

予測モデルがよく適合している。線形モデルでは予測誤差の特に大きいいくつかのソフトウェア開発が見られたが、非線形モデルではそのような斜線から離れた点が少なくなっている。また、平均的に見ても、評価用の30件のソフトウェア開発に対する非線形モデルの予測誤差(残差分散)が非線形モデルの予測誤差の1/2.5倍に小さくなっている。一方、 Y_e については、線形予測モデルと非線形予測モデルとの間に予測精度の差は認められない。データと線形予測モデルが適合する場合には、非線形予測モデルを使用しても、予測精度が改善されないためであると考えられる。線形予測モデルは非線形予測モデルの特殊な場合であるため、 Y_e のようにパラメータの推定を適切に行えば、予測精度が悪化することはないと考えられる。上記の Y_b と Y_e についての結果は、このような非線形モデルの優位性を肯定するものである。

6.2 選択された説明変数

まず、バグの数 Y_b に対して選択された説明変数について考察する。選択された説明変数の組合せは、線形モデルと非線形モデルと($\{X_{bm}, X_{sn}, X_{sm}, X_{sr}, X_{dd}, X_{pf}\}$ と $\{X_{bm}, X_{sn}, X_{sm}, X_{ed}, X_{df}\}$)の間で同様の傾向を示していた。どちらのモデルに対しても、最初に選択された説明変数は、コードレビューにおいて発見されるバグの数 X_{bm} であった。データの収集源であるソフトウェア開発では、テストの直前の工程として、コードレビューが実施されていた。変数 X_{bm} が最も重要な説明変数として選択された理由は、コードレビューにおいて見逃されたバグの数がテスト時に発見されるバグの数に直接影響するためであると考えられる。2番目と3番目に選択された説明変数は、プログラムの規模に関する変数である。その後選択された説明変数は、設計に関する変数である。どちらのモデルについても、類似した変数が含まれておらず、変数の種類が適当に分散していることがわかる。

選択された説明変数の組合せと比較するために、プログラムの規模に関する変数だけを説明変数として、 Y_b の線形予測モデルと非線形予測モデルを構築した。

たリンクだけの重みをランダムに与えて学習を開始した。前者と比較して後者の方が早くから学習が収束していることがわかる。

変数 Y_b 、 Y_e のそれぞれに対する説明変数の選択の過程を表3に示す。下線で示すように、 Y_b については5個の説明変数をもつときに、 Y_e に対して4個の説明変数をもつときにAICの値が最小であった。これらの予測モデルを、評価用の30件のソフトウェア開発にそれぞれ適用した結果を図4に示す。

また、説明変数の選択過程で得られた表3の予測モデルについて、30件のソフトウェア開発にそれぞれ適用した結果を表4に示す。線形予測モデルと同様に、AICの評価による順位と予測誤差を基準とした順位を比較しよう。表3(a)と表4(a)の最右列を比較した結果、および、表3(b)と表4(b)の最右列を比較した結果からわかるように、 Y_b 、 Y_e のどちらについても、AICの評価による順位と実際の予測誤差を基準とした順位が一致している。AICを基準とする非線形予測モデルに対しても有効であることがわかる。

6. 考 察

6.1 線形モデル・非線形モデルの比較

5.で構築した線形モデルと非線形モデルの予測精度について考える。 Y_b については、図2(a)と図3(a)を比較するとわかるように、線形予測モデルよりも非線形

構築したモデルの予測精度を表5の前半に示す。5で構築したモデルの方が、予測誤差が小さいことがわかる。

次に、テストの労力 Y_e に対して選択された説明変数について考察する。 Y_e についても、選択された説明変数の組合せが線形モデルと非線形モデルとの間でおおよそ一致していた ($\{X_{bm}, X_{ef}, X_{em}, X_{sm}, X_{do}\}$ と $\{X_{bm}, X_{ed}, X_{df}, X_{sm}\}$)。最初に選択された説明変数は、 Y_e の予測モデルと同様に、 X_{bm} である。2番目と3番目に選択された説明変数は、設計やコーディングに費やした労力である。2番目と3番目の説明変数の単位(人月)と目的変数 Y_e の単位が一致している点が興味深い。その後選択された説明変数は、プログラムの規模に関する変数である。バグの数 Y_b と同様に、類似した変数が含まれておらず、変数の種類が適当に分散していることがわかる。

また、 Y_b と同様に、プログラムの規模に関する変数だけを説明変数として、 Y_e の線形予測モデルと非線形予測モデルを構築した。構築したモデルの予測精度を表5の後半に示す。やはり、5で構築したモデルの方が、予測誤差が小さいことがわかる。

7. むすび

本論文では、ニューラルネットを使用して、ソフトウェアの信頼性を予測するための非線形モデルを系統的に構築する方法を示した。提案する方法を実際のソフトウェア開発データに適用した結果より、説明変数の適切な組合せが選択されることおよび線形予測モデルと比較して予測精度の高い非線形モデルが構築されることを示した。

また、実験結果より、データに含まれる19変数の中でテスト開始時のソフトウェア信頼性を予測する上で重要な変数は、コードレビューの質、設計に要した労力(設計ドキュメントの量)、プログラムの規模に関する変数であることがわかった。

本論文で使用したデータは、ソフトウェアツールの開発に関するデータであるが、他のデータに対する評価実験は行っていない。今後、更に多くの種類のソフトウェアについて、提案する方法を適用し、方法の有効性をより明確にしていく予定である。

文 献

(1) Akaike H.: "A new look at the statistical model identification", IEEE Trans. on Automatic Control, AC-19, 6, pp. 716-723 (1974).
 (2) Boehm B. W.: "Software Engineering Economics", IEEE

Trans. on Software Eng., SE-10, 1, pp. 4-21 (1984).

(3) Funahashi K.: "On the approximate realization of continuous mappings by neural networks", Neural Networks, 2, 3 (1989).
 (4) Halstead M.: "Elements of Software Science", Elsevier, New York (1977).
 (5) 久米 均, 飯塚悦功: "回帰分析", 岩波書店(1987).
 (6) Rumelhart D. E., Hinton G. E. and Williams R. J.: "Learning representations by backpropagating errors", Nature, 323, pp. 533-536 (1986).
 (7) 坂本慶行, 石黒真木夫, 北川源四朗: "情報量統計学", 共立出版(1983).
 (8) Takada Y., Torii K. and Thompson M. C.: "Better predictions even in reduced input variable numbers using neural networks in software development", Proc. of 16th Annual Software Engineering Workshop, NASA, pp. 103-120 (1991).

(平成5年7月6日受付, 6年1月11日再受付)



高田 義広

平1 阪大・基礎工・情報卒。平3 同大大学院修士課程了。平6 同大学院博士課程了。同年奈良先端科学技術大学院大学・助手。工博。ソフトウェアの信頼性予測の研究に従事。情報処理学会, 日本音響学会各会員。



松本 健一

昭60 阪大・基礎工・情報卒。平1 同大大学院博士課程中退。同年同大・基礎工・情報・助手。平5 奈良先端科学技術大学院大学・助教。工博。ソフトウェア開発における計測環境, ソフトウェア品質保証の枠組みに関する研究に従事。情報処理学会, IEEE 各会員。



鳥居 宏次

昭37 阪大・工・通信卒。昭42 同大大学院博士課程了。同年電気試験所(現電子技術総合研究所)入所。昭50 ソフトウェア部言語処理研究室室長。昭59 阪大・基礎工・情報教授。平4 奈良先端科学技術大学院大学・教授。工博。ソフトウェア工学の研究に従事。情報処理学会, 日本ソフトウェア学会, 人工知能学会, ACM, IEEE 各会員。