

工数見積もり手法における変数選択手法の比較

天寄 聡介

鳥取環境大学 情報システム学科

概要

ソフトウェア開発では正しく工数を見積もることは重要な問題であるため、様々な工数見積もり手法が提案されてきた。工数見積もり手法の多くはモデルの単純化及び予測精度向上のために変数選択を行っている。工数見積もり手法では相関係数を用いるもの、leave-one-out 法のようなリサンプリング法を用いるものなど様々な変数選択手法が用いられている。しかし、変数選択法間での比較評価を行った結果については報告されていない。そこで、本研究はいくつかのリサンプリング法について、変数選択の性能の比較評価を行った。その結果、データサイズにより最適な変数選択法が異なる可能性があることが明らかとなった。

Comparative Study on Variable Selection Methods for Effort Estimation Model

Sousuke Amasaki

Dept. of Information Systems, Tottori University of Environmental Studies

Abstract

Effort estimation has still been one of the most important issues in software development. Thus, several effort estimation models were proposed. In many of effort estimation models, model selection methods play important role in terms of accuracy improvement and simplification of the model. Although there are also several model selection methods such as correlation-based method, resampling method, etc., comparative study on these methods were not conducted in the past research. Thus, in this paper, comparative study on variable selection methods for effort estimation model were performed. As a result, it was found that there is the possibility that the best variable selection method depends on the size of dataset.

1 はじめに

要求される品質を満たしつつ、定められた期間・コストでソフトウェアを完成させ出荷することは、ソフトウェア開発において最も重要なことである。この目的を達成するには、ソフトウェア開発の実施に先立って適切な計画立案を行う必要がある。特に、ソフトウェア開発に必要な工数を適切に見積もることの重要性は以前から指摘されてきている。こうした背景から、現在までに様々な工数見積もり手法が提案されてきている [4, 7, 10]。

COCOMO [2] のように説明変数間の関係に仮定をおく場合を除き、工数見積もり手法では過去に収

集したデータに基づいて変数選択が行われる。変数選択を行うことにより予測精度の向上およびモデルの単純化が期待できること、モデルの単純化によってモデルの学習に必要なデータセットのサイズが小さくできること、いわゆる「次元の呪い」を軽減できることなどがその理由である。

現在提案されている工数見積もり手法では数種類の変数選択手法が用いられているが、それらの性能に関する比較評価は行われていない。そこで、本研究では工数見積もり手法で広く使われている変数選択手法の性能について比較評価を行った。

本論文の以降の構成は以下の通りである。まず 2 節では変数選択手法の関連研究について述べる。次

に3節では本研究で比較評価を行う変数選択法について詳細に述べる。そして、4節では以降の評価に先立って必要となる事項について述べる。5節では比較評価のための実験手順について述べ、6節でその結果について議論する。最後に7節でまとめと今後の課題について述べる。

2 関連研究

工数見積もり手法で行われる変数選択手法は大きく以下の2つに分類できる。

1. 工数見積もり手法から独立に実施可能な手法
2. 工数見積もり手法の見積もり値を利用する手法

1に属する手法には説明変数の候補と目的変数との相関係数を利用する手法やステップワイズ法による変数選択の結果を利用する方法 [9, 11, 12] がある。

ステップワイズ法は線形回帰モデルの変数選択に用いられる手法で、主に AIC (Akaike Information Criteria) [5] という統計量を線形回帰モデルを用いて求め、その結果を比較して適切な変数の選択を行う。このため、本来は2に属する手法であるが、線形回帰モデル以外の工数見積もり手法においても変数選択だけのために用いられることも多い。

一方、2に属する手法としてはリサンプリング法が広く利用されている。リサンプリング法ではまずデータをモデル学習用と見積もり用に2分割して仮の見積もり値を求める。この作業は複数の異なるデータ分割の仕方に対して繰り返し行う。そして、この結果得られた複数の見積もり値を基に見積もり誤差の推定値を得る。以上の手順を全ての変数の組み合わせに対して行い、推定値を比較して最適な変数を選択する。代表的なリサンプリング法としては交差検証法 (cross-validation) が挙げられる。工数見積もり手法の研究では交差検証法の一つである leave-one-out 法を用いた事例が存在する [17]。

リサンプリング法における誤差の推定量としては、残差平方の平均値を用いるのが一般的である。しかし、工数見積もり手法の研究においては見積もり手

法の評価に $MMRE$ [14] を用いている場合が多いため、変数選択においても $MMRE$ が使用されている [17]。しかし、 $MdMRE$ [14] を誤差の推定量に利用した変数選択が可能なシステムも存在する [15]。

以上のように工数見積もり手法では複数の変数選択手法が用いられているが、手法間の性能評価は行われていない。

3 変数選択手法

3.1 相関係数に基づく手法

相関係数に基づいた変数選択法では、相関係数の高い順に変数を選択する。工数見積もりに使用されるデータに関しては性質上正規性を仮定できないため、ノンパラメトリック相関係数が用いられる。本研究では、スピアマンの相関係数を用いた。

具体的な変数選択の手順を以下に示す。

1. 相関係数を求める。

説明変数 $X = \{x_1, x_2, \dots, x_m\}$ のそれぞれと目的変数 y との間で相関係数 $C = \{c_1, c_2, \dots, c_m\}$ を求める。

2. 変数を選択する。

相関係数 C を降順にソートして上位 k 個の変数を選択する。本研究では $k = 3$ とした。

3.2 ステップワイズ法

ステップワイズ法による変数選択では、まず線形回帰モデルに説明変数を1つ追加する。そしてそのモデルの AIC を評価し、AIC に改善が見られるようであればその変数を追加する。逆に AIC が改善しない、もしくは悪化する場合はその変数をモデルから除外する。以上の手順を繰り返して最適な変数の組み合わせを求める。

ステップワイズ法では線形回帰モデルを利用して変数選択を行うが、線形回帰モデルには仮定するモデルにより幾つかのバリエーションが存在する。本

研究では工数見積もりが目的であることを考慮して、負の二項分布の一般化線形回帰モデルを用いた。

3.3 リサンプリング法

3.3.1 変数選択の手順

リサンプリング法を用いた変数選択では、まず説明変数の全ての組み合わせについてリサンプリング法を用いて誤差の推定量を求める。そして最も統計量の小さい説明変数の組み合わせを最適なモデルとして選択するという手順になる。具体的に変数選択の手順を以下に示す。

1. 説明変数のべき集合を求める。

説明変数の集合 $X = \{x_1, x_2, \dots, x_m\}$ のべき集合から空集合を除いた集合 $P(X)$ を求める。 $P(X)$ は説明変数の全ての組み合わせを要素とする。

2. 統計量を求める。

説明変数の組み合わせ $p_i \in P(X) (1 \leq i \leq 2^m - 1)$ の全てに対してリサンプリング法により統計量 s_i を求める。

3. 最適な変数を選択する。

(2) で求めた推定量 $S = \{s_1, s_2, \dots, s_{2^m-1}\}$ の中で最も値の小さい推定量 s_{min} に対応する説明変数の組み合わせ p_{min} を求める。

本研究ではリサンプリング法として leave-one-out 法, bootstrap 法, .632 bootstrap 法を用いる。次節以降でこれらの手法について述べる。

3.3.2 leave-one-out 法

k -fold cross-validation 法において、 $k = N$ (N はデータのサイズ) の場合を特に leave-one-out 法と呼ぶ。工数見積もり手法に関する研究では変数選択に限らずモデル評価にも広く用いられている手法である。以下に手順を示す。

1. サイズ N のデータセット D を i 番目 ($1 \leq i \leq N$) のデータ d_i と $n-1$ 個のデータセット D'_i に分割する。
2. データセット D'_i を用いて学習済みモデル M_i を作成する。
3. 学習モデル M_i にデータ d_i を適用して見積り値 \hat{y}_i を求める。
4. 上記の手順を全ての i について繰り返して N 個の見積り値を求める。
5. N 個の見積り値とそれに対応する実測値から誤差の推定量 s を計算する。

leave-one-out 法は広く用いられているが、データサイズが小さい場合に最良のモデルを選択しない場合があること、データセット中の値の小さな違いが変数選択に大きな影響を及ぼすことなどが指摘されている [3]。後者については k の値を小さくすることが有効な一方で、 k が小さい場合にはモデル学習用のデータのサイズが小さくなり、学習モデルが不完全になるという問題がある。

3.3.3 bootstrap 法 [8]

bootstrap 法は特にデータの少ない場合に有効なリサンプリング法である。計算手順を以下に示す。

1. サイズ N のデータセット D から重複を許して N 個のデータをランダムに抽出し、サイズ N のデータセット D'_i を作成する。また同時に一回も抽出されなかったデータのみを集めてデータセット \bar{D}'_i を作成する。
2. データセット D'_i を用いて学習済みモデル M_i を作成する。
3. データセット \bar{D}'_i を学習済みモデル M_i に適用して見積り値 $\hat{y}_{ij} (1 \leq j \leq \|\bar{D}'_i\|)$ を求める。
4. 見積り値 \hat{y}_{ij} とそれに対応する実測値 y_{ij} を元に誤差の推定量 e_i を求める。

5. 上記の手順を予め定めた回数 R だけ繰り返して誤差の推定量 $E = \{e_1, e_2, \dots, e_R\}$ を求める.
6. E の平均値を最終的な誤差の推定量 s とする.

本研究では、誤差の推定量として残差平方を用い、 $R = 25$ として実験を行った.

3.3.4 .632 bootstrap 法 [8]

.632 bootstrap 法は 3.3.3 節で述べた bootstrap 法の改良版であり、より正確な推定量を求めることができる。以下に手順を示す。

1. bootstrap 法により誤差の推定量を求めて s_{boot} とおく.
2. サイズ N のデータセット D を用いて学習済みモデル M を作成する.
3. データセット D を学習済みモデル M に適用して見積もり値 $\hat{y}_j (1 \leq j \leq N)$ を求める.
4. 見積もり値 \hat{y}_j とそれに対応する実測値 y_j を元に誤差の推定量 s_{app} を求める.
5. 以下の式に当てはめて統計量 s を求める.

$$s = 0.368s_{app} + 0.632s_{boot}$$

4 準備

4.1 工数見積もり手法

1 節で述べたように現在までに様々な工数見積もり手法が提案されているが、本研究における評価実験では analogy-based reasoning (ABR) を用いる。

ABR は Shepperd ら [17] らによって提案された工数見積もり手法である。ABR では説明変数を多次元ベクトルとみなして、過去に収集された各データと見積もり対象から収集されたデータとの間で類似度を求める。そして類似度の高い k 個のデータ中の目的変数の値を基に見積もりを行う。本質的には k -NN 法 [13] と同一の手法であるが、見積もり値を得る最

表 1: 評価実験用データセットの概要

データセット名	説明変数の数	データ数
Desharnais [6]	9	77
Desharnais1 [6]	8	44
Desharnais2 [6]	8	23
Desharnais3 [6]	8	10

終段階の計算方法においてソフトウェア工学の領域知識を活用した linear adjustment という手法も提案されている [18].

本研究では $k = 3$ とし、見積もり値の計算を重みなし平均により行う。

4.2 評価基準

工数見積もり手法の性能評価のために様々な評価基準が提案されている [14]。これらの評価基準はテスト用のデータセットを工数見積もり手法に適用して得られた見積もり値と実際の値から算出される。

変数選択手法の性能が良いほど工数見積もり手法の見積もり精度は高くなると考えられることから、本研究では工数見積もり手法の評価基準を変数選択手法の比較に用いた。

評価基準の中で広く用いられているのが $MMRE$ と $PRED(X)$ である。このうち、 $MMRE$ については、その値が見積もり値に大きく左右されてしまうため、 $MdMMRE$ という評価基準も合わせて考慮すべきであると指摘されている [4]。 $MMRE$, $MdMMRE$, $PRED(X)$ は MRE という評価基準に基づいて算出される。 MRE , $MMRE$, $MdMMRE$, $PRED(X)$ の定義はそれぞれ以下の通りである。

$$MRE = \frac{|(\text{実測値}) - (\text{見積もり値})|}{(\text{見積もり値})}$$

$$MMRE = (\text{MRE の平均値})$$

$$MdMMRE = (\text{MRE の中央値})$$

$$PRED(X) = \frac{(\text{MRE が } X\% \text{ 未満のデータ数})}{(\text{評価に用いたデータ数})}$$

本研究では $PRED(X)$ の X として他の研究でよく利用されている $X = 25$ を用いる。これは、 MRE が 25% 以下であったデータの数の総データ数に対する割合を示す。

5 評価実験

5.1 データセット

表 1 に本研究で使用するデータセットの一覧を示す。これらは全て実際の開発現場で収集されたデータである。

表 1 の Desharnais1, Desharnais2, Desharnais3 は開発環境を考慮してデータセット Desharnais を分割したものである。Desharnais は開発環境を考慮して分割することで見積もり精度が向上することが工数見積もり手法の先行研究 [1, 16] で確認されている。異なる条件のデータセットを用いた検証を行うため、本研究ではこれら 3 つのデータセットを用いた評価実験も同時に行った。このデータ分割ために説明変数を 1 つ用いているため、説明変数は Desharnais よりも 1 つ少ない。

本研究では、評価実験に先立ちデータに対して以下の前処理を行った。

- 欠損値を含むデータの除去
工数見積もり手法ではデータ中の欠損値が精度に影響を及ぼす場合がある。そこで欠損値を含むデータを除去した。表 1 のデータ数は欠損値を除去した後の数である。
- データのスケーリング
ABR では説明変数の単位が異なる場合、そのままでは正確に類似度を求めることが困難となる。そこで他の研究と同様に各変数毎にデータを標準化した。

以上の処理によって得られた 4 つのデータセットに対して評価実験を行った。

5.2 実験方法

本研究では、ABR を工数見積もり手法として 3 節に挙げた変数選択手法の比較評価を行う。比較評価の手順を以下に示す。

1. 過去に収集済みのデータ D より無作為に 50% のデータを抽出し、 D'_i とおく。また、残り 50% のデータを \bar{D}'_i とおく。
2. 変数選択手法毎に D'_i から最適な変数を選択する。
3. D'_i と選択された変数を用いて学習済みモデル M_i を求める。
4. \bar{D}'_i を入力として M_i から予測値 $\hat{y}_{ij} (1 \leq j \leq \|D'\|)$ を求める。
5. 予測値 \hat{y}_{ij} と実測値 y_{ij} から評価基準 $MMRE_i$, $MdMRE_i$, $PRED_i(25)$ を求める。
6. 上記の手順を予め定めた回数 R だけ繰り返す。
7. R 個の $MMRE_i$, $MdMRE_i$, $PRED_i(25)$ のそれぞれについて平均値と標準偏差を求める。

本研究では $R = 20$ として実験を行った。

6 実験結果

実験結果を表 2 から表 5 に示す。なお、Desharnais3 はデータサイズが小さすぎるためステップワイズ法を適用できなかった。よって、表 5 の該当する欄は「-」と記してある。これらの表を観察した結果以下の点が明らかとなった。

- リサンプリング法の中では、632 bootstrap 法が妥当である。
 $MMRE$ 及び $PRED(25)$ の平均値を比較した時、Desharnais3 を用いた実験 (表 5) を除いた全ての実験で leave-one-out 法の性能が最も低い。また、 $MMRE$ 及び $PRED(25)$ の標準偏差も全体的に leave-one-out 法の方が大きい。一方、

表 2: 実験結果 (Desharnais)

	bootstrap		.632		leave-one-out		相関		stepwise	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
<i>MMRE</i>	0.69	0.21	0.69	0.21	0.70	0.26	0.75	0.15	0.61	0.15
<i>MdMRE</i>	0.43	0.09	0.43	0.09	0.44	0.12	0.41	0.06	0.37	0.07
<i>PRED(25)</i>	0.33	0.06	0.33	0.06	0.32	0.07	0.31	0.06	0.35	0.07

表 3: 実験結果 (Desharnais1)

	bootstrap		.632		leave-one-out		相関		stepwise	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
<i>MMRE</i>	0.57	0.14	0.57	0.14	0.58	0.16	0.56	0.15	0.57	0.14
<i>MdMRE</i>	0.40	0.09	0.40	0.09	0.39	0.10	0.41	0.12	0.39	0.10
<i>PRED(25)</i>	0.36	0.07	0.36	0.07	0.33	0.10	0.34	0.13	0.35	0.10

bootstrap 法と .632 bootstrap 法を比べてみると、両手法は同じ変数を選んでおり性能に差がない。しかし、計算時間の差が小さいこと、一般には .632 bootstrap 法の方がより正確に推定量を求めることができると考えられていることを考慮すると、リサンプリング法を用いる場合、.632 bootstrap 法を用いるのが妥当と考えられる。

- 工数見積もり手法から独立した手法はデータの特徴に影響を受ける。

ステップワイズ法について *MMRE* 及び *PRED(25)* を観察すると、データサイズが最も大きい Desharnais では他の手法に比べて良い結果であることがわかる。しかし、よりサイズの小さい他のデータセットの実験結果を比較すると、.632 bootstrap 法とほぼ同程度の結果となることがわかる。また、データサイズが極端に小さい場合には手法の適用そのものが不可能だった。

相関係数を用いた変数選択法について同様に観察すると、データサイズが小さい場合 (Desharnais2 及び Desharnais3) に最も良い結果を示した。しかし、データサイズが最も大きい Desharnais では最も悪い結果となった。

- bootstrap 法は変数選択には不向きである。bootstrap 法は実験全体を通して、最良の結果

を与えることがなかった。このことから、変数選択には不向きな手法だと考えられる。

以上の観察結果から、データサイズが極端に小さい場合を除き、ステップワイズ法による変数選択が最も良い結果を与えると考えられる。

7 まとめ

本研究では工数見積もり手法で用いられるモデル選択手法の性能について比較評価を行った。その結果、変数選択手法としてはステップワイズ法が最も良い結果を与えることを実験により確認した。

しかし、今回の結果は特定の工数見積もり手法及びデータセットによる実験であり、結果の一般性を主張するためには他の工数見積もり手法やデータセットを用いた同様の実験を行う必要がある。

参考文献

- [1] M. Auer, A. Trendowicz, B. Graser, E. Haunschmid, and S. Biffl. Optimal project feature weights in analogy-based cost estimation: Improvement and limitations. *IEEE Transactions on Software Engineering*, 32(2):83–92, 2006.

表 4: 実験結果 (Desharnais2)

	bootstrap		.632		leave-one-out		相関		stepwise	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
<i>MMRE</i>	0.46	0.18	0.46	0.18	0.52	0.20	0.42	0.13	0.46	0.16
<i>MdMRE</i>	0.38	0.15	0.38	0.15	0.36	0.09	0.30	0.08	0.35	0.17
<i>PRED(25)</i>	0.39	0.12	0.39	0.12	0.36	0.13	0.43	0.12	0.40	0.12

表 5: 実験結果 (Desharnais3)

	bootstrap		.632		leave-one-out		相関		stepwise	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
<i>MMRE</i>	0.57	0.21	0.57	0.21	0.55	0.21	0.53	0.23	-	-
<i>MdMRE</i>	0.52	0.25	0.52	0.25	0.50	0.22	0.43	0.24	-	-
<i>PRED(25)</i>	0.27	0.22	0.27	0.22	0.29	0.21	0.35	0.21	-	-

- [2] B. W. Boehm. *Software Engineering Economics*. Prentice-Hall, 1981.
- [3] L. Breiman. Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24:2350–2383, 1996.
- [4] L. C. Briand, T. Langley, and I. Wiecek. A replicated assessment and comparison of common software cost modeling techniques. In *Proc. of 22nd International Conference on Software Engineering*, pages 377–386, 2000.
- [5] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information Theoretic Approach*. Springer, 2002.
- [6] J. M. Desharnais. Analyse statistique de la productive des projets informatique a partie de la technique des point des fonction. Master's thesis, Univ. of Montreal, 1989.
- [7] J. Dolado. On the problem of the software cost estimat. *Information and Software Technology*, 43(1):61–72, January 2001.
- [8] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.
- [9] T. M. Khoshgoftaar and E. B. Allen. Multivariate assessment of complex software systems: A comparative study. In *Proc. of 1st International Conference on Engineering of Complex Computer Systems(ICECCS'95)*, pages 389–396, 1995.
- [10] C. L. Martín, J. L. Pasquier, C. M. Yáñez, and A. T. Gutiérrez. Software development effort estimation using fuzzy logic: A case study. In *Proc. of the 6th Mexican International Conference on Computer Science (ENC'05)*, pages 113–120, 2005.
- [11] E. Mendes, N. Mosley, and S. Counsell. Do adaptation rules improve web cost estimaton? In *Proc. of the 14th ACM conference on Hypertext and Hypermedia*, pages 173–183, 2003.
- [12] E. Mendes, N. Mosley, and S. Counsell. Early web size measures and effort prediction for web costimation. In *Proc. of 9th International Software Metrics Symposium*, pages 18–29, 2003.
- [13] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [14] I. Myrtveit, E. Stensrud, and M. Shepperd. Reliability and validity in comparative studies

of software prediction models. *IEEE Trans. on Software Engineering*, 31(5):380–391, 2005.

- [15] M. Shepperd. ArchANGEL. <http://dec.bournemouth.ac.uk/ESERG/ANGEL/>.
- [16] M. Shepperd and C. Schofield. Estimating software project effort using analogies. In *IEEE Trans. on Software Engineering*, pages 736–743, 1997.
- [17] M. Shepperd, C. Schofield, and B. Kitchenham. Effort estimation using analogy. In *Proc. of 18th International Conference on Software Engineering*, pages 170–178, 1996.
- [18] I. Wieczorek and M. Ruhe. How valuable is company-specific data compared to multi-company data for software cost estimation? In *Proc. of the 8th IEEE Symposium on Software Metrics (METRICS'02)*, pages 237–246, 2002.