

# 類似性に基づくソフトウェア開発工数見積りにおける外れ値除去法の比較

角田 雅照<sup>†</sup>      門田 暁人<sup>†</sup>      渡邊 瑞穂<sup>†\*</sup>      柿元 健<sup>††</sup>  
 松本 健一<sup>†</sup>

## Comparison of Outlier Deletion Methods for Analogy Based Software Effort Estimation

Masateru TSUNODA<sup>†</sup>, Akito MONDEN<sup>†</sup>, Mizuho WATANABE<sup>†\*</sup>,  
 Takeshi KAKIMOTO<sup>††</sup>, and Ken-ichi MATSUMOTO<sup>†</sup>

あらまし 本論文では、類似性に基づくソフトウェア開発工数見積り方法（Analogy 法）における外れ値除去法の効果を、三つのプロジェクトデータセット（ISBSG データ、Kitchenham データ、Desharnais データ）を用いて実験的に比較する。比較対象の外れ値除去法は、これまで提案されている 4 種類の外れ値除去法（*k*-means 法を用いた除去法、LTS を用いた除去法、Cook の距離を用いた除去法、Mantel 相関を用いた除去法）と、本論文で新たに提案する除去法である。提案方法は、Analogy 法の特性を考慮して、類似プロジェクトにおいて工数（生産性）が極端に異なる類似プロジェクトを外れ値とみなし、見積り時の計算から除外する。実験の結果、提案方法は比較した外れ値除去法の中で平均的に最も高い見積り精度を示し、どのデータセットを用いた場合でも見積り精度が大きく低下することなく、*ABRE* (Absolute Balanced Relative Error) 平均値で最大 28.8%の改善が見られた。

キーワード case based reasoning, 工数予測, 異常値, プロジェクト管理, 生産性

### 1. ま え が き

ソフトウェア開発プロジェクトを成功に導くために、開発工数を高い精度で見積もることは非常に重要であり、これまで、様々な定量的工数見積り手法が提案されてきた。近年、類似性に基づく工数見積り方法（Analogy 法）[27] に対する注目が高まっており、Analogy 法に関する様々な提案や適用事例が報告されている [7], [8], [19], [30], [31]。Analogy 法は、過去プロジェクトの実績データセット（開発規模や開発工数などが記録されたデータセット。以降、単にデータセットとする）から、見積り対象のプロジェクトと類

似したプロジェクトを選定し、それらの実績工数に基づいて工数を見積もる。Analogy 法による見積りは、見積りの根拠となる類似プロジェクトが提示されるため、（プロジェクトマネージャなど）見積りを行う者にとって直観的に理解しやすいという利点がある [31]。また、Analogy 法では、重回帰分析などのようにデータセット全体に対してただ一つの見積り式を構築するのではなく、見積り対象プロジェクトごとに個別に見積り式を構築するため、プロジェクトの個別性をより強く反映した見積りが行えるという特長がある。

ただし、データセットには、見積りの根拠データとして用いるべきではないプロジェクトが含まれている場合がある [26]。例えば、例外的な分量の作業手戻りが発生したプロジェクトの場合、同一規模の他のプロジェクトに比べ、開発工数が極端に大きくなる。また、工数の収集・集計時にミスが発生したプロジェクトの場合、実際の工数と異なる工数が記録される。これらのプロジェクトは見積り精度の低下を招くため、見積り時に除外する必要がある。しかしほとんどの場合、

<sup>†</sup> 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 and Technology,  
 Nara Institute of Science and Technology, 8916-5 Takayama-  
 cho, Ikoma-shi, 630-0192 Japan

<sup>††</sup> 香川高等専門学校電気情報工学科, 高松市  
 Department of Electrical and Computer Engineering,  
 Kagawa National College of Technology, 355 Chokushicho,  
 Takamatsu-shi, 761-8058 Japan

\* 現在, 日本アイ・ピー・エム株式会社

データセットにはこのような個別のプロジェクトの内情までは記録されていないため、これらのプロジェクトを特定することは難しい。また、仮にデータ収集時にこういった内情を知り得た場合でも、どの程度の分量の作業手戻りが発生したプロジェクトを除くべきかといった規準を設定することは容易ではない。

このため、統計的手法に基づく外れ値除去法を用いて、これらのプロジェクトを除去することが行われる。外れ値除去法とは、特定の変数の値が極端に大きなプロジェクトや、(工数や規模、開発期間などの)変数の値の組合せが他のプロジェクトと大きく異なるプロジェクトを外れ値とみなし、データセットから除去する方法である。重回帰分析の場合、Cookの距離[29]を用いて外れ値を除去することが一般的である。Cookの距離を用いた除去法では、重回帰分析を行い、モデルの係数に大きな影響を与えるケースを外れ値とみなす。Cookの距離を用いた除去法以外にも、工数見積り時にデータセットから外れ値を除去する方法がいくつか提案されている[7],[26]。ただし、これらの外れ値除去法をAnalogy法に適用した事例はほとんどなく、更にこれらの外れ値除去法を適用した場合の効果もこれまで比較されておらず、いずれの外れ値除去法がAnalogy法に適しているのかは明らかではない。

そこで本論文では、Analogy法に複数の外れ値除去法を適用し、それぞれの効果を比較する。実験では、これまで提案されている4種類の外れ値除去法( $k$ -means法を用いた除去法[26], LTSを用いた除去法[26], Cookの距離を用いた除去法[29], Mantel相関を用いた除去法[7])を、3種類のデータセット(ISBSGデータ[6], Kitchenhamデータ[11], Desharnaisデータ[1])に適用して外れ値を除去し、工数見積りを行い、見積り精度を比較する。これらはソフトウェア開発企業で収集されたデータであり、無償若しくは有償にて利用可能で、かつ多くの論文で工数見積りモデル構築のために使われており、プロジェクト数が多いという特徴をもつ。

更に本論文では、Analogy法の特性を考慮した外れ値除去法を提案し、従来の外れ値除去法と効果を比較する。Analogy法では、類似プロジェクトを複数個抽出し、それらの実績工数に基づいて見積り工数を計算するが、類似プロジェクトの実績工数(3.で後述する規模補正法を用いる場合、生産性)がばらついている場合、見積り精度が低下することが指摘されている[8],[24]。そこで本論文では、類似プロジェクトに

おいて実績工数(生産性)が大きく異なるプロジェクトを外れ値とみなし、工数見積りの計算から除外する。具体的には、類似プロジェクト抽出後に、ロバストZ-score法[22]で標準化した実績工数がしきい値を上回るプロジェクトを外れ値とみなす。

以降、2.ではAnalogy法について説明し、3.で従来の外れ値除去法、4.で提案方法について説明する。5.では評価実験について説明し、6.で実験結果、7.で考察を述べる。最後に8.でまとめと今後の課題を述べる。

## 2. 類似性に基づく工数見積り

Analogy法では、表1に示す $m \times n$ 行列で表されるデータセットを入力として用いる。表において、 $Proj_i$ は $i$ 番目のプロジェクト、 $Metric_j$ は $j$ 番目の変数、 $x_{ij}$ は $Proj_i$ の $Metric_j$ の値、 $fp_i$ は $Proj_i$ の(ファンクションポイントなどの)開発規模、 $y_i$ は $Proj_i$ の工数を示す。ここで $Proj_a$ を見積り対象のプロジェクト、 $\hat{y}_a$ を $y_a$ の見積り値とする。このとき、Analogy法による工数見積りは以下の手順で行われる。

(1) 各変数 $Metric_j$ の値域を $[0, 1]$ に正規化し、値域(単位)の違いが類似度計算に影響を与えないようにする(正規化を行わない場合、値域の大きな変数が類似度に大きな影響を与えてしまう)。 $j$ 番目の変数 $Metric_j$ の最大値を $\max(Metric_j)$ 、最小値を $\min(Metric_j)$ としたとき、 $v_{ij}$ の正規化後の値 $v'_{ij}$ は次式により求める。

$$v'_{ij} = \frac{v_{ij} - \min(Metric_j)}{\max(Metric_j) - \min(Metric_j)} \quad (1)$$

(2) データセットから、見積り対象のプロジェクト $Proj_a$ と類似した過去プロジェクト $Proj_i$ を見つけるため、二つのプロジェクト間の「距離」を類似度として用いる。プロジェクトを $m$ 個の変数で表現し、 $m$ 次元空間におけるユークリッド距離を求めることが一般的である[30]。この距離が短いほど、二つのプロジェクトは類似していることを表す。 $Proj_a$ と $Proj_i$

表1 Analogy法で用いるデータセット  
Table 1 Dataset used by analogy based effort estimation.

	Effort	Size	$Metric_1$	...	$Metric_j$	...	$Metric_m$
$Proj_1$	$y_1$	$fp_{11}$	$x_{11}$	...	$x_{1j}$	...	$x_{1m}$
...	...	...	...	...	...	...	...
$Proj_i$	$y_i$	$fp_{i1}$	$x_{i1}$	...	$x_{ij}$	...	$x_{im}$
...	...	...	...	...	...	...	...
$Proj_n$	$y_n$	$fp_{n1}$	$x_{n1}$	...	$x_{nj}$	...	$x_{nm}$

の距離  $\text{Dist}(Proj_a, Proj_i)$  は次式により求める。

$$\text{Dist}(Proj_a, Proj_i) = \sqrt{\sum_{h=1}^m (v'_{ah} - v'_{ih})^2} \quad (2)$$

(3) 類似度が高い  $k$  個の過去プロジェクト  $Proj_i$  の実績工数  $y_i$  より、見積り対象プロジェクト  $Proj_a$  の見積り工数  $\hat{y}_a$  を計算する。一般的には、類似プロジェクトの実績工数の平均値を見積り工数とする計算法が用いられるが、本論文では複数の適用事例においてより高い見積り精度が報告されている [8], [19], [31], 類似プロジェクトの実績工数を規模補正した値を見積り工数とする計算法 (以降、規模補正法と呼ぶ) を用いる。規模補正法では、見積り対象プロジェクトの規模が類似プロジェクトの規模の  $s$  倍 ( $s$  は 0 より大きい実数) であれば、工数も同様に  $s$  倍になっているという仮定に基づき、類似プロジェクトの規模と見積り対象のプロジェクトの規模の比に基づいて類似プロジェクトの工数を補正している。  $Proj_i$  の開発規模  $fp_i$  と  $Proj_a$  の開発規模  $fp_a$  を用いて、  $Proj_i$  の補正工数  $adjy_i$  は式 (3) により、見積り工数  $\hat{y}_a$  は式 (4) により計算する。ここで、  $Simprojects$  は、  $Proj_a$  と類似度の高い上位  $k$  個のプロジェクトの集合を表す。

$$adjy_i = y_i \times \frac{fp_a}{fp_i} \quad (3)$$

$$\hat{y}_a = \frac{\sum_{h \in Simprojects} adjy_h}{k} \quad (4)$$

### 3. 従来の外れ値除去法

外れ値除去法とは、データサンプル (データセット) 中のケース (プロジェクト) が外れ値であるか検証し、外れ値と判断されたケースをデータサンプルから除去する処理である。これまで、ソフトウェア開発工数見積りにおいて、重回帰分析適用時に Cook の距離を用いて外れ値を除去することは行われていたが ([18] など)、外れ値除去法の効果を確認した研究は少なく [26], Analogy 法に適した外れ値除去法もほとんど存在しない [7]。

Seo ら [26] は、ソフトウェア開発工数見積り時に LTS 法を用いた除去法と  $k$ -means 法を用いた除去法を適用することを提案し、重回帰分析、ニューラルネットワーク、ベイジアンネットワークそれぞれによる見積り時に外れ値除去法を適用し、外れ値除去の効果を確認している。ただし、Analogy 法に適用した場合の

効果は確かめていない。

Keung ら [7] は、Analogy 法への適用を前提とした、Mantel 相関を用いた除去法を提案している。Mantel 相関を用いた除去法により、どのプロジェクトが除去されるかは分析しているが、Mantel 相関を用いた除去法により、見積り精度がどの程度改善するかは確かめていない。以降では、それぞれの外れ値除去法について詳述する。

#### 3.1 $k$ -means 法を用いた除去法

$k$ -means 法を用いた除去法は、 $k$ -means 法によるクラスタリング結果に基づいて外れ値を特定し、除去する手法である [26]。  $k$ -means 法は、データを  $k$  個のクラスタに分け、互いに似ているケースを同じクラスタに割り当てる。  $k$  は初期値として与える必要があるが、  $k$ -means 法を用いた除去法では、シルエット値の平均値が最小となる  $k$  を採用する。シルエット値は、各ケースが所属しているクラスタが適切かどうかを判断する指標であり、あるケースが所属しているクラスタとそれ以外のクラスタとの距離に基づいて計算される。シルエット値の値域は  $[-1, 1]$  であり、1 に近いほどそのケースは適切なクラスタに割り当てられているといえる。  $k$ -means 法を用いた除去法では、シルエット値が 0 未満のケースと、2 個以下の個体しか存在しないクラスタに含まれるケースを外れ値とみなし、データセットから除外する。なお、  $k$ -means 法を適用する前に、あらかじめ式 (1) によりデータを正規化しておく。

#### 3.2 LTS を用いた除去法

LTS (Least trimmed squares) を用いた除去法は、重回帰分析を行い、残差の大きいケースを外れ値とみなす手法である [26]。 LTS 法はロバストな重回帰分析の手法である。通常重回帰分析では残差二乗和が最小になるようにモデルを構築するのに対し、LTS 法では残差を小さい順に並べた場合の  $h$  番目 ( $n/2 < h < n$ ,  $n$  はデータ件数) までの残差を用いて、残差二乗和が最小になるようにモデルを構築する。 LTS を用いた除去法では、  $h = 0.75n$  とし、残差の大きい  $0.25n$  個のケースを外れ値とみなし、データセットから除外する。

#### 3.3 Cook の距離を用いた除去法

Cook の距離を用いた除去法は、重回帰分析を行い、モデルの係数に大きな影響を与えるケースを外れ値とみなす。 Cook の距離は、あるケースをモデル推定の計算から除外した場合に、全てのケースの残差がどの程度変化するかを示す距離であり、Cook の距離が  $4/n$  ( $n$  はデータ件数) 以上となるケースを外れ値と

みなし [29], データセットから除外する. Cook の距離が大きい場合, そのケースをモデル構築の計算から除外したことにより, 偏回帰係数が大きく変化したことを示す. Cook の距離を用いた除去法は, 重回帰分析適用時に外れ値を除去する方法であるが, 工数見積りの研究における外れ値除去法としては最も一般的であるため, 本論文では Analogy 法に適用し, その効果を確かめる.

### 3.4 Mantel 相関を用いた除去法

Mantel 相関を用いた除去法は, 説明変数の値が類似しているにもかかわらず, 目的変数の値が類似していないケースを外れ値とみなす手法であり [7], Analogy 法に適用することを前提としている. 通常的相关係数 (Pearson の相関係数など) は二つの変数の関連の強さを表すのに対し, Mantel 相関は (説明変数のセットと目的変数など) 二つの変数セット間の関連の強さを表す. Mantel 相関により「プロジェクトの開発規模や開発期間 (説明変数の値) が似ていると, 開発工数 (目的変数の値) も似ているかどうか」を調べることができる. Mantel 相関は, 各プロジェクトのペアについて, 説明変数に基づくユークリッド距離と目的変数に基づくユークリッド距離を計算し, それらの相関係数を計算することにより求められる.

Mantel 相関を用いた除去法では, 以下の手順により外れ値を特定する.

- (1) 全てのプロジェクトについて,  $i$  番目のプロジェクトを除外した Mantel 相関  $r_i$  を計算する.
- (2)  $r_i$  の平均値  $\bar{r}$ , 標準偏差  $rs$  を計算する.
- (3)  $i$  番目のプロジェクトが Mantel 相関の平均値に与える影響の大きさを  $t$  と呼び,  $r_i - \bar{r}$  により求める.
- (4)  $t$  を標準偏差  $rs$  で割り, 標準化した値が 4 (Keung ら [7] が推奨する値.  $4\sigma$ ) を上回るプロジェクトを外れ値とみなし, データセットから除外する.

## 4. 提案する外れ値除去法

提案する外れ値除去法は, 見積り対象プロジェクトと類似したプロジェクトの中で, 工数が大きく異なるプロジェクトを外れ値とみなす. Analogy 法は「類似するプロジェクトは, 工数も互いに似た値をとるであろう」という前提に基づいて見積もっており, 提案方法は Analogy 法の前提に当てはまらないプロジェクトを外れ値とみなしている. Analogy 法では類似プロジェクトの実績工数がばらついている場合, 見積り精

度が低下することが指摘されており [24], 提案法はこの問題点に対応している.

規模補正法を用いる場合, 提案方法は生産性に基づいて外れ値を除去することになる. 式 (3) を見ると, 類似プロジェクトの補正工数  $adjy_i$  は生産性の逆数  $y_i/fp_i$  ( $y_i$  は実績工数,  $fp_i$  は開発規模) に見積り対象プロジェクトの開発規模  $fp_a$  を乗じており, ( $adjy_i$  計算時には)  $fp_a$  は定数とみなされるため, 補正工数  $adjy_i$  に基づいて外れ値を除去することは, 生産性に基づいて外れ値を除去することと同じとなる.

提案方法では, 類似プロジェクトの実績工数 (規模補正法を用いる場合は補正工数, すなわち生産性) の中央値と比較して, 大きく異なるプロジェクトを外れ値とみなす. ただし, 類似プロジェクトの実績工数の分散が大きいと中央値との差も大きくなるため, 実績工数を標準化し, しきい値を超えるプロジェクトを外れ値とみなす. 一般に, 標準化には正規分布を前提とする Z-score 法が用いられるが, 類似プロジェクトの実績工数は必ずしも正規分布に従っているとは限らない. そこで提案方法では, 正規分布を前提としないロバスト Z-score 法 [22] を用いて標準化する. ロバスト Z-score 法では, 平均値の代わりに中央値を, 標準偏差の代わりに四分位範囲を用いる. より詳細には, 類似度が高い  $k$  個の過去プロジェクトから, 以下の手順で外れ値を除去する.

- (1) 類似プロジェクトの工数  $y_i$  の中央値  $M$  と四分位範囲  $IQR$  を計算する. 規模補正法を用いる場合,  $y_i$  は類似プロジェクトの実績工数ではなく, 規模補正した工数とする.
- (2) 以下の式 (ロバスト Z-score 法 [22]) により, 標準化された工数  $y'_i$  を計算する.

$$y'_i = \frac{y_i - M}{NIQR} \quad (5)$$

ここで  $NIQR = 0.7413 \cdot IQR$  であり, 正規分布の標準偏差に該当する (平均値 0, 標準偏差 1 の正規分布の  $IQR$  は 1.34898 であり,  $IQR$  に 1.34898 の逆数を乗じることにより, 標準偏差と等しくなる).

(3)  $y'_i$  の絶対値がしきい値  $th$  を上回る (中央値  $M$  から標準偏差に該当する  $NIQR$  の  $th$  倍離れている) プロジェクトを外れ値とみなし, 見積り工数計算から除外する. 見積り工数  $\hat{y}_a$  は以下の式により計算する (規模補正法を用いる場合,  $y_h$  は規模補正した工数となる). ここで,  $Elimprojects$  は,  $Proja$  と類似度の高い上位  $k$  個のプロジェクトから,  $d$  個の外れ値

プロジェクト除外した集合を表す。

$$\hat{y}_a = \frac{\sum_{h \in E_{lim} projects} y_h}{k - d} \quad (6)$$

本論文ではしきい値  $th$  を 1.65 (標準正規分布の片側 5%点にあたる) とした。

提案方法は、開発現場における (人手による) 類推見積り法において、外れ値を除去する方法と定性的には類似しているといえ、その方法を定量的に実施できるようにしたとみなすこともできる。また、他の外れ値除去法と比べるとシンプルであることから、開発現場に受け入れられやすいという副次的効果が期待される。提案手法がシンプルである理由は、類似プロジェクト抽出後に適用することが前提となっているためである。類似プロジェクト同士は説明変数の値が類似しているため、従来の外れ値除去法のように説明変数を考慮に入れる必要性は低くなる。そのため、比較的簡便な方法により、目的変数の値のみに着目して外れ値を除去することができる。

## 5. 評価実験

実験により、従来の外れ値除去法と提案する外れ値除去法の性能を評価した。実験では、ISBSG データ [6]、Kitchenham データ [11]、Desharnais データ [1] の三つのデータセットを用いて、Analogy 法により開発工数を見積もった。見積り時に従来の外れ値除去法と提案する外れ値除去法をそれぞれ適用し、精度を比較することにより、それぞれの外れ値除去法の性能を評価した。見積り時期はプロジェクトの計画完了時であると想定し、その時点で値が明確になっている変数のみを説明変数として用いた。

外れ値を除去してから Analogy 法で見積りを行う場合と、類似プロジェクト抽出後に外れ値を除去して見積りを行う場合の性能についてそれぞれ評価した。ただし、 $k$ -means 法を用いた除去法では、2 個以下の個体しか存在しないクラスタに含まれるケースを外れ値とみなすと、ほとんどのプロジェクトが外れ値となるため、シルエット値が 0 未満のプロジェクトのみを除去対象とした。

類似プロジェクト抽出後に Cook の距離を用いた除去法や LTS を用いた除去法を適用する場合、(未調整、または調整済) FP のみを説明変数とした。これは、(これらの除去法で使用される) 重回帰分析を適切に行うためには、一般に説明変数の個数の 5 倍 [28]、ま

たは 5~10 倍程度のデータ件数が必要である [12] とされており、説明変数を絞り込んだ方がより適切に重回帰分析が行われると考えられるためである。

### 5.1 実験に用いたデータセット

ISBSG データは、International Software Benchmark Standard Group (ISBSG) が収集した 20 カ国のソフトウェア開発企業のデータである [6]。本論文ではリリース 9 のデータを用いており、1989 年から 2004 年までの 3026 件のプロジェクトについて、99 種類の変数が記録されている。ただし、ISBSG データにはデータの信頼性が低いプロジェクトなどが含まれており、これらを分析から除外する必要がある。そこで ISBSG データを用いて工数見積りの研究を行っている Lokan ら [17] の条件 (データの品質が A または B、FP が IFPUG 法で計測されているなど) に従ってデータを抽出した。更に欠損値 (変数に値が記録されていないこと) が含まれているプロジェクトを除外した。593 件を実験に用いた。また、説明変数についても Lokan らと同じ四つの変数 (未調整 FP、開発種別、言語種別、プラットフォーム) を用いた。開発種別、言語種別、プラットフォームはカテゴリ変数のため、ダミー変数化して用いた。

Kitchenham データは、Kitchenham が 2002 年に文献の中で公開した、あるソフトウェア開発企業のデータである [11]。データセットに含まれる 145 件のプロジェクトのうち、欠損値が含まれるプロジェクトを除いた 135 件を用いた。説明変数は、工数見積りに用いるには不適切な変数 (プロジェクト管理者による工数の見積り値など) と開発期間を除いた二つの変数 (調整済み FP、開発種別) である。このうち開発種別はカテゴリ変数のため、ダミー変数化して用いた。

Desharnais データ [1] は、Desharnais によって収集されたカナダのソフトウェア開発企業における 80 年代のデータセットである [4]。データセットに含まれる 81 件のプロジェクトのうち、欠損値が含まれるプロジェクトを除いた 77 件を用いた。説明変数は、開発年度と調整済み FP (ファンクションポイント)、開発期間を除いた七つの変数 (開発チームの経験年数、プロジェクトマネージャの経験年数、トランザクション数、未調整 FP、調整係数、開発言語、エンティティ数) である。このうち開発言語はカテゴリ変数のため、ダミー変数化して用いた。Cook の距離を用いた除去法と LTS を用いた除去法を適用する際には、トランザクション数とエンティティ数を説明変数から除

外した。これは、未調整 FP = トランザクション数 + エンティティ数という関係があり、説明変数として用いることは、多重共線性の点から不適切であるためである。

## 5.2 評価指標

工数見積り精度の評価指標として、*AE* (Absolute Error), *MRE* (Magnitude of Relative Error) [3], *MER* (Magnitude of Error Relative to the estimate) [10], *ABRE* (Absolute Balanced Relative Error) [20], *BRE* (Balanced Relative Error) [20] の五つの指標の平均値と中央値を用いた<sup>(注1)</sup>。工数の実測値を  $x$ , 見積り値を  $\hat{x}$  とするとき、それぞれの指標は以下の式により求められる。

$$AE = |x - \hat{x}| \quad (7)$$

$$MRE = \frac{|x - \hat{x}|}{x} \quad (8)$$

$$MER = \frac{|x - \hat{x}|}{\hat{x}} \quad (9)$$

$$ABRE = \begin{cases} \frac{|\hat{x} - x|}{x}, \hat{x} - x \geq 0 \\ \frac{|\hat{x} - x|}{\hat{x}}, \hat{x} - x < 0 \end{cases} \quad (10)$$

$$BRE = \begin{cases} \frac{(\hat{x} - x)}{x}, \hat{x} - x \geq 0 \\ \frac{(\hat{x} - x)}{\hat{x}}, \hat{x} - x < 0 \end{cases} \quad (11)$$

*BRE* 以外の指標は、値が小さいほど工数見積りの精度が高いことを示す。*BRE* は 0 に近いほど工数見積りに偏りが少ないことを示す (正の値の場合、過大見積りに偏っていることを示し、負の値の場合は過小見積りに偏っていることを示す)。直感的には *MRE* は実測値との相対誤差、*MER* は見積り値との相対誤差であるといえる。ただし、*MRE* と *MER* は過大見積りと過小見積りに対し、アンバランスな評価になるという問題がある [2], [16]。(見積り工数が 0 以上の) 過小見積りの場合、*MRE* は最大でも 1 にしかならない (例えば実測値が 1000 人時、見積り値が 0 人時の場合、*MRE* は 1 となる)。逆に過大見積りの場合、*MER* は最大でも 1 未満としかならない。そこで本論文では、*MRE*, *MER* に加え、過大見積りと過小見積りをバランスよく評価する指標 [21] である *ABRE* を評価指標に用いた。また、外れ値除去によって工数見積りの偏りが変化するかどうかを確かめるために *BRE* を用いた。

## 5.3 実験手順

それぞれの外れ値除去法を適用してソフトウェア開

発工数を見積もる実験を行い、外れ値除去法の性能比較を行った。実験では、工数見積りモデルの研究において *K-fold cross validation* とともに最も一般的な方法である [23], *holdout* 法 [5] を用いた。*holdout* 法を適用する場合、フィットデータのサイズが小さくなり、精度に悪影響を与えないか注意する必要がある。工数見積りの実験において信頼できる結果を得るためには、20 件以上のフィットデータが存在することが理想的であるとされているが [9], 最もデータ件数が少ない *Desharnais* データに *holdout* 法を適用しても、39 件を学習データとして用いることができるため、悪影響はないと判断した。

類似プロジェクト抽出前に *Cook* の距離を用いた除去法と *LTS* を用いた除去法 (重回帰分析) を適用する場合、前述のように説明変数の個数の 5 倍、または 5~10 倍程度のデータ件数が必要となるが、データ数と比較して (ダミー変数を含めた) 説明変数が最も多い *Desharnais* データにおいても、説明変数の個数の 6.5 倍にあたる 39 件のデータがラーニングデータに含まれており、問題はないと考えられる。

外れ値を除去してから *Analogy* 法で見積りを行う場合、以下の手順に従って実験を行った (図 1)。

(1) データを無作為に半数ずつに分け、一方をフィットデータ、もう一方をテストデータとする。フィットデータは見積りの根拠とするデータセット (過去プロジェクトに該当する)、テストデータは見積り対象とするデータセット (開発中のプロジェクトに該当する) である。

(2) 外れ値除去法を適用してフィットデータから外れ値を除去した後、フィットデータから各フィット

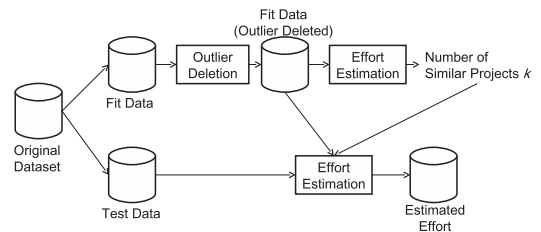


図 1 実験手順 (類似プロジェクト抽出前に外れ値を除去する場合)

Fig. 1 Experimental procedure (Outliers are deleted before selecting similar projects).

(注1) : *ABRE* と *BRE* の平均値は、それぞれ文献 [20] の *ARE*, *RE* に該当する。本論文では、他の評価指標の定義と合わせるために式変形を行い、用語を変更した。

データの類似プロジェクトを抽出し、それらを用いて対象とするフィットデータのプロジェクトの工数を（未知のものとして）見積もる。類似プロジェクト数  $k$  を 1 から 20 まで変化させて見積りを行い、それぞれの場合で残差二乗和を計算し、残差二乗和が最も小さくなった  $k$  を採用する。

(3) 手順 (2) で外れ値を除去したフィットデータから各テストデータの類似プロジェクトを抽出し、それらを用いて対象とするテストデータのプロジェクトの工数を（未知のものとして）見積もる。類似プロジェクト数は手順 (2) で採用した値を用いる。

(4) テストデータの工数実測値と見積り値から、各評価指標を計算する。

(5) (1)～(4) の手順を 10 回繰り返す (10 組のフィットデータ、テストデータ、各評価指標が得られる)。

類似プロジェクト抽出後に外れ値を除去して見積りを行う場合、以下の手順に従って実験を行った (図 2)。

(1) データを無作為に半数ずつに分け、一方をフィットデータ、もう一方をテストデータとする。

(2) フィットデータから各フィットデータの類似プロジェクトを抽出し、類似プロジェクトに外れ値除去法を適用して外れ値を除去してから、それらを用いて対象とするフィットデータのプロジェクトの工数を（未知のものとして）見積もる。類似プロジェクト数  $k$  を 1 から 20 まで変化させて見積りを行い ( $k < 3$  など、 $k$  が小さすぎて外れ値除去法が適用できないときは、外れ値を除去せずに見積もった)、それぞれの場合で残差二乗和を計算し、残差二乗和が最も小さくなった  $k$  を採用する。

(3) フィットデータから各テストデータの類似プロジェクトを抽出し、類似プロジェクトに外れ値除去法を適用して外れ値を除去してから、それらを用いて対象とするテストデータのプロジェクトの工数を（未

知のものとして）見積もる。類似プロジェクト数は手順 (2) で採用した値を用いる。

(4) テストデータの工数実測値と見積り値から、各評価指標を計算する。

(5) (1)～(4) の手順を 10 回繰り返す。

なお、手順 (2) における残差二乗和とは、 $AE$  の二乗を合計したものであり、見積りモデルを選択する際に広く用いられている [16]。また、手順 (2) においてフィットデータが 20 件未満の場合は、 $k$  の最大値はフィットデータに含まれるプロジェクト件数とした。

## 6. 実験結果

表 2～表 4 に、各データセットにおける外れ値除去法を適用しない場合 (Analogy 法だけを用いる場合) の評価指標、適用しない場合と適用した場合の評価指標の差、及び除去率 (除去プロジェクト件数 / 全プロジェクト件数。類似プロジェクト抽出後に除去した場合、除外プロジェクト数 / 類似プロジェクト数の平均値) を示す。評価指標は 10 個のテストデータから得られた値を平均したものである。差が負となっているものは、外れ値除去法を適用することにより評価指標が悪化したことを示す。差の検定ではノンパラメトリックな検定である Wilcoxon の符号付順位と検定を用いた<sup>(注2)</sup> (各評価指標は正規分布しているとはいえ、また同じフィットデータに対してそれぞれの外れ値除去法を適用した結果を比較するため)。有意水準は 5% とし、有意差があった場合は斜体で表している。

### 6.1 ISBSG データに外れ値除去法を適用した場合 (類似プロジェクト抽出後に適用)

**提案手法**  $MER$  平均値は悪化していたが、 $ABRE$  の平均値で 28.83%、中央値で 4.86% 改善しており、更に  $AE$  の平均値や中央値など、その他の評価指標も改善していたことから ( $MER$  中央値と  $ABRE$  中央値以外で有意差があった)、外れ値除去の効果があったといえる。

**Mantel 相関を用いた除去法** プロジェクトが全く除外されおらず、外れ値除去が機能していなかった。

**$k$ -means 法を用いた除去法** 10 個中 8 個の評価指標が改善し、改善しなかった評価指標についても悪化はごくわずかであった。ただし、見積り精度の改善の程度は  $ABRE$  の平均値で 2.63%、中央値で 2.12% で

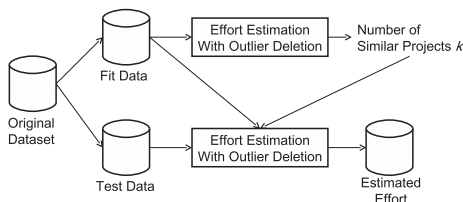


図 2 実験手順 (類似プロジェクト抽出後に外れ値を除去する場合)

Fig. 2 Experimental procedure (Outliers are deleted after selecting similar projects).

(注2)：この検定では、外れ値除去法未適用時と適用時を比較して、いくつかのテストサブセットで評価指標が上回っている (または下回っている) にかよって  $p$  値が決定され、差の大きさには影響を受けない。

表 2 外れ値除去法を適用した場合の評価指標 (ISBSG データ)  
Table 2 Evaluation criteria after outlier deletion (ISBSG data).

外れ値除去法	AE 平均値	AE 中央値	MRE 平均値	MRE 中央値	MER 平均値	MER 中央値	ABRE 平均値	ABRE 中央値	BRE 平均値	BRE 中央値	除去率	
未適用	3680	1607	166.93%	59.59%	91.27%	58.05%	210.36%	91.69%	-87.24%	-21.38%	0.00%	
抽出後 適用	提案方法	602	285	47.80%	5.60%	-16.90%	0.29%	28.83%	4.86%	75.89%	19.17%	15.41%
	Mantel 相関	0	0	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	k-means 法	61	-6	2.46%	0.10%	0.52%	1.58%	2.63%	2.12%	1.36%	-1.49%	0.74%
	Cook の距離	158	42	9.29%	1.50%	0.90%	0.94%	9.52%	2.22%	9.63%	3.39%	3.91%
	LTS	477	239	37.25%	3.63%	-11.42%	-0.02%	24.18%	4.30%	57.61%	20.32%	21.56%
抽出前 適用	提案方法	-379	-83	-22.08%	-3.31%	12.55%	2.37%	-9.39%	0.93%	-42.34%	-16.51%	15.12%
	Mantel 相関	131	4	4.53%	-0.82%	-1.34%	-0.43%	3.19%	-0.32%	6.84%	1.35%	1.21%
	k-means 法	-99	-122	0.68%	-3.20%	-20.04%	-2.16%	-17.86%	-10.07%	24.01%	7.06%	2.79%
	Cook の距離	309	-4	12.24%	-2.91%	-19.08%	-0.75%	-5.99%	-7.37%	34.92%	7.00%	4.65%
	LTS	419	118	34.24%	0.10%	-23.34%	-4.01%	11.25%	-6.64%	66.87%	19.47%	24.92%

表 3 外れ値除去法を適用した場合の評価指標 (Kitchenham データ)  
Table 3 Evaluation criteria after outlier deletion (Kitchenham data).

外れ値除去法	AE 平均値	AE 中央値	MRE 平均値	MRE 中央値	MER 平均値	MER 中央値	ABRE 平均値	ABRE 中央値	BRE 平均値	BRE 中央値	除去率	
未適用	1335	635	95.80%	40.82%	50.81%	36.69%	110.95%	49.57%	-54.36%	-11.10%	0.00%	
抽出後 適用	提案方法	-20	7	17.45%	2.91%	-10.08%	-2.07%	7.61%	1.67%	37.77%	8.14%	17.21%
	Mantel 相関	14	2	1.02%	0.28%	0.45%	0.51%	1.27%	0.51%	0.63%	0.17%	0.03%
	k-means 法	-14	19	0.37%	-0.27%	-0.15%	-0.72%	0.20%	0.31%	0.81%	-0.72%	1.58%
	Cook の距離	-73	1	0.58%	0.61%	-3.77%	-1.54%	-2.77%	-2.24%	7.42%	5.23%	9.79%
	LTS	-223	15	10.52%	1.10%	-12.83%	-1.39%	-1.26%	-0.59%	33.21%	9.63%	19.64%
抽出前 適用	提案方法	-81	-7	-13.80%	-0.41%	3.46%	-2.06%	-9.63%	-3.25%	-23.39%	-8.05%	14.41%
	Mantel 相関	-23	-2	-0.04%	1.67%	1.63%	0.91%	1.30%	2.00%	-2.14%	0.14%	1.76%
	k-means 法	19	17	1.96%	-0.05%	-1.12%	0.71%	0.98%	0.80%	3.97%	1.18%	1.91%
	Cook の距離	46	40	4.86%	2.90%	2.99%	2.89%	6.52%	5.09%	1.82%	2.19%	5.44%
	LTS	28	-6	-1.02%	1.76%	0.72%	0.73%	-0.42%	1.77%	-1.91%	0.99%	25.00%

表 4 外れ値除去法を適用した場合の評価指標 (Desharnais データ)  
Table 4 Evaluation criteria after outlier deletion (Desharnais data).

外れ値除去法	AE 平均値	AE 中央値	MRE 平均値	MRE 中央値	MER 平均値	MER 中央値	ABRE 平均値	ABRE 中央値	BRE 平均値	BRE 中央値	除去率	
未適用	2031	1195	45.81%	29.87%	38.41%	30.78%	55.10%	35.36%	-12.82%	-6.46%	0.00%	
抽出後 適用	提案方法	-23	-48	0.24%	-1.89%	-2.40%	-0.38%	-1.54%	-1.88%	5.24%	2.52%	12.5%
	Mantel 相関	0	0	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	k-means 法	-2	-10	-0.04%	-0.02%	0.01%	-0.09%	-0.04%	-0.14%	-0.08%	0.00%	0.49%
	Cook の距離	-24	41	-0.14%	0.31%	-0.30%	0.09%	-0.39%	-0.20%	0.87%	2.01%	6.04%
	LTS	7	63	1.52%	0.29%	-0.99%	0.44%	0.38%	0.79%	4.95%	4.90%	9.19%
抽出前 適用	提案方法	-255	-403	-29.68%	-6.39%	-4.34%	-2.24%	-29.64%	-7.01%	-33.13%	-9.10%	12.05%
	Mantel 相関	35	8	-1.99%	-1.07%	1.16%	1.34%	-1.11%	-0.22%	-5.74%	-6.54%	1.03%
	k-means 法	0	0	0.03%	-0.66%	-0.04%	0.08%	-0.02%	-0.52%	0.13%	0.00%	0.51%
	Cook の距離	-8	80	-4.04%	-2.38%	-9.65%	-5.02%	-11.03%	-8.37%	10.95%	-2.41%	8.97%
	LTS	-10	12	-3.47%	-3.23%	-1.91%	-0.79%	-4.72%	-4.62%	-2.11%	-1.02%	25.64%

あり、更に外れ値除去法を適用しない場合と有意差があった評価指標はなかったことから、外れ値除去の効果はごくわずかであるといえる。

**Cook の距離を用いた除去法** 全ての評価指標が改善していた。ただし、外れ値除去法を適用しない場合との有意差は見られず、また、ABRE の平均値と中央値の改善の程度はそれぞれ 9.52%、2.22%であり、提案方法や LTS を用いた除去法よりも外れ値除去の効

果が大きいかといえない。

**LTS を用いた除去法** MER の平均値と中央値は悪化していたが、ABRE の平均値で 24.18%、中央値で 4.30%改善し、その他の評価指標も改善しており (MER 中央値と ABRE 中央値以外で有意差があった)、提案方法と同程度の効果が見られた。



## 6.2 ISBSG データに外れ値除去法を適用した場合 (類似プロジェクト抽出前に適用)

**提案手法** 10 個中 3 個の評価指標しか改善しておらず、更に *ABRE* 平均値が 9.39%悪化していた。したがって、提案方法を類似プロジェクト抽出前に適用すべきでないといえる。

**Mantel 相関を用いた除去法** 6 個の評価指標が改善し、改善しなかった評価指標についても悪化は 1.34%以下であった。ただし、改善の程度は *ABRE* 平均値で 3.19%であり、*ABRE* 中央値ではほとんど変化がなく、更に外れ値除去法を適用しない場合と有意差があった評価指標は *MRE* 平均値だけであったことから、外れ値除去の効果はごくわずかであるといえる。

**k-means 法を用いた除去法, Cook の距離を用いた除去法** 4 個以下の評価指標しか改善しておらず、更に *ABRE* の平均値、中央値とも 6%以上悪化していた。したがって、*k-means* 法を用いた除去法、Cook の距離を用いた除去法は ISBSG データには適していないといえる。

**LTS を用いた除去法** 7 個の評価指標が改善していたが (*AE* 平均値、*MRE* 平均値、*BRE* の平均値と中央値に有意差があった)、*ABRE* 中央値が悪化していた。どの評価指標を重視するかによって、LTS を用いた除去法に効果があるかどうかの判断は異なるが、*BRE* 中央値を除いて、類似プロジェクト抽出後に提案手法を適用した場合の方が評価指標の改善の程度が大きく、少なくとも類似プロジェクト抽出後に提案手法を適用した場合よりも効果が大きいとはいえない。

## 6.3 Kitchenham データに外れ値除去法を適用した場合 (類似プロジェクト抽出後に適用)

**提案手法** 10 個中 7 個の評価指標が改善しており、*ABRE* の平均値で 7.61%、中央値で 1.67%改善していた (*MRE* 平均値、*BRE* の平均値と中央値が有意に改善し、*MER* 平均値が有意に悪化していた)。ただし 6.4 で後述するように、類似プロジェクト抽出前適用の Cook の距離を用いた除去法の方が効果が高い可能性がある。

**Mantel 相関を用いた除去法** 全ての評価指標が改善していたが (*ABRE* の平均値と中央値はそれぞれ 1.27%、0.51%改善していた)、その程度は小さく、有意差もなかったことから、外れ値除去の効果はごくわずかであるといえる。

**k-means 法を用いた除去法** 5 個の評価指標が改

善していたが (*ABRE* の平均値と中央値はそれぞれ 0.20%、0.31%改善していた)、その程度は小さく、有意差もなかったことから、外れ値除去の効果はごくわずかであるといえる。

**Cook の距離を用いた除去法** 5 個の評価指標が改善しており、*BRE* の平均値と中央値が 5%以上改善していた。ただし、*ABRE* の平均値と中央値が若干悪化しており、かつ指標値が改善している評価指標において有意差がなかったことから、少なくとも類似プロジェクト抽出後に提案手法を適用した場合よりも外れ値除去の効果は大きいとはいえない。

**LTS を用いた除去法** 5 個の評価指標が改善していた (*MRE* 平均値、*BRE* の平均値と中央値が有意に改善しており、*MER* 平均値が有意に悪化していた)。ただし、*ABRE* の平均値と中央値が若干悪化しており、かつ *AE* 平均値の悪化の程度が比較的大きいことから、少なくとも類似プロジェクト抽出後に提案手法を適用した場合よりも外れ値除去の効果は大きいとはいえない。

## 6.4 Kitchenham データに外れ値除去法を適用した場合 (類似プロジェクト抽出前に適用)

**提案手法** 10 個中 1 個の評価指標しか改善しておらず、更に *ABRE* 平均値が 9.63%悪化していた。したがって、提案方法を類似プロジェクト抽出前に適用すべきでないといえる。

**Mantel 相関を用いた除去法** 6 個の評価指標が改善していたが (*ABRE* の平均値と中央値はそれぞれ 1.30%、2.00%改善していた)、その程度は小さくなく、有意差もなかったことから、外れ値除去の効果はごくわずかであるといえる。

**k-means 法を用いた除去法** 8 個の評価指標が改善していた (*BRE* 平均値が有意に改善していた)。ただし、*ABRE* の平均値と中央値の改善の程度が小さく (それぞれ 0.98%、0.80%)、類似プロジェクト抽出後に提案手法を適用した場合よりも外れ値除去の効果が大きいとはいえない。

**Cook の距離を用いた除去法** 全ての評価指標が改善しており (*MER* 平均値が有意に改善していた)、*ABRE* の平均値と中央値が 5%以上改善していたことから、外れ値除去の効果があったといえる。類似プロジェクト抽出後適用の提案手法と比較すると、*ABRE* の平均値の改善の差は小さく、かつ中央値の改善の程度は Cook の距離を用いた除去法の方が大きかった。Cook の距離を用いた除去法では、*BRE* の平均値と

中央値の改善の程度が提案方法に比べて小さいことに注意する必要があるが、Kitchenham データに対する外れ値除去の効果は、Cook の距離を用いた除去法の方が提案方法よりも高い可能性がある。

**LTS を用いた除去法** 6 個の評価指標が改善していたが (*ABRE* の平均値は 0.42%悪化し、中央値は 1.77%改善していた)、その程度は大きくなく、有意差もなかったことから、外れ値除去の効果はあるとしてもごくわずかであるといえる。

#### 6.5 Desharnais データに外れ値除去法を適用した場合 (類似プロジェクト抽出後に適用)

**提案手法** 10 個中 3 個の評価指標しか改善しておらず、改善している評価指標において有意差が見られず (*MER* 平均値が有意に悪化していた)、*ABRE* の平均値と中央値が若干悪化していたことから、外れ値除去の効果があるとはいえない。ただし、*ABRE* の平均値と中央値の悪化は最大でも 1.88%であり、見積り精度を大きく低下させているともいえない。

**Mantel 相関を用いた除去法** プロジェクトが全く除外されておらず、外れ値除去が機能していなかった。

**k-means 法を用いた除去法** 各評価指標にほとんど変化が見られず、Desharnais データに対して効果がなかった。

**Cook の距離を用いた除去法** 5 個の評価指標が改善していたが、有意差が見られず、*ABRE* の平均値と中央値が若干悪化していたことから、外れ値除去の効果があるとはいえない。ただし、*ABRE* の平均値と中央値の悪化は最大でも 0.39%であり、見積り精度を大きく低下させているともいえない。

**LTS を用いた除去法** 9 個の評価指標が改善していた。ただし、*ABRE* の平均値と中央値の改善の程度が小さく (それぞれ 0.38%, 0.79%)、かつ指標値が改善している評価指標において有意差がなかったことから、外れ値除去の効果はごくわずかであるといえる。

#### 6.6 Desharnais データに外れ値除去法を適用した場合 (類似プロジェクト抽出前に適用)

**提案手法** 10 個の評価指標全てが悪化しており、更に *ABRE* 平均値が 29.64%悪化していた。したがって、提案方法を類似プロジェクト抽出前に適用すべきでないといえる。

**Mantel 相関を用いた除去法** 4 個の評価指標が改善していたが、有意差が見られず、*ABRE* の平均値と中央値が最大で 1.11%悪化し、*ABRE* の平均値と中央値も 5%以上悪化していたことから、外れ値除去

の効果があるとはいえない。

**k-means 法を用いた除去法** 各評価指標にほとんど変化が見られず、Desharnais データに対して効果がなかった。

**Cook の距離を用いた除去法** 2 個の評価指標しか改善しておらず、更に *ABRE* の平均値と中央値が 8%以上悪化していた。したがって、Desharnais データに対し、類似プロジェクト抽出前に Cook の距離を用いた除去法を適用すべきでないといえる。

**LTS を用いた除去法** 1 個の評価指標しか改善しておらず、更に *ABRE* の平均値と中央値が 4.5%以上悪化していた。特に LTS 法では、*ABRE* 平均値と中央値で、適用しない場合と有意差が見られた (有意に悪化していた)。したがって、Desharnais データに対し、類似プロジェクト抽出前に LTS を用いた除去法を適用すべきでないといえる。

#### 6.7 ABRE 平均値に基づく比較

外れ値除去法を適用することにより *ABRE* 平均値が大きく改善していた場合、他の評価指標もおおむね改善しており、適用しない場合と有意差も見られ、外れ値除去の効果があったといえる。そこで *ABRE* 平均値に基づいて各手法の比較を行う。各データセットに外れ値除去法を適用した場合の *ABRE* 平均値の変化 (表 2~表 4 の数値をまとめたもの) と、その平均値を表 5 に示す。

あらゆるデータセットにおいて、ある一つの手法が最も精度が高くなるのが最も望ましい結果であるが、データマイニング手法の能力はデータセットの性質に強く依存し [14]、ソフトウェア開発データは多様であるため、平均的には性能の高い手法であっても、データセットによっては性能が低くなるのが広く見られる [13]~[15], [25]。従来の多くの論文では、見積り精度の悪化が少ないことや、比較的多くのデータセットで高い性能を示すことなど、相対的に性能が安定して高いことを根拠として、手法の有効性を判断している。本論文でもこれらの論文の考えを踏襲し、平均的に高い見積り精度を示し、かつ精度悪化が小さい手法が、より有効性の高い手法であると判断する。表 5 より以下の結論を下した。

- 提案方法を類似プロジェクト抽出後に適用することにより、平均的に最も高い見積り精度が得られる。また、二つのデータセットで 7.5%以上精度が改善しており、精度が悪化している一つのデータセットにおいても低下は 1.54%であり、最も性能が安定している。

表 5 外れ値除去法を適用した場合の ABRE 平均値の変化  
Table 5 Difference of average ABRE after outlier deletion.

外れ値除去法	抽出後				抽出前			
	平均値	ISBSG データ	Kitchenham データ	Desharnais データ	平均値	ISBSG データ	Kitchenham データ	Desharnais データ
提案方法	11.63%	28.83%	7.61%	-1.54%	-16.23%	-9.40%	-9.63%	-29.64%
Mantel 相関	0.42%	0.00%	1.27%	0.00%	1.13%	3.19%	1.30%	-1.11%
k-means 法	0.93%	2.63%	0.20%	-0.04%	-5.63%	-17.86%	0.98%	-0.02%
Cook の距離	2.12%	9.52%	-2.77%	-0.39%	-3.50%	-5.99%	6.52%	-11.03%
LTS	7.77%	24.18%	-1.26%	0.38%	2.04%	11.25%	-0.42%	-4.72%

• LTS を用いた除去法を類似プロジェクト抽出後に適用することにより、2 番目に高い見積り精度が得られる。二つのデータセットで見積り精度が改善しており、精度が悪化している一つのデータセットにおける精度の低下も 1.26% であることから、性能が安定しているといえる。ただし、7.5% 以上精度が改善しているデータセットは一つだけであり、提案方法に比べると見積り精度改善の効果が低い。

• Mantel 相関を用いた除去法を類似プロジェクト抽出後に適用した場合、二つのデータセットで見積り精度が改善しており、残り一つのデータセットにおける精度の低下は 1.11% であり、性能が安定しているといえる。ただし、最も精度が改善している場合でも 3.19% の精度改善しか見られず、提案方法（類似プロジェクト抽出後）と LTS を用いた除去法（同）に比べ、見積り精度改善の効果が低い。

• k-means を用いた除去法を類似プロジェクト抽出後に適用した場合、二つのデータセットで見積り精度が改善しており、残り一つのデータセットにおける精度の低下は 0.04% であり、性能が安定しているといえる。ただし、最も精度が改善している場合でも 2.63% の精度改善しか見られず、提案方法（類似プロジェクト抽出後）と LTS を用いた除去法（同）に比べ、見積り精度改善の効果が低い。

• 上記以外では、類似プロジェクト抽出前、後のどちらに適用した場合においても、二つのデータセットで精度が悪化しており、性能が安定しているとはいえない。また、提案方法と比べて平均的な見積り精度改善の効果が低い。

一見、外れ値除去を行ったり、Analogy 法で対象プロジェクトを絞り込んだりすることは見積り精度向上に役立つように思われるが、実際には Analogy 法に外れ値除去法を適用しても、精度が向上しない場合があった。これまで工数見積りに対する外れ値除去法の効果を確かめた研究は少なく、この点を実験により示

したことも、本論文の貢献の一つと考える。また、開発現場への適用が容易な外れ値除去法（提案方法）により、（より複雑な）従来法と同等以上の見積り精度向上が見込まれることを明らかにしたことも、本論文の貢献の一つであると考えられる。

## 7. 考 察

### 7.1 外れ値除去法の適用時期

提案方法を類似プロジェクト抽出前に適用すること、及び Mantel 相関を用いた除去法、k-means 法を用いた除去法を抽出後に適用することは、以下に述べる問題があり、実験においても高い精度向上が見られなかったことから、避けるべきであるといえる。

• 提案方法は生産性が大きく異なるプロジェクトを外れ値とみなすため、抽出前に適用すると、単純に（プロジェクトの特性を考慮せずに）生産性（規模補正法を用いていない場合は工数）の大きいプロジェクトと小さいプロジェクトが削除される。すなわち、説明変数（プロジェクト特性）を全く考慮せずにプロジェクトを削除することになり、原理的に外れ値除去法としてうまく機能しない。

• Mantel 相関を用いた除去法では、説明変数をセットにしてユークリッド距離を求めているが、類似プロジェクト抽出後は説明変数の値が非常に類似しており（直感的には、相関係数の一方の変数がほぼ定数になるイメージである）、マンテル相関（のテコ比）の差が現れにくく、結果の信頼性（外れ値特定精度）が低くなる。

• k-means 法を用いた除去法の場合、対象データ件数（類似プロジェクト数）が少ない上、既に Analogy 法でクラスタリングされている（とみなすことができる）類似プロジェクトに対し、更にクラスタリングを行うことになるため、信頼性が低くなる。

Mantel 相関を用いた除去法と k-means 法を用いた除去法を類似プロジェクト抽出後に適用した場合、抽

出前に適用した場合と比べて除去率が低い傾向が見られたが、これはしきい値が影響しているのではなく、上記のように適用自体に問題があるためであると考えられる。なお、LTS を用いた除去法の除去率が低い理由は、類似プロジェクト数が少ないためである。LTS を用いた除去法では一律に 25% のデータを削除するが、類似プロジェクトが少ない場合、例えば類似プロジェクト数が 5 件の場合、 $(5 \times 0.75 = 3.75)$  件より 4 件のプロジェクトが見積りに使われ、1 件 (20%) が除去される。

## 7.2 外れ値除去法間の効果の差

提案方法を類似プロジェクト抽出後に適用した場合と、(いくつかの) 従来法を抽出前に適用した場合で、外れ値除去の効果に差が見られたことに関して、以下の理由が考えられる (提案方法を類似プロジェクト抽出前に適用すること、及び Mantel 相関を用いた除去法、 $k$ -means 法を用いた除去法を抽出後に適用することに問題があることは前節で述べたとおりである)。

- 提案方法は Analogy 法の特徴を考慮し、Analogy 法的前提に当てはまらないプロジェクトを外れ値とみなしている。類似プロジェクトの生産性 (規模 / 工数) がばらついている場合、規模補正法による見積り精度が低下することが指摘されており [8], [19], 提案方法はこの問題点に対応している。そのため、性能が高くなっていると考えられる。

- $k$ -means 法を用いた除去法で除外されるプロジェクトは、データセットの他のプロジェクトと似ていないため、そもそも Analogy 法では見積り時に利用されない (類似プロジェクトとならない) 可能性がある。また、 $k$ -means 法ではフィットデータにおいて似たプロジェクトが少ない場合、外れ値とみなされるが、一般に開発規模の大きなプロジェクトは数が少ないため、それらが外れ値として除外される可能性がある。その場合、テストデータに含まれる開発規模の大きなプロジェクトの見積り精度が低下すると考えられる。これらが性能に影響を与えていると考えられる。

- Cook の距離を用いた除去法、LTS を用いた除去法は重回帰モデルをベースとしているが、類似プロジェクト抽出前に適用する場合、重回帰モデルで外れ値となるプロジェクトが Analogy 法で外れ値となるとは限らず、このためにあまり性能が安定しなかったと考えられる。

## 7.3 除去率と外れ値除去法の関連

データセットにより多少異なるが、適用のタイミン

グにかかわらず、提案方法、Cook の距離を用いた除去法、LTS を用いた除去法では、比較的除去率が高く、Mantel 相関を用いた除去法、 $k$ -means 法を用いた除去法の場合、低い除去率であった。このため、除去率と各手法には関連があると考えられる。また、除去率と外れ値除去法の効果の関連 (例えば、除去率の最も高い手法が、最も効果が高いなど) は、現時点では明確に見られなかった。

## 7.4 データセットによる外れ値除去法の効果の差

ISBSG データの場合、類似プロジェクト抽出後に外れ値除去法を適用した場合 (提案方法、Cook の距離を用いた除去法、LTS を用いた除去法) の効果が比較的高かった。ISBSG データのように、複数の企業から収集された多様なプロジェクトが含まれるデータセットの場合、データセット全体から外れ値を除去するよりも、類似プロジェクトを抽出してから外れ値を除去する方が適していると考えられる。

Desharnais データの場合、ほとんどの外れ値除去法において、*ABRE* 平均値の変化が 5% 未満となっていることから、外れ値除去法が有効に機能しなかったのではなく、そもそも見積り精度に影響を与えるような外れ値は含まれていないのではないかと推測される。実績工数の分布に着目した場合、Desharnais データは極端に大きな値が含まれていないと指摘されていることも [9], 外れ値がほとんど含まれていないと推測した根拠の一つである。

## 8. む す び

本論文では、類似性に基づく工数見積り方法 (Analogy 法) に複数の外れ値除去法を適用し、それぞれの除去法を実験的に評価した。また、Analogy 法に適した外れ値除去法を新たに提案した。実験では、類似プロジェクト抽出前に適用した場合と、類似プロジェクト抽出後に適用した場合の効果を確認した。ソフトウェア開発企業で収集された 3 種類のデータセットを用いて外れ値除去法の効果を比較した結果、提案方法を類似プロジェクト抽出後に適用することにより、平均的に最も高い見積り精度が得られることが分かった。今後の課題としては、より多くのデータセットを用いて同様の実験を繰り返すことにより、提案方法が有用となるコンテキストをより明確にしていけることが重要となる。

謝辞 本論文で利用した工数見積りツールの作者である大杉直樹氏に、深く感謝致します。本研究の一

部は、文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた。また、本研究の一部は、文部科学省科学研究補助費（若手 B：課題番号 22700034）による助成を受けた。

### 文 献

- [1] G. Boetticher, T. Menzies, and T. Ostrand, PROMISE Repository of empirical software engineering data, <http://promisedata.org/> repository, West Virginia University, Department of Computer Science, 2007.
- [2] C. Burgess and M. Lefley, “Can genetic programming improve software effort estimation? A comparative evaluation,” *J. Information and Software Technology*, vol.43, no.14, pp.863–873, 2001.
- [3] S. Conte, H. Dunsmore, and V. Shen, *Software Engineering, Metrics and Models*, Benjamin/Cummings, 1986.
- [4] J. Desharnais, “Analyse statistique de la productivité des projets informatique a partie de la technique des point des fonction,” Master Thesis, University of Montreal, 1989.
- [5] J. Han and M. Kamber, “Data mining: Concepts and techniques,” Morgan Kaufmann, 2000.
- [6] International Software Benchmarking Standards Group (ISBSG), ISBSG Estimating: Benchmarking and research suite, ISBSG, 2004.
- [7] J. Keung, B. Kitchenham, and R. Jeffery, “Analogy-X: Providing statistical inference to analogy-based software cost estimation,” *IEEE Trans. Softw. Eng.*, vol.34, no.4, pp.471–484, 2008.
- [8] C. Kirsopp, E. Mendes, R. Premraj, and M. Shepperd, “An empirical analysis of linear adaptation techniques for case-based prediction,” *Proc. International Conference Case-Based Reasoning*, pp.231–245, Trondheim, Norway, June 2003.
- [9] C. Kirsopp and M. Shepperd, “Making inferences with small numbers of training sets,” *IEE Proc. Software*, vol.149, no.5, pp.123–130, 2002.
- [10] B. Kitchenham, S. MacDonell, L. Pickard, and M. Shepperd, “What accuracy statistics really measure,” *IEE Proc. Software*, vol.148, no.3, pp.81–85, 2001.
- [11] B. Kitchenham, S.L. Pfleeger, B. McColl, and S. Eagan, “An empirical study of maintenance and development estimation accuracy,” *J. Systems and Software*, vol.64, no.1, pp.57–77, 2002.
- [12] J. Klijnen, “Sensitivity analysis and related analyses: A survey of statistical techniques,” *J. Statistical Computation and Simulation*, vol.57, no.1-4, pp.111–142, 1987.
- [13] Y. Kultur, B. Turhan, and A. Bener, “Ensemble of neural networks with associative memory (ENNA) for estimating software development costs,” *Knowledge-Based Systems*, vol.22, no.6, pp.395–402, 2009.
- [14] J. Li and G. Ruhe, “Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+,” *Empir. Softw. Eng.*, vol.13, no.1, pp.63–96, 2008.
- [15] J. Li, G. Ruhe, A. Al-Emran, and M. Richter, “A flexible method for software effort estimation by analogy,” *Empir. Softw. Eng.*, vol.12, no.1, pp.65–106, 2007.
- [16] C. Lokan, “What should you optimize when building an estimation model?,” *Proc. International Software Metrics Symposium (METRICS)*, Como, Italy, Sept. 2005.
- [17] C. Lokan and E. Mendes, “Cross-company and single-company effort models using the ISBSG Database: A further replicated study,” *Proc. International Symposium on Empirical Software Engineering (ISESE)*, pp.75–84, Rio de Janeiro, Brazil, Sept. 2006.
- [18] E. Mendes, S. Martino, F. Ferrucci, and C. Gravino, “Cross-company vs. single-company web effort models using the Tukutuku database: An extended study,” *J. Syst. Softw.*, vol.81, no.5, pp.673–690, 2008.
- [19] E. Mendes, N. Mosley, and S. Counsell, “A replicated assessment of the use of adaptation rules to improve web cost estimation,” *Proc. International Symposium on Empirical Software Engineering (ISESE)*, pp.100–109, Rome, Italy, Sept. 2003.
- [20] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, “Robust regression for developing software estimation models,” *J. Syst. Softw.*, vol.27, no.1, pp.3–16, 1994.
- [21] K. Møllokken-Østvold and M. Jørgensen, “A comparison of software project overruns-flexible versus sequential development models,” *IEEE Trans. Softw. Eng.*, vol.31, no.9, pp.754–766, Sept. 2005.
- [22] National Association of Testing Authorities (NATA), *Guide to NATA proficiency testing, Version 2*, National Association of Testing Authorities, 2002.
- [23] V. Nguyen, B. Steece, and B. Boehm, “A constrained regression technique for COCOMO calibration,” *Proc. International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp.213–222, Kaiserslautern, Germany, Oct. 2008.
- [24] N. Ohsugi, A. Monden, N. Kikuchi, M. Barker, M. Tsunoda, T. Kakimoto, and K. Matsumoto, “Is this cost estimate reliable? — The relationship between homogeneity of analogues and estimation reliability,” *Proc. International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp.384–392, Madrid, Spain, Sept. 2007.
- [25] A. Oliveira, P. Braga, R. Lima, and M. Cornélio, “GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation,” *Information and Software Technology*, vol.52, no.11, pp.1155–1166, 2010.

- [26] Y. Seo, K. Yoon, and D. Bae, "An empirical analysis of software effort estimation with outlier elimination," Proc. International Workshop on Predictor Models in Software Engineering (PROMISE), pp.25–32, Leipzig, Germany, May 2008.
- [27] M. Shepperd and C. Schofield, "Estimating software project effort using analogies," IEEE Trans. Softw. Eng., vol.23, no.12, pp.736–743, 1997.
- [28] H. Tan, Y. Zhao, and H. Zhang, "Conceptual data model-based software size estimation for information systems," ACM Trans. Softw. Eng. Methodol., vol.19, no.2, pp.1–37, 2009.
- [29] 田中 豊, 垂水共之 (編), Windows 版 統計解析ハンドブック 多変量解析, 共立出版, 1995.
- [30] A. Tosun, B. Turhan, and A. Bener, "Feature weighting heuristics for analogy-based effort estimation models," Expert Systems with Applications, vol.36, no.7, pp.10325–10333, 2009.
- [31] F. Walkerdien and R. Jeffery, "An empirical study of analogy-based software effort estimation," Empir. Softw. Eng., vol.4, no.2, pp.135–158, 1999.

(平成 23 年 10 月 3 日受付, 11 月 24 日再受付)



柿元 健 (正員)

2003 神戸市立工業高等専門学校専攻科電気電子工学専攻了。2005 奈良先端科学技術大学院大学情報科学研究科博士前期課程了。2008 同大博士後期課程了。同年大阪大学大学院情報科学研究科特任助教。2010 香川高等専門学校電気情報工学科講師。博士(工学)。ソフトウェア信頼性/開発工数予測などの研究に従事。情報処理学会, IEEE 各会員。



松本 健一 (正員)

1985 阪大・基礎工・情報工学卒。1989 同大大学院博士課程中退。同年同大学基礎工学部情報工学科助手。1993 奈良先端科学技術大学院大学情報科学研究科助教授。2001 同大学教授。工博。エンピリカルソフトウェア工学, 特に, プロジェクトデータ収集/利用支援の研究に従事。日本ソフトウェア科学会, ACM 各会員, IEEE Senior Member。



角田 雅照 (正員)

1997 和歌山大・経済卒。2004 奈良先端科学技術大学院大学情報科学研究科博士前期課程了。2007 同大同研究科博士後期課程了。同年同大学同研究科特任助教。博士(工学)。ソフトウェアメトリックスの研究に従事。情報処理学会, ヒューマンインタ

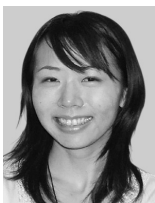
フェース学会, 教育システム情報学会, IEEE 各会員。



門田 暁人 (正員)

1994 名大・工・電気卒。1998 奈良先端科学技術大学院大学情報科学研究科博士後期課程了。同年同大同研究科助手。2004 同大学助教授。2007 同大学准教授。2003~2004 Auckland 大学客員研究員。博士(工学)。ソフトウェアメトリックス, ソフト

ウェアプロテクション, ヒューマンファクタ等の研究に従事。情報処理学会, 日本ソフトウェア科学会, IEEE, ACM 各会員。



渡邊 瑞穂

2006 京都教育大・教育卒。2008 奈良先端科学技術大学院大学情報科学研究科博士前期課程了。同年日本アイ・ビー・エム(株)入社。在学時, ソフトウェア開発工数予測などの研究に従事。