

モデルに基づいた工数見積もりにおける 重み付き Moving Window 法の評価

天寄 聡介^{1,a)}

概要：工数見積もりモデルの構築において、直近の開発終了プロジェクトのみを用いる window 法が有効であると考えられている。window 法の根拠となるのは、直近に開発終了したプロジェクトのデータの方が、より古いプロジェクトデータよりも組織の特徴を表しているという仮定である。window 法では直近の開発終了プロジェクトを全て同じ重み付けで扱っている。しかしながら、直近の開発終了プロジェクト間にも時間的順序があり、この順序関係をプロジェクトデータの重み付けに用いることで工数見積もり精度が向上すると考えられる。本研究では、window 法で選択されたプロジェクトデータへの時間順序に基づいた重み付けが工数見積もり精度の変化に及ぼす影響を調査した。この調査では CSC と Maxwell の 2 つのデータセットを用意し、過去の研究と同様の方法により実験を行った。調査の結果、CSC データセットにおいては重み付けを行うことで工数見積もり精度が統計的に有意に改善することが観察できた。この結果は window 法の有効性の理解に貢献すると考えられる。

キーワード：Window 法, 工数見積もり, 線形回帰モデル

Evaluation on Weighted Moving Windows on Model-based Effort Estimation

Abstract: In construction of an effort estimation model, it seems effective to use a window of training dataset so that the model is trained with a subset of only recently finished projects. In windowing approach, all projects selected are considered that they have the same importance. However, the selected projects can be ordered in chronological manner, and it is expected that effort estimation accuracy could be improved with the use of weights according to chronological similarity to a target project. In this study, we examined the effects of the weighting method on effort estimation accuracy with the windowing approach. We applied the same experiment settings as the past study. As a result, we confirmed that weighting methods significantly improved estimation accuracy on CSC dataset. This result contributes to understand property of the windowing approach.

Keywords: Windowing, Effort Estimation, Linear Regression

1. はじめに

ソフトウェア工数見積もりはソフトウェア開発において最も重要な作業の一つである。ソフトウェア工数見積もりは次の二つに大別できる [14]。すなわち、専門家による見積もりとモデルに基づいた見積もりである。モデルに基づいた見積もり手法の例としては COCOMO や SLIM などが挙げられる。現在では、モデルに基づいた工数見積もり

に関する研究が特に盛んに行われている [6]。

ソフトウェア工数見積もりモデルの評価は、実際のソフトウェア開発プロジェクトで収集された定量的なデータセットを用いて行われる。評価実験では、このデータセットをランダムにモデルの訓練用と検証用に分割して行う random hold-out 法や、leave-one-out 法などの交差検証法がよく用いられる。訓練用データを用いてソフトウェア工数見積もりモデルを構築し、その性能は検証用データを用いて評価する。

ソフトウェア開発プロジェクトのデータはプロジェクト

¹ 岡山県立大学 Okayama Prefectural University

^{a)} amasaki@cse.oka-pu.ac.jp

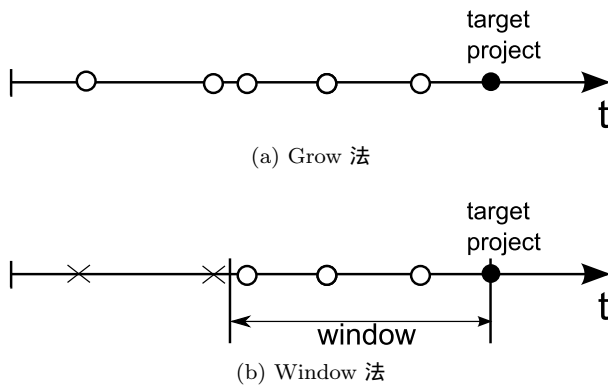


図 1: grow 法と window 法
Fig. 1 Growing and Windowing approach

が終了するたびに収集される．そのため，ある時点において収集済みのプロジェクトのデータを用いて，その時点より先に開始されるプロジェクトの工数を見積もる性能が重要である．この観点に立つと，プロジェクトデータ収集の時系列に沿ってソフトウェア工数見積もりモデルを評価する方がより適切であると考えられる [15]．さらに，一定以上過去に終了したプロジェクトのデータは見積もりの根拠としての有用性が相対的に低いと考えられるため，そのようなデータは使用しない方がより正確な見積もりを行うことが可能とも推測できる．

以上の観点から，過去のプロジェクトデータ全てを利用する場合と，直近のプロジェクトデータのみを利用する場合とで工数見積もりの精度を比較する研究が行われている [2], [11]．本論文では直近のプロジェクトデータのみを利用する方法を window 法と呼ぶ．文献 [11] では，線形回帰モデルに基づいた工数見積もりモデルを使用した場合の評価を行っている．また，文献 [2] では，類推法に基づいた工数見積もりモデルを使用した場合の評価を行っている．さらに，文献 [1] では，同じプロジェクトデータを対象として工数見積もりモデルの間で比較評価を行っている．以上の研究はいずれも直近のプロジェクトデータは全て同程度に重要であると考えている．一方，時系列上で近くにあるプロジェクトデータの方が利用価値が高いという考え方に従えば，時系列に沿ってプロジェクトデータに重み付けを行う方法も考えられる．

本研究では，時系列に沿って重み付けを行った場合における window 法の工数見積もり精度に対する影響を評価した．

2. 準備

2.1 Window 法

図 1 に時系列に沿った工数見積もりモデルの評価方法を示す．水平矢印は時間軸を表す．黒丸は工数見積もりの対象となるプロジェクトを示す．白丸は終了したプロジェクトを示す．

利用するプロジェクトデータに制限を設けない方法（以降では grow 法と呼ぶ）を図 1a に示す．grow 法では，工数見積もり実施時に収集済みの全てのプロジェクトデータを用いて工数見積もりモデルを構築する．プロジェクトデータの収集を始めてしばらくはデータ数が少なく，プロジェクトデータの特性間の関係についても単純な分析しかできない．そのため，工数見積もりモデルも単純な構造になる．しかし，プロジェクトデータの数が増加していくため，特性間の関係をより詳細に分析し，精度が高い工数見積もりモデルを構築できるようになる．

利用するプロジェクトデータに制限を設ける window 法を図 1b に示す．window 法では収集済みのプロジェクトデータを次の 2 つに分類する．すなわち，直近のプロジェクトと古いプロジェクトである．そして，直近のプロジェクトデータを用いて工数見積もりモデルを構築する．図 1b の window は，直近のプロジェクトと見なす範囲を定めている．この例では grow 法で使用されたプロジェクトデータのうち 3 つのみを直近のプロジェクトに分類している．

2.2 重み付き線形回帰モデル

線形回帰モデルは工数見積もりによく用いられる手法の一つである．最も単純な線形回帰モデルでは工数とソフトウェア規模を以下の関係で表す．

$$\text{Effort} = b_0 + b_1 \text{Size} + \epsilon \quad (1)$$

ここで ϵ は誤差を表す．この誤差を最小化しようパラメータ b_0 及び b_1 を訓練用データから求める．具体的には以下の式の値を最小化しようパラメータを求める．

$$\sum_{i=1}^n (\text{Effort}_i - b_0 - b_1 \text{Size}_i)^2 \quad (2)$$

n は訓練用データの数を表す．

式 (2) では訓練用データに対する誤差の評価は全て等しく扱われている．重み付き線形回帰モデルでは個々の訓練用データの相対的な重要性を重み付けによって制御する．具体的には以下の式の値を最小化しようパラメータを求める．

$$\sum_{i=1}^n w_i (\text{Effort}_i - b_0 - b_1 \text{Size}_i)^2 \quad (3)$$

ここで w_i は訓練用データに対する重みである．

本研究では，工数見積もりの対象となるプロジェクトと時系列的に近いプロジェクトに大きな重みをつける．重みの配分には複数の方法がある [10] が，本研究では表 1 の四種類の重み付けの方法を用いた．Rectangular は重み付けなしと同じである． x の値は訓練用データに対して以下の方法で求める．

$$x = \frac{n_i}{n} \quad (4)$$

表 1: 重み付け関数
Table 1 Weighted functions

Name	Formula
Rectangular (Uniform)	$W(x) = 1, x < 1$
Triangular	$W(x) = 1 - x , x < 1$
Epanechnikov	$W(x) = 1 - x^2, x < 1$
Gaussian	$W(x) = \exp(-(2.5x)^2/2)$

ここで n_i は訓練用データを日付順に昇順で並べた時の順位である。つまり、新しいほど順位が低くなり、 x の値は小さくなる。

3. データセット

本研究では PROMISE Repository で一般に公開されているデータセットを用いた。PROMISE Repository で公開されているデータセットのうち、CSC データセットと Maxwell データセットを選択した。これらのデータセットには開発期間や開発開始もしくは完了日が記載されている。また、ISBSG が収集しているデータセットと異なり、それぞれ単一の企業で収集されたデータセットであり、時間経過に伴う生産性の効果などがより観察しやすいと考えられる。以下にそれぞれのデータセットの特徴について述べる。

3.1 CSC データセット

このデータセットは 1994 年から 1999 年にかけて Computer Science Corporation で収集されたものである。CSC データセットには 145 プロジェクトのデータが含まれている。プロジェクトは新規開発が保守開発のいずれかである。ソフトウェア規模は調整済みファンクションポイントで計測されており、工数は人時で記録されている。

CSC データセットにはプロジェクトの顧客の区別も記録されている。Kitchenham ら [7] は 6 社ある顧客のうち、Client 1 と Client 2 向けのプロジェクトデータを用いて工数見積もりモデルを構築している。しかし、Client 1 と Client 2 向けに構築された工数見積もりモデルは特徴が異なっており、また、特に Client 1 向けのプロジェクトの数は 16 プロジェクトと少ない。本研究では Client 2 向けのプロジェクトデータのみを対象とする。Client 2 向けのプロジェクトデータには 新規開発と保守開発の両方のプロジェクトが含まれる。Client 1 向けのプロジェクトを除くことでデータセットの均質性が高まったと言える。本研究では Kitchenham らと同様に 1 つプロジェクトを取り除いている。このプロジェクトは外れ値と見なせる特徴をいくつか備えている。例えば、このプロジェクトは CSC において初めて取り組まれたクライアントサーバ型の開発である。また、当時としては経験のない大規模なプロジェクトであった。これらの点から、このプロジェクトをのぞくこ

表 2: CSC データセットの統計量
Table 2 Summary Statistics for Ratio-scaled Variables (CSC)

Variable	Mean	Median	StDev	Min	Max
Size	394.8	258.2	388.3	30.0	2075.8
Effort	2461.0	1650.0	2575.0	219.0	15670.0

とは十分な理由があると言える。

以上の選別の結果、本研究では 105 プロジェクトのデータを用いた。新規開発が 38 プロジェクトで、67 プロジェクトが保守開発である。工数見積もりモデルを構築するために新規開発が保守開発かを示す変数を含めた。表 2 に 105 プロジェクトの工数および規模の統計量を示す。

3.2 Maxwell データセット

Maxwell データセットは文献 [12] に記載されている。このデータセットはフィンランドで最も大きな商業銀行の一つにおいて 1985 年から 1994 年にかけて収集されたものである。

Maxwell データセットには 63 プロジェクトが含まれる。各プロジェクトについてアプリケーションの種別やスタッフのスキルレベルなど 22 項目の情報が記録されている。ソフトウェア規模はファンクションポイント法で計測されており、工数は人時で記録されている。CSC データセットとは対照的に新規開発が保守開発かといった情報は記録されていない。本研究では変数選択によって記録されている項目から工数見積もりに有用なものをステップワイズ法により取捨選択する。しかし、Maxwell データセットの場合は記録されている情報の項目が多く、window 幅によってはステップワイズ法の適用が難しい [5]。そこで本研究では事前にデータセット全体を分析して以下の 4 つの情報を候補とした。

- ソフトウェア規模 (Size)
- ユーザインターフェイスの種類 (Ifc)
- 要求の不安定さ (T08)
- 要求の品質 (T09)

ここで括弧内は対応する情報が記録されている変数名である。Ifc は水準数 2 の名義尺度で、ユーザインターフェイスがテキストベースか GUI であるかを示す。しかしプロジェクトデータの多くが GUI であったため、テキストベースのプロジェクト 4 件を省いた。このことでデータセットがより均質になったと言える。T08 と T09 は順序尺度の変数である。データセット全体を分析してみると、各変数の水準の効果はほぼ線形に変化していた。このような場合、比尺度に置き換えた方が工数見積もりモデルの構築には適している。そこで本研究ではこれら 2 つの変数を比尺度に置き換えた。

以上の結果から、最終的に 58 プロジェクトのデータセットを実験に用いることにした。プロジェクトデータの情報

表 3: Maxwell データセットの統計量 (比尺度)

Table 3 Summary Statistics for Ratio-scaled Variables (Maxwell)

Variable	Mean	Median	StDev	Min	Max
Size	654.6	377.0	796.9	48.0	3643.0
Effort	8426.0	5190.0	10794.7	583.0	63694.0

表 4: Maxwell データセットの統計量 (順序尺度)

Table 4 Summary Statistics for Ordinal-scaled Variables (Maxwell)

Variable	Very Low	Low	Normal	High	Very High
T08	0	5	17	19	17
T09	0	2	9	33	14

として含まれるのは、ソフトウェア規模、T08、T09 である。表 3 に比尺度のメトリクスの情報を示す。また、表 4 に T08、T09 の各水準に属するプロジェクト数を示す。

3.3 評価基準

工数見積もりモデルの性能は見積もり値と実績値の誤差で評価する。本研究では以下の 2 つの誤差に基づいて性能を評価する。

$$\text{MRE} = |\text{Act} - \text{Est}|,$$

$$\text{MAE} = \frac{|\text{Act} - \text{Est}|}{\text{Act}}.$$

MRE は最もよく用いられているが過大見積もりが小さく評価される傾向がある。一方で、MAE は過大見積もりおよび過小見積もりに対して中立的に評価できる。

MRE と MAE は検証用データセットに対する見積もりに対して計算する。評価基準の計算はその結果を用いて計算する。既存研究では次の評価基準が用いられている。すなわち、MMRE と MMAE である [14]。MMRE および MMAE はそれぞれ MRE と MAE の算術平均である。

見積もり精度の違いを評価するため、Wilcoxon の順位和検定を有意水準 $\alpha = 0.1$ で用いる。具体的には統計ソフトウェア R の coin パッケージの wilcoxsign_test をデフォルトのオプションで使用した。

3.4 実験手順

本研究では時系列に沿って評価実験を行う。このとき、適切な window 幅は自明でないため、一定の範囲で window 幅を変化させて実験を行う。ある window 幅 w における実験の手順は次の通りである。

- (1) 全てのプロジェクトデータをプロジェクト開始日を基準としてソートする
- (2) 少なくとも window 幅の範囲のプロジェクトが完了した日付の後で、最も早く開始されているプロジェクト p_i を見つける

- (3) プロジェクト p_i の開始日より前に完了したプロジェクトを訓練用に、そして p_i を検証用のデータとして window 法と grow 法を用いた工数見積もりを行う

- (4) 次に開始日が早いプロジェクト p_{i+1} を見つける

- (5) 上記手順 (3) から (4) を検証用のプロジェクトがなくなるまで繰り返す

- (6) 工数見積もりの精度を比較評価する

window 幅の設定について 2 通りの方法が考えられる。一つは日付に基づく方法である。これは、例えば、ある時点から w 日前までのプロジェクトを工数見積もりに用いる考え方である。もう一つはプロジェクトの数に基づく方法である。これは、ある時点より前に過去に完了した w プロジェクトを工数見積もりに用いる考え方である。日付に基づく方法の場合、直近の一定期間における生産性を反映した工数見積もりが可能になる。しかし、一定期間内に完了するプロジェクトの数が全ての期間に渡って同程度にならない。この点は単一の組織から収集されたデータセットにおいて顕著である。その理由として、単一の組織から収集できるプロジェクトデータの数が少ない点が挙げられる。一方、プロジェクトの数に基づく方法の場合、常に一定のプロジェクト数で工数見積もりモデルを構築できるため、モデルの精度が安定すると考えられる。しかし、訓練用のデータの古さが見積もりを行うたびに異なる。

本研究では、既存研究と比較可能な結果が得られるよう、プロジェクトの数に基づく方法を採用する。

次に必要となるのは window 幅の設定方法である。既存研究では複数の window 幅で実験が行われている。この際、各 window 幅で線形回帰モデルに基づいた工数見積もりモデルを構築し、構築されたモデルが全て統計的に有意であることを確認している。そして、この条件を満たす最小の window 幅を window 幅の下限とみなしている。また、window 幅の上限については、工数見積もりの精度の評価に必要なデータ数が確保できるように設定されている。

文献 [2] で著者らは CSC データセットにおける最小の window 幅を $w = 10$ としている。また、統計的検定により工数見積もり精度の差を評価するため、検証用データの数を 10 としている。また、Maxwell データセットについては最小の window 幅を $w = 10$ 、最大の window 幅を $w = 35$ とした。本研究もこれらの基準に従った。

4. 結果と考察

4.1 CSC データセット

図 2 に重み付きの有無による工数見積もり精度の差を示す。横軸は window 幅を示し、縦軸は重み付き線形回帰モデルと通常の線形回帰モデル (Rectangular) の間の工数見積もり精度の差を示す。縦軸の値が 0 より小さいとき、重み付き線形回帰モデルの方が工数見積もり精度が高い。

図 2a について、window 幅が 15 から 20 付近の時およ

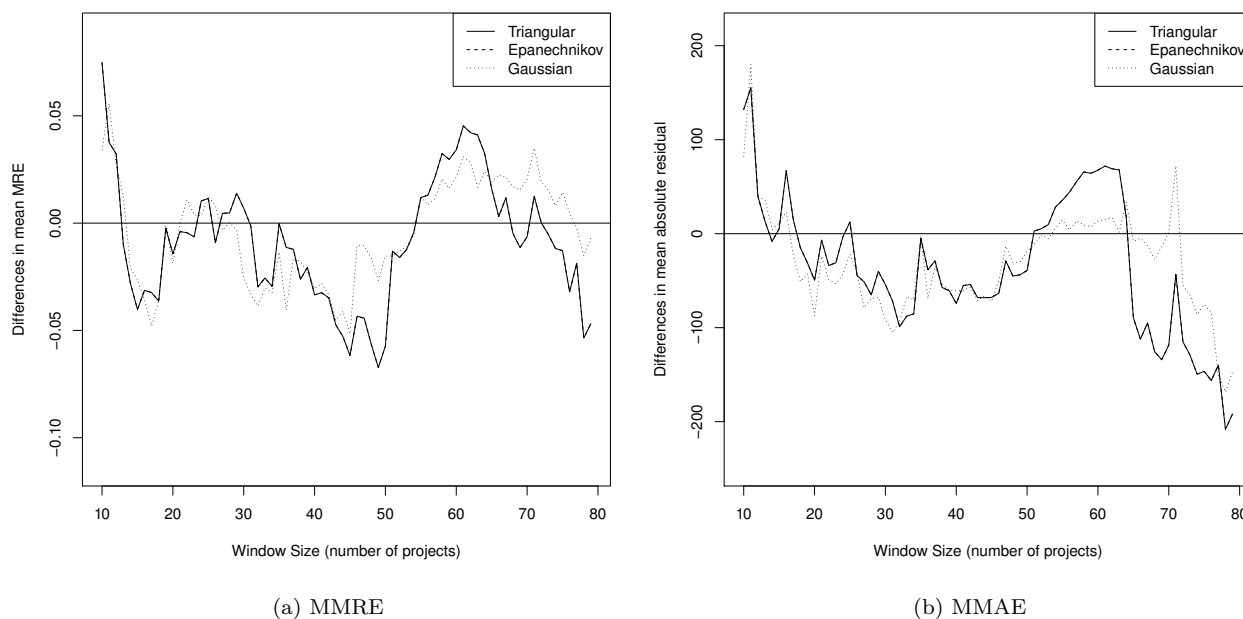


図 2: 線形回帰モデルとの見積もり精度の差 (CSC, window 法)

Fig. 2 Difference in estimation accuracy against non-weighted regression (CSC, windowing)

び 30 から 55 付近にかけて、全ての重み付き関数において MMRE の観点から工数見積もり精度の向上が確認できた。一方、window 幅が 20 から 30 の間および 55 から 70 付近の時、全ての重み付き関数において MMRE の観点から工数見積もり精度の低下が確認できた。

図 2b について、window 幅が 15 から 50 付近の時および 60 以上の時、全ての重み付き関数において MMAE の観点から工数見積もり精度の向上が確認できた。一方、window 幅が 15 より小さい時および 65 以上の時、Gaussian を除く全ての重み付き関数において MMAE の観点から工数見積もり精度の向上が確認できた。

重み付き関数の間の MMRE の違いに着目すると、図 2a では Triangular と Epanechnikov の間に明確な差は確認できなかった。一方、Gaussian と他の重み付き関数との間には工数見積もり精度の変化の違いが確認できた。同様の違いは図 2b においても確認できる。

window 幅が 65 付近より小さい場合、Gaussian は他の重み付き関数と大きな違いはないが、window 幅が 65 付近より大きくなると、Gaussian による工数見積もり精度は確認できなかった。

重み付き関数 Rectangular と Triangular の window 幅ごとの MMRE の一部を表 5 に示す。表 5 には重み付き関数間の MMRE の差に関する統計的検定の結果の p 値も記載している。表 5 において、有意水準 $\alpha = 0.1$ で有意であったのは $w = 10, 45, 60, 75$ である。図 2a からわか

るように、 $w = 10, 60$ では重み付けを行わない方が工数見積もり精度が高かった。一方、 $w = 45, 75$ では重み付け Triangular を行う方が工数見積もり精度が高い。今回の実験では、統計的に有意な差が生じた window 幅は 21 個あり、そのうち 12 個が重み付け Triangular を支持した。一方、9 個は重み付けなしを支持した。

以上のことから、CSC データセットにおいて、window 法とともに重み付け関数を用いて工数見積もり精度が有意に向上する場合があると言える。一方で、重み付けを行わない方が有意に工数見積もり精度が高くなる場合もあることも確認できた。

4.2 Maxwell データセット

図 3 に重み付きの有無による工数見積もり精度の差を示す。図 3a では window 幅が 20 の時および 25 より大きい時、全ての重み付き関数において MMRE の観点から工数見積もり精度の向上が確認できた。一方、window 幅が 20 より小さい場合、全ての重み付き関数において MMRE の観点から工数見積もり精度の低下が確認できた。

図 3b について、window 幅が 20 から 28 の時、Gaussian を除く全ての重み付き関数において MMAE の観点から工数見積もり精度の向上が確認できた。一方、window 幅が 20 より小さい場合、全ての重み付き関数において MMAE の観点から工数見積もり精度の低下が確認できた。

いずれの場合においても、window 幅が大きくなるにつ

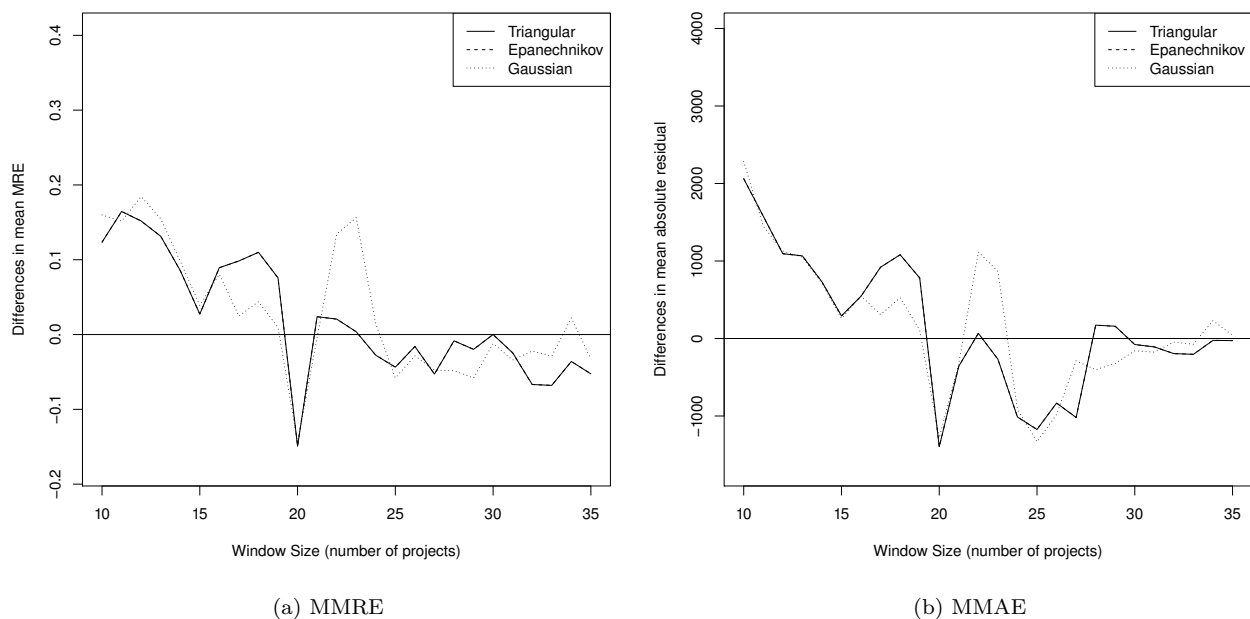


図 3: 線形回帰モデルとの見積もり精度の差 (Maxwell, window 法)

Fig. 3 Difference in estimation accuracy against non-weighted regression (Maxwell, windowing)

表 5: window 幅ごとの MMRE (CSC)
Table 5 MMRE with different window sizes (CSC)

window 幅	検証 データ数	Triangular	Rectangular	p-value
10	84	0.626	0.552	0.004
15	78	0.522	0.562	0.603
20	71	0.465	0.480	0.233
25	69	0.443	0.431	0.893
30	64	0.454	0.447	0.894
35	60	0.403	0.403	0.895
40	53	0.393	0.427	0.217
45	39	0.395	0.457	0.065
50	31	0.428	0.485	0.969
55	30	0.482	0.471	0.349
60	26	0.472	0.438	0.086
65	23	0.466	0.450	0.976
70	20	0.488	0.495	0.478
75	16	0.522	0.535	0.070

表 6: window 幅ごとの MMRE (Maxwell)
Table 6 MMRE with different window sizes (Maxwell)

window 幅	検証 データ数	Triangular	Rectangular	p-value
10	37	0.828	0.705	0.035
15	29	0.610	0.582	0.469
20	23	0.566	0.716	0.605
25	19	0.417	0.460	0.546
30	13	0.358	0.358	0.917
35	10	0.411	0.463	0.333

以上のことから, Maxwell データセットにおいて, window 法とともに重み付け関数を用いても工数見積もり精度は有意には向上しないと言える.

5. Threats to Validity

本研究における信頼性への脅威は既存研究と同様である. 具体的には以下の 2 点である.

まず, 第一に, 使用しているデータセットの数が少ない. このデータセットはある特定の組織から収集したものであり, 一般的なソフトウェア開発プロジェクトと同様の特徴を備えているとは必ずしもいえない. 従って, 他の組織において収集したデータセットを用いた場合に本研究と同様の結果を示すとはいえない. この点は工数見積もりモデルの研究全般にいえることである. しかしながら, はずれ値などが結果に影響を及ぼさないようにデータセットの選択

れて重み付き線形回帰モデルの相対的な工数見積もり精度が向上する傾向が確認できた.

重み付け関数 Rectangular と Triangular の window 幅ごとの MMRE の一部を表 6 に示す. p 値が示すように, 統計的に有意な差があったといえるのは, window 幅が小さい場合のみであった. window 幅が 20 の場合は p 値は 0.605 と有意ではなかった. MMAE についても同様の結果であった.

や前処理を入念に行っている．また，今回使用したデータセットは単一の組織から収集されたものとしてはサンプルサイズも大きく，収集期間も十分に長い．このため，少なくとも，本研究の結果はデータセットを収集した組織の特徴を十分に捉えていると考えている．

第二に，本研究で比較評価を行っている工数見積もりモデルの種類が少ない．本研究の比較評価の中に含まれていないものとしては，線形回帰モデルを発展させたリッジ回帰などの手法や回帰木が挙げられる [4], [9]．また，昨今では機械学習法に基づいた工数見積もりモデルにも注目が集まっている [17]．比較評価に用いる工数見積もりモデルの種類については今後の課題である．

6. 関連研究

ソフトウェア開発組織では開発プロジェクトのデータを収集し工数見積もりの根拠として用いている．収集活動を継続的に行うことでプロジェクトデータが組織に蓄積されていく．ある程度プロジェクトデータが蓄積されると，工数見積もりモデルの訓練を行うことが可能となる．ここで訓練とは，工数見積もりモデルに含まれるパラメータをプロジェクトデータを用いて調整することを指すことが多い．以上のような，実際のソフトウェア開発組織におけるプロジェクトデータの蓄積過程を模倣するため，Mendes と Lokan [13] はプロジェクトデータを時系列に沿って並べ，訓練用と検証用のデータセットに分割して工数見積もりモデルの評価を行った．

Mendes らの研究では，ある時点において終了している全てのプロジェクトデータを工数見積もりモデルの訓練に用いている．しかし，全てのプロジェクトデータを利用することが最も良い選択であるともいえない．ソフトウェア開発組織のパフォーマンスはプロセス改善などにより変化している可能性があるため，ある時点からみて一定以上過去のプロジェクトデータが組織の特徴を表しているとは必ずしもいえない．直近のプロジェクトデータのみ利用して，時間的な観点からみて訓練用プロジェクトデータを均質にした方が，工数見積もりの精度が高くなると期待できる．一方で，この方法では訓練に使用できるプロジェクトデータの数が減ってしまう．その結果，詳細なコスト要因の分析や精度が高い工数見積もりモデルの構築ができない場合がある．

Lokan と Mendes [11] は window 法 が工数見積もり精度に及ぼす影響について線形回帰モデルを用いて評価し，以下の結果を得た．

- window 法によって統計的に有意に見積もり精度が向上する
- window 幅がおおきくなるほど見積もり精度が向上する
- 見積もり精度の観点から「スイートスポット」が存在

する

Amasaki ら [2] は工数見積もりモデルとして類推法を用いて同様の比較評価を行った．類推法は線形回帰モデルと並びよく研究が行われている工数見積もりモデルである [3], [8], [16]．工数見積もりモデルの訓練方法において，類推法は線形回帰モデルと対照的である．線形回帰モデルの場合，あらかじめ訓練用のデータセット全てをもちいてモデルの調整を行う．このため，window 法を用いることで訓練用のプロジェクトデータの均質性が増し，見積もり精度が向上すると期待できる．一方，類推法では，まず，見積もり対象プロジェクトと類似した特徴を持つプロジェクトデータが訓練用のプロジェクトデータから選択される．そして選択されたプロジェクトデータに基づいて工数見積もりを行う．このように対照的な工数見積もり手法であるため，window 法の影響が異なっていると期待できる．実験の結果，window 法を用いる方がより高い精度を示す場合があることが確認されたが，統計的に有意とは言えない結果であった．

以上に述べたように，window 法には一定の効果が認められる．一方で，既存研究で用いられた window 法では，古いプロジェクトを含めるか含めないかの二択であり，本研究のように連続的な重み付けは行われていない．

7. おわりに

本研究では，従来の window 法と重み付き関数を組み合わせた重み付き window 法の効果を評価した．その結果，CSC データセットを用いた場合，一部の window 幅の場合において，統計的に有意な工数見積もり精度が向上することが確認できた．一方で，Maxwell データセットを用いた場合には統計的に有意な工数見積もり精度の向上は確認できなかった．

今回の実験では，重み付け関数が有効に作用するためには，対象のデータセットおよび window 幅を考慮にいれる必要性が示唆された．一方で，今回使用した重み付き関数は一部に限られたものであり，また，既存研究 [11] で使用された ISBSG データセットにおける実験も行っていない．これらの点に関する実験を行うことは今後の課題である．

参考文献

- [1] Amasaki, S. and Lokan, C.: The Effects of Moving Windows to Software Estimation: Comparative Study on Linear Regression and Estimation by Analogy, *IWSM-MENSURA '12: Proceedings of the 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 7th International Conference on Software Process and Product Measurement*, IEEE, p. (to appear.) (2012).
- [2] Amasaki, S., Takahara, Y. and Yokogawa, T.: Performance Evaluation of Windowing Approach on Effort Estimation by Analogy, *IWSM-MENSURA '11: Proceedings of the 2011 Joint Conference of the 21st In-*

- ternational Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement*, IEEE Computer Society, pp. 188–195 (2011).
- [3] Azzeh, M.: A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation, *Empirical Software Engineering*, Vol. 17, No. 1-2, pp. 90–127 (2011).
 - [4] Dejaeger, K., Verbeke, W., Martens, D. and Baesens, B.: Data Mining Techniques for Software Effort Estimation: A Comparative Study, *IEEE Transactions on Software Engineering*, Vol. 38, No. 2, pp. 375–397 (2011).
 - [5] Harrell, F. E.: *Regression Modeling Strategies*, Springer (2001).
 - [6] Jørgensen, M. and Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies, *Software Engineering, IEEE Transactions on*, Vol. 33, No. 1, pp. 33–53 (2007).
 - [7] Kitchenham, B., Lawrence Pfleeger, S., McColl, B. and Eagan, S.: An empirical study of maintenance and development estimation accuracy, *The Journal of Systems & Software*, Vol. 64, No. 1, pp. 57–77 (2002).
 - [8] Kocaguneli, E., Menzies, T., Bener, A. and Keung, J. W.: Exploiting the Essential Assumptions of Analogy-Based Effort Estimation, *Software Engineering, IEEE Transactions on*, Vol. 38, No. 2, pp. 425–438 (2012).
 - [9] Kocaguneli, E., Menzies, T. and Keung, J.: On the Value of Ensemble Effort Estimation, *Software Engineering, IEEE Transactions on*, No. 99, p. 1 (2011).
 - [10] Loader, C.: *Local Regression and Likelihood*, Statistics and Computing, Springer.
 - [11] Lokan, C. and Mendes, E.: Applying moving windows to software effort estimation, *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, IEEE Computer Society, pp. 111–122 (2009).
 - [12] Maxwell, K. D.: *Applied Statistics for Software Managers*, Prentice Hall (2002).
 - [13] Mendes, E. and Lokan, C.: Investigating the use of chronological splitting to compare software cross-company and single-company effort predictions: a replicated study, *Proceedings of the 13th Conference on Evaluation & Assessment in Software Engineering (EASE 2009)*, BCS (2009).
 - [14] Port, D. and Korte, M.: Comparative studies of the model evaluation criterions mmre and pred in software cost estimation research, *ESEM '08: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, ACM (2008).
 - [15] Premraj, R., Shepperd, M., Kitchenham, B. and Forselius, P.: An empirical analysis of software productivity over time, *Software Metrics, 2005. 11th IEEE International Symposium*, IEEE, pp. 10 pp.–37 (2005).
 - [16] Tsunoda, M., Kakimoto, T., Monden, A. and Matsumoto, K.: An empirical evaluation of outlier deletion methods for analogy-based cost estimation, *Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, ACM, p. 17 (2011).
 - [17] Wen, J., Li, S., Lin, Z., Hu, Y. and Huang, C.: Systematic literature review of machine learning based software development effort estimation models, *Information and Software Technology*, Vol. 54, No. 1, pp. 41–59 (2012).