
協調フィルタリングによる工数見積もり手法におけるデータ数と見積もり精度の関係の分析

Relationship between the Number of Data and Prediction Accuracy of CF-based Effort Estimation

柿元 健* 角田 雅照† 大杉 直樹‡ 門田 暁人§ 松本 健一¶

Summary. In recent years, collaborative filtering based (CF-based) estimation method has become an attractive means to estimate the software development effort. In CF-based method, the number of available project data greatly affects the estimation accuracy since it needs to find past projects that are similar to the estimation target from a given project data set. In this paper, we experimentally evaluated the relationship between the number of available data and the estimation accuracy of CF-based method. The result showed that MRE (mean relative error) of CF-based method becomes lower than that of the conventional regression model when the number of available data exceeds 50 projects.

1 はじめに

ソフトウェア開発プロジェクトにおいて、工数を見積もるための手法は数多く提案されている。工数の見積もりは、過去のプロジェクトで計測されたプロジェクトの特性値を用いて数学的な見積もりモデルを作成し、現行のプロジェクトで計測された特性値を作成されたモデルに代入することで見積もり値を算出することで行われる。モデルを作成するために用いるプロジェクトの特性値は、ソフトウェアメトリクス（または、単にメトリクス）と呼ばれる尺度のもとで計測、数値化される。

工数を見積もる1つの手段は、COCOMO [1] や SLIM [9] などの汎用的な（あらかじめ定義された）見積もりモデルを利用することである。例えば、COCOMO では、ソフトウェアの規模（ソースコード行数）の見積もり値から開発工数、期間、要員数、生産性を見積もり値をそれぞれ算出することができる。規模を与えるだけで容易に見積もりが行えるので、従来、COCOMO は多くの企業で用いられてきた。さらに、COCOMO を拡張した COCOMO II では、見積もり規模としてファンクションポイントを採用し、コストドライバと呼ばれるプロジェクトの特性値を与えることで、より正確な工数見積もりが行える。ただし、ソフトウェアの開発プロセスや開発手法は急速に多様化しており、これらのあらかじめ定義されたモデルでは開発組織やプロジェクトの個別性を十分に反映できず、高精度な見積もりを行うことが困難になってきている [3]。

工数を見積もるもう1つの手段は、重回帰分析やニューラルネット [13] などを用い、自社の過去プロジェクトのデータに特化した工数見積もりモデルを構築することである。過去プロジェクトのデータには、完了したプロジェクトごとに工数や開発規模、検出バグ数などの特性値が記録、蓄積されている。過去プロジェクトにより多くのプロジェクトを含むほど、また、より多種類の特性値を含むほど、より高い予測精度が得られることが期待される [7]。ただし、これらの見積もりモデルでは、

*Takeshi Kakimoto, 奈良先端科学技術大学院大学 情報科学研究科

†Masateru Tsunoda, 奈良先端科学技術大学院大学 情報科学研究科

‡Naoki Ohsugi, 奈良先端科学技術大学院大学 情報科学研究科

§Akito Monden, 奈良先端科学技術大学院大学 情報科学研究科

¶Ken-ichi Matsumoto, 奈良先端科学技術大学院大学 情報科学研究科

多様化するプロジェクトの特性値と工数との関係をただ1つのモデル式により表現するという点ではCOCOMOなどのあらかじめ定義されたモデルと同じであり、個別性の高いプロジェクトでは十分な予測精度を得ることが難しい。

近年、第三の手段として、プロジェクト間の類似性に基づく見積もり方法が注目されており、事例ベース推論(CBR) [12]、および、協調フィルタリング(CF-based 見積もり手法) [8] [15] が提案されている。これらの手法は、重回帰分析と同様に、過去プロジェクトのデータを必要とするが、データ全体に対してただ1つの見積もりモデルを構築するのではなく、見積もり対象プロジェクトごとに個別に見積もりモデルを構築するため、プロジェクトの個別性をより強く反映した予測が行える。一般に、ソフトウェア開発現場では、経験に頼った見積もり方法として、過去に手がけた類似の案件の実績データを元に、新規プロジェクトの工数を見積もる場合がある(「類推見積もり」などとも呼ばれている)が、CBRやCF-based 見積もり手法は、このような経験に頼った見積もりを、系統的に行う方法である。

CF-based 見積もり手法では、見積もり対象プロジェクトと類似する過去プロジェクトが存在する場合に高い精度で見積もりが行える。そのため、過去プロジェクトの数が多いほど類似プロジェクトが見つかる可能性が高くなり、見積もり精度が向上することが推測される。また、より多数の特性値が含まれるほど、類似プロジェクトであるか否かの判別が正確に行えることとなり、見積もり精度が向上すると期待できる。ただし、どの程度の数のプロジェクトや特性値を用いて見積もりを行えば高い精度で見積もりが行えるかは従来明らかとされていない。

本論文では、CF-based 見積もり手法のプロジェクト数、及び特性値数と見積もり精度の関係を実験的に評価し明らかにする。実験は、含まれるプロジェクト数と特性値数が異なる過去プロジェクトのデータを用いて、実測値が既知である複数のプロジェクトに対して見積もりを行った。そして、得られた見積もり値と実測値の誤差から見積もり精度の関係を調べた。

以降、2で見積もり手法について述べる。3で実験について説明した後、4で実験の結果とその考察について述べる。最後に5でまとめを述べる。

2 見積もり手法

2.1 協調フィルタリングによる見積もり手法

協調フィルタリングは、非常に多くのアイテム(書籍、楽曲、Webページなど)に対するユーザの評価などから何らかのアイテムを推薦するために一般的に用いられてきた [10] [11]。協調フィルタリングを工数の見積もりに対して適用した手法(CF-based 見積もり手法)が、文献 [8] [15] において提案されている。

CF-based 見積もり手法のようなプロジェクト間の類似性に基づいた見積もり手法には二つの特徴がある [4]。あらかじめ見積もりモデルを作成しないため、環境の変化が激しく一つの見積もりモデルが有効である時間が短いような状況においても、最新のデータの特徴を反映して予測が行えることと、ひとつの見積もりモデルを全ての事例に用いないため、個別の事例に対しても比較的適合しやすいことである。

CF-based 見積もり手法による見積もりは次の手順で行われる。

1. データセット中のプロジェクトにおける各特性値を平均値が0、分散が1となり、 $[-2, 2]$ の間にデータの約95%が含まれるように正規化し、特性値ごとの単位の違いが予測に影響を与えないようにする。プロジェクト*i*の特性値*j*の値 $m_{i,j}$ の正規化された値 $Z(m_{i,j})$ は式(1)で定義される。

$$Z(m_{i,j}) = \frac{m_{i,j} - \mu_j}{\sigma_j} \quad (1)$$

ここで、 μ_j は特性値*j*の平均値、 σ_j は特性値*j*の値の標準偏差である。

2. 予測対象のプロジェクトと類似した完了プロジェクトで工数が既知であるプロジェクトを見つけるため、プロジェクト間の類似度を求める。プロジェクトを特性値を成分としたベクトルとして扱い、ベクトル間の角度のコサインにより類似度を算出する手法である。プロジェクト i とプロジェクト a の類似度 $sim_{i,a}$ は式 (2) で定義される。

$$sim_{i,a} = \frac{\sum_{j \in M_i \cap M_a} Z(m_{i,j}) \times Z(m_{a,j})}{\sqrt{\sum_{j \in M_i \cap M_a} Z(m_{i,j})^2} \sqrt{\sum_{j \in M_i \cap M_a} Z(m_{a,j})^2}} \quad (2)$$

ここで、 M_i 、 M_a はプロジェクト i 、 a の欠損値でない特性値の集合である。

3. 類似プロジェクトの工数から予測対象のプロジェクトの工数の見積もり値を求める。プロジェクト i の特性値 j の値 $m_{i,j}$ の見積もり値 $\hat{m}_{i,j}$ は式 (3) で定義される。

$$\hat{m}_{i,j} = \frac{\sum_{k \in k \text{ nearestProjects}} (sim_{i,k} \times m_{k,j})}{\sum_{k \in k \text{ nearestProjects}} sim_{i,k}} \quad (3)$$

ここで、 $k \text{ nearestProjects}$ は、プロジェクト i との類似度が高い k 個のプロジェクトの集合である。本論文の実験では、同じデータセットを用いて行った実験 [5] において最良であった 14 とした。

各手順で用いるアルゴリズムは複数存在するが、本論文の実験において用いたアルゴリズムのみを示している。これらのアルゴリズムは、同じデータセットを用いて行った実験 [5] の事前実験において最良であったアルゴリズムの組み合わせである。

2.2 ステップワイズ重回帰分析

多変量解析の手法の一つである重回帰分析は、ソフトウェア開発見積もりモデルの作成において数多く用いられている [2] [16]。また、ステップワイズ重回帰分析は見積もり手法の比較対照としてしばしば用いられる [6]。

重回帰分析は、ある変数 (目的変数) と、それに対して影響すると考えられる複数の変数 (説明変数) の間の関係を一次式で表した見積もりモデルを作成し、作成された見積もりモデルに基づいて説明変数から目的変数を見積もる手法である。工数見積もりにおいては、見積もる工数が目的変数であり、見積もる工数以外の特性値が説明変数の候補となる。目的変数と説明変数の関係を表す一次式は、絶対誤差が最小二乗法で最小となるように定数が与えられる。そのため、モデル構築時に用いたデータで目的変数の分散が大きな場合には、目的変数の値が大きなケースにほど大きな影響を受けてしまう。

ステップワイズ重回帰分析は、ステップワイズ変数選択法により説明変数を決定し重回帰分析を行う手法である。ステップワイズ変数選択法は、目的変数に対して強く影響する変数を説明変数として選択する手法の一つである。ステップワイズ変数選択法による変数選択は次の手順で行われる [14]。

1. 初期モデルを作成する。ステップワイズ変数選択法には、変数増減法と変数減増法があるが、変数増減法では、変数を全く含まないモデルを初期モデルとして作成し、変数減増法では、全ての変数を含むモデルを初期モデルとして作成する。

表 1 基礎データの各特性値の統計量

	設計工数	製造工数	試験工数	開発規模	設計工数 (正社員)
平均値	1.153	2.0613	1.446	233.047	0.134
中央値	0.085	0.134	0.0703	43.9	0.0171
分散	47.455	174.62	101.076	1092691.039	0.608
最大値	67.458	144.036	111.876	11301	8.724
最小値	0.001	0.001	0.001	0.4	0
	設計工数 (派遣)	設計工数 (外注)	製造工数 (正社員)	製造工数 (派遣)	製造工数 (外注)
平均値	0.696	0.0096	0.034	0.117	1.507
中央値	0.0421	0	0.0025	0.026	0
分散	25.808	0.002	0.012	0.067	152.447
最大値	58.734	0.488	0.981	1.586	144.036
最小値	0	0	0	0	0

基礎データの値は特性値ごとに何らかの値であらかじめ除算されている

- 作成されたモデルに対して、各説明変数の係数が 0 でないかの検定を行い、指定した有意水準で棄却されない場合に変数を選択する。ただし、多重共線性を回避するために、選択する変数の分散拡大要因が一定数以上の場合、またはその変数を選択することによって、他の変数の分散拡大要因が一定数以上となる場合、その変数は選択しない。
- 検定により適切な変数が選択されたと判断されるまで手順 2 を繰り返す。

3 実験

3.1 実験で利用したデータ

実験で利用した基礎となるデータ（基礎データ）は、プロジェクト数 140 件、特性値数 10 個のデータの欠損を含まないデータである。基礎データに含まれる 10 個の特性値とは、設計工数、製造工数、試験工数、開発規模、設計工数（正社員分）、設計工数（派遣社員分）、設計工数（外注分）、製造工数（正社員分）、製造工数（派遣社員分）、製造工数（外注分）である。基礎データは、ある企業で得られたプロジェクト数 1081 件、特性値数 14 件、データ全体の欠損の割合 59.83% のデータからデータの欠損を含まないように抽出したものである。また、全ての値が特性値ごとに何らかの値であらかじめ除算されているデータである。

基礎データに含まれるプロジェクトは、新規開発、機能拡張、パッケージのカスタマイズといった様々な開発形態のプロジェクトが含まれており、約 6 年の間に実施されたプロジェクトが含まれている。また、プロジェクトの実施期間も数週間から数年までのプロジェクトが含まれている。基礎データにおける基礎データの各特性値の統計量を表 1 に示す。

3.2 評価基準

評価基準として、絶対誤差、相対誤差のそれぞれの平均値、中央値の四種類の誤差を用いて評価を行った。各評価基準は値が小さいほど見積もり精度が高いことを表す。

それぞれの評価基準は以下の式 (4) ~ (7) で計算される。ここで、M 件のプロジェクトがあるとすると、また、実測値と見積もり値をそれぞれ X_i 、 \hat{X}_i ($i = 1 \sim M$) とし、 $A_i = |\hat{X}_i - X_i|$ 、 $R_i = \frac{|\hat{X}_i - X_i|}{X_i}$ とおく。

絶対誤差平均値

$$\text{絶対誤差平均値} = \frac{\sum_{i=1}^M A_i}{M} \quad (4)$$

絶対誤差中央値

$$\text{絶対誤差中央値} = \begin{cases} A_n & M = \text{奇数 } (2n - 1) \\ & (A_1 \leq A_2 \leq \dots \leq A_n \leq \dots \leq A_{2n-1}) \\ \frac{A_n + A_{n+1}}{2} & M = \text{偶数 } (2n) \\ & (A_1 \leq \dots \leq A_n \leq A_{n+1} \leq \dots \leq A_{2n}) \end{cases} \quad (5)$$

相対誤差平均値

$$\text{相対誤差平均値} = \frac{\sum_{i=1}^M R_i}{M} \quad (6)$$

相対誤差中央値

$$\text{相対誤差中央値} = \begin{cases} R_n & M = \text{奇数 } (2n - 1) \\ & (R_1 \leq R_2 \leq \dots \leq R_n \leq \dots \leq R_{2n-1}) \\ \frac{R_n + R_{n+1}}{2} & M = \text{偶数 } (2n) \\ & (R_1 \leq \dots \leq R_n \leq R_{n+1} \leq \dots \leq R_{2n}) \end{cases} \quad (7)$$

3.3 実験手順

実験は、プロジェクト数を変化させた実験と、特性値数を変化させた実験の二種類の実験を行った。それぞれの実験は次の手順で行った。

- プロジェクト数の変化
 1. 3.1 で述べた基礎データを無作為に 70 件ずつに二等分し、見積もりモデルを作成するデータ（以降フィットデータと呼ぶ）とフィットデータを用いて実際に見積もりを行うデータ（以降テストデータと呼ぶ）とした。
 2. プロジェクト数が異なるデータを作成するために、フィットデータから無作為に 10 回 n 個のプロジェクトを抽出し新たなフィットデータを作成した。 n は 2 ~ 69 まで変化させた。
 3. 全てのフィットデータを使って一つのテストデータに対して CF-based 見積もり手法により見積もりを行い、各評価基準を求めた。見積もり対象の特性値を試験工数とし、テストデータの試験工数は未知数として見積もりを行った。見積もり時に式 (3) の k *nearestProjects* は 14 としているが、プロジェクト数が 14 以下のフィットデータの場合には、 k *nearestProjects* の値はプロジェクト数として見積もりを行った。
 4. ステップワイズ重回帰分析でも同様に見積もりを行った。

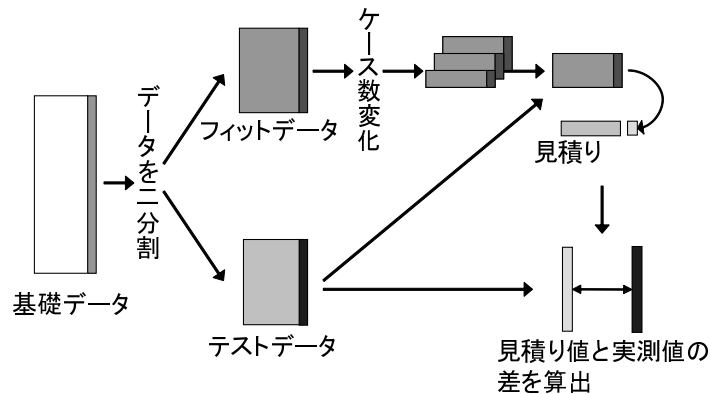


図1 プロジェクト数の変化の実験におけるデータの加工手順

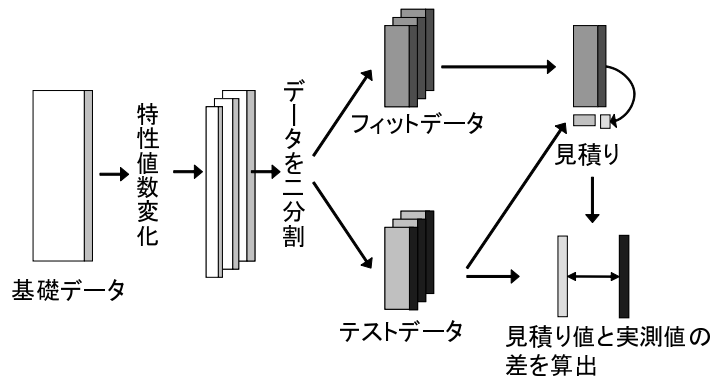


図2 特性値数の変化の実験におけるデータの加工手順

● 特性値数の変化

1. 3.1 で述べた基礎データを，時系列を考慮し，開発規模，設計工数，設計工数(正社員分)，設計工数(派遣社員分)，設計工数(外注分)，製造工数，製造工数(正社員分)，製造工数(派遣社員分)，製造工数(外注分)の順で特性値を増加させ，特性値数異なるデータを作成した．
2. 作成した特性値数異なるデータを無作為に均等に70件ずつに二等分しフィットデータとテストデータとし，新たなフィットデータとテストデータの組を作成した．
3. 全てのフィットデータとテストデータの組に対してCF-based見積もり手法により見積もりを行い，各評価基準を求めた．見積もり対象の特性値を試験工数とし，テストデータの試験工数は未知数として見積もりを行った．
4. ステップワイズ重回帰分析でも同様に見積もりを行った．

実験におけるデータ加工の手順を示した図を図1と図2に示す．

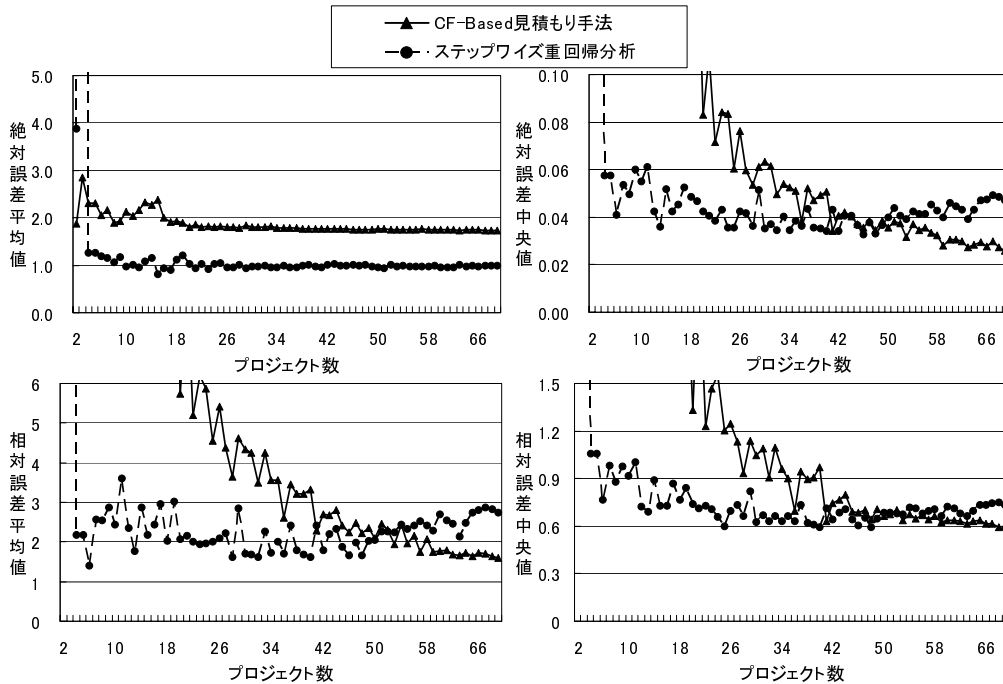


図 3 各プロジェクト数における見積もり精度

4 実験の結果と考察

4.1 プロジェクト数の変化

プロジェクト数を变化させた時の CF-based 見積もり手法とステップワイズ重回帰分析の各評価基準の見積もり精度の変化を図 3 に示す。各グラフの横軸はプロジェクト数を、縦軸は評価基準の値を示す。

各評価基準で共通して、CF-based 見積もり手法では、プロジェクト数の増加に伴って、徐々に見積もり精度が向上し、また、精度の変動幅も小さくなっている。ステップワイズ重回帰分析では、あるプロジェクト数で見積もり精度が急激に向上し、しばらく向上が続くが、その後、徐々に見積もり精度が低下している。また、絶対誤差平均値を除いて、プロジェクト数が約 50 件を超えるとステップワイズ重回帰分析よりも高い見積もり精度が得られている。

この結果より、CF-based 見積もり手法では、プロジェクト数が少ない時には似ているプロジェクトも少ないために見積もり精度が低いですが、プロジェクト数の増加に伴って似ているプロジェクトも多くなるため見積もり精度が向上することを示している。ステップワイズ重回帰分析では、見積もり精度の変化が向上から低下へと変わった時点で最適な見積もりモデルが作成されているが、その後は、プロジェクトが増加することで最適な見積もりモデルから外れていくことがわかる。

絶対誤差平均値において、CF-based 見積もり手法の見積もり精度がステップワイズ重回帰分析の見積もり精度よりも悪い結果となっている。これは、重回帰分析が絶対誤差平均値が最小となるように見積もりモデルを作成し、また、大きな値のデータに大きく影響を受けるのに対して、CF-based 見積もり手法では誤差を最小にするようなモデルを作成するわけではないため、大きな値のデータを予測する場合には絶対誤差が必然的に大きくなっているためだと考えられる。これは、絶対誤差平均値以外では、プロジェクト数が多い場合にはステップワイズ重回帰分析よりも高

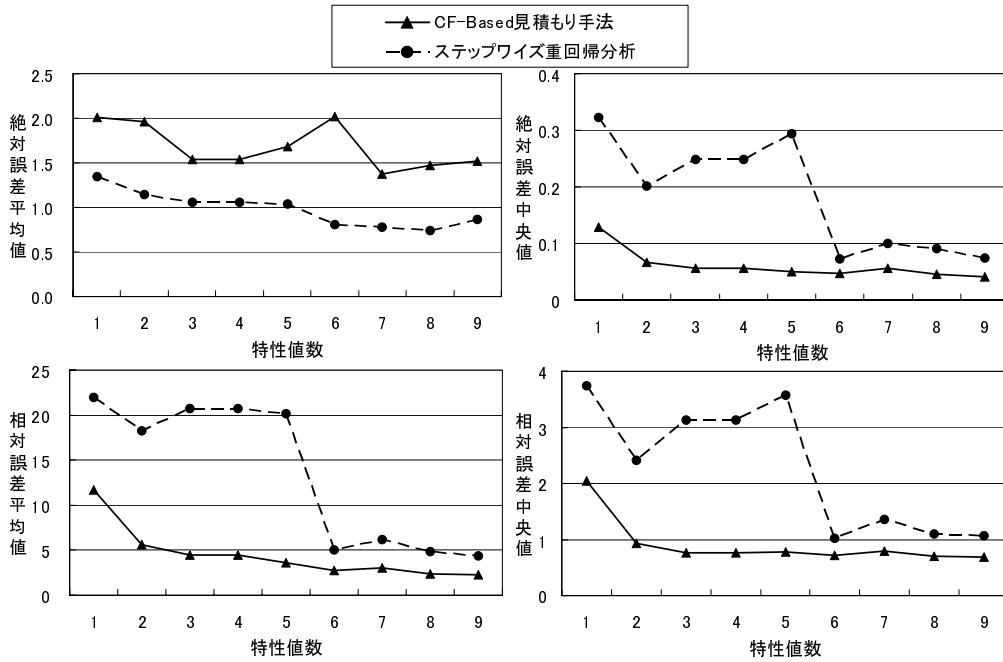


図4 各特性値数における見積もり精度

い見積もり精度を示していることから推測される。

絶対誤差、相対誤差ともに、平均値と中央値を比較すると中央値の見積もり結果の方が高い見積もり精度が得られている。これは、見積もり結果中に見積もり精度が著しく低い結果が一部含まれているために、平均値が中央値よりも悪くなっていると考えられる。

4.2 特性値数の変化

特性値数を変化させた時の CF-based 見積もり手法とステップワイズ重回帰分析の見積もり精度の変化を図4に示す。各グラフの横軸は特性値数を、縦軸は評価基準を示す。

絶対誤差平均値を除いて、CF-based 見積もり手法では、特性値数の増加に伴って多くの場合に見積もり精度が向上しているのに対して、ステップワイズ重回帰分析では一部の特性値が追加された場合において見積もり精度が著しく向上し、それ以外の特性値が追加された場合には見積もり精度の変動は小さい。この結果より、CF-based 見積もり手法では、特性値を増加させても見積もり精度が大きく低下することはないが、ステップワイズ重回帰分析では、特性値の増加によって見積もり精度が大きく低下する可能性があることがわかる。

また、プロジェクト数の変化と同様に、絶対誤差平均値では、他の評価基準と全く異なる傾向を示し、ステップワイズ重回帰分析の見積もり精度が高い結果が得られている。また、絶対誤差、相対誤差ともに、平均値よりも中央値で高い見積もり精度が得られている。これらの原因は、プロジェクト数の変化と同様であると考えられる。

4.3 考察

実験の結果より、CF-based 見積もり手法は、プロジェクトデータの数の増加に伴って見積もり精度は向上している。ただし、プロジェクト数が少ない場合には、

ステップワイズ重回帰分析の方が高い見積もり精度を示している。また、プロジェクト数が著しく少ない場合には、CF-based 見積もり手法、ステップワイズ重回帰分析とも高い見積もり精度は得られない。そこで、見積もりに用いるプロジェクト数によって使用する見積もり手法を変えることで、より高い見積もり精度で見積もりを行うことが可能となると考えられる。そのためには、より多くのデータセットを用いた実験が必要であると考えられる。

特性値数の変化に対しては、変数選択を行っていない CF-based 見積もり手法で、変数の追加に伴い見積もり精度が徐々に向上している。一方、変数選択を行っているステップワイズ重回帰分析で、変数が追加されることで見積もり精度が悪化する場面がある。これより、多重共線性を考慮しているにも関わらず、ステップワイズ重回帰分析の変数選択が有効に働いていない可能性が考えられる。ただし、今回の実験では実験データの特性値数が少ないため、最大で 9 個の特性値を用いた見積もりしか行っていないため、今後はより多くの特性値を含むデータを用いて実験を行う必要があると考えられる。

実験で使用した実験データに含まれる特性値は、規模と工数に関する特性値であるため、それぞれの特性値間には少なからず依存関係があると考えられる。また、設計工数と製造工数では、総計とは別に正社員分、派遣社員分、外注分の個別の工数が特性値として含まれており、これらの間の依存関係は特に高いと考えられる。実験結果において、CF-based 見積もり手法では個別の工数が追加された場合でも、一部を除いて見積もり精度が向上している。一方、ステップワイズ重回帰分析では個別の工数が追加されることで見積もり精度が低下している。これは、ステップワイズ重回帰分析では一般的に依存関係のある変数を用いて予測した場合には予測精度が低下するためだと考えられる。これらの結果より、CF-based 見積もり手法では、変数間に依存関係がある場合でも予測精度は低下しないと言える。工数見積もりで使用するデータでは、それぞれ依存関係がある特性値が含まれている場合が多いため、CF-based 見積もり手法は有効であると考えられる。

両方の実験において、絶対誤差平均値の結果が他の評価基準の結果と大きく異なっている。これは、実験の結果でも述べたように、ステップワイズ重回帰分析が絶対誤差平均値が最小となるようにモデルを作成するためだと考えられる。他の評価基準での結果ではグラフはほぼ同一の形を示しているため、見積もり精度を確かめる際に絶対誤差平均値だけを用いることは適切ではないと言える。ただし、絶対誤差平均においてもステップワイズ重回帰分析を超える見積もり精度が得られることが望ましい。

5 まとめ

本論文では、協調フィルタリングによる工数見積もり手法 (CF-based 見積もり手法) のプロジェクト数、及び特性値数と見積もり精度の関係を実験的に評価し明らかとした。実験の結果、CF-based 見積もり手法は、プロジェクトの数の増加に伴って見積もり精度が向上し、プロジェクト数が約 50 件を超えるとステップワイズ重回帰分析よりも見積もり精度が高くなり、また、特性値数の増加に伴って見積もり精度が向上し、ステップワイズ重回帰分析よりも見積もり精度が高くなるという結果が得られた。

今後は、より多くのデータセットを用いて実験を行うことで、実験結果を一般性を持つ結果とする予定である。また、ステップワイズ重回帰分析以外の見積もり手法とも比較を行い、CF-based 見積もり手法のアルゴリズムも今回用いたアルゴリズムを拡張したアルゴリズムを用いる予定である。

参考文献

- [1] B.W. Boehm, Software engineering economics, Prentice Hall, New Jersey, 1981.

- [2] L. Briand, T. Langley, and I. Wiczorrek, "A replicated assessment and comparison of common software cost modeling techniques," In Proc. 22nd IEEE International Conf. on Software Eng., Limerick, pp.377-386, 2000.
- [3] S. Chulani, B. Boehm, and B. Steece, "Bayesian analysis of empirical software engineering cost models," IEEE Trans. on Software Eng., Vol.25, No.4, pp.573-583, 1999.
- [4] C. Hayes, P. Cunningham, and B. Smyth, "A case-based reasoning view of automated collaborative filtering," Proc. 4th Int'l Conf. on Case-Based Reasoning, pp.243-248, 2001.
- [5] 柿元健, 角田雅照, 大杉直樹, 門田暁人, 松本健一, "協調フィルタリングに基づく工数見積もりのロバスト性評価," ソフトウェア工学の基礎 XI, 日本ソフトウェア科学会 FOSE2004, pp.73-84, Nov. 2004.
- [6] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, "A comparative study of cost estimation models for web hypermedia applications," Empirical Software Engineering, vol.8, issue 2, pp.163-196, 2003.
- [7] I. Myrvtveit, and E. Stensrud, "A controlled experiment to assess the benefits of estimating with analogy and regression models," IEEE Trans. on Software Eng., Vol.25, No.4, pp.510-525, 1999.
- [8] N. Ohsugi, M. Tsunoda, A. Monden, K. Matsumoto, "Applying collaborative filtering for effort estimation with process metrics," 5th International Conference on Product Focused Software Process Improvement, Lecture Notes in Computer Science, Vol.3009, pp.274-286, Kyoto, Japan, 2004.
- [9] L. H. Putnam, "A general empirical solution to the macro sizing and estimating problem," IEEE Trans. on Software Eng., Vol.4, No.4, pp.345-361, 1971.
- [10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," Proc. ACM Conf. Computer Supported Cooperative Work, pp.175-186, Chapel Hill, North Carolina, United States, 1994.
- [11] B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," Proc. 10th International World Wide Web Conference , pp.285-295, Hong Kong, 2001.
- [12] M. Shepperd, and C. Schofield, "Estimating software project effort using analogies," IEEE Trans. Software Eng., vol.23, no.12, pp.76-743, 1997.
- [13] K. Srinivasan, and D. Fisher, "Machine learning approaches to estimating software development effort," IEEE Trans. Software Eng., vol.21, no.2, pp.126-137, 1995.
- [14] 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," p.240, 共立出版, 東京, 1995.
- [15] 角田雅照, 大杉直樹, 門田暁人, 松本健一, "協調フィルタリングを用いたソフトウェア開発工数予測方法," 情処学論, Vol.46, No.5, pp.1156-1164, 2005.
- [16] C. Walston, and C. Felix, "A method of programming measurement and estimation," IBM Systems Journal, Vol.1, pp.54-73, 1977.