

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

角 田 雅 照[†] 大 杉 直 樹[†] 門 田 暁 人[†]
松 本 健 一[†] 佐 藤 慎 一^{††}

ソフトウェア開発における工数予測を目的として、過去のソフトウェア開発プロジェクトにおいて記録された多種類のソフトウェアメトリクス値を入力データとし、協調フィルタリングにより予測工数を求める方法を提案する。協調フィルタリングは、未計測の値（欠損値）が大量に含まれているデータを入力とした場合でも予測が行えるという特長があるが、ソフトウェア工数予測に適用する方法はこれまで提案されていない。提案方法では、まず、入力となるメトリクス値を正規化し、値域を揃える。次に、正規化したメトリクス値を用いて、予測対象（開発中）のプロジェクトと、過去に行われたプロジェクトとの類似度を計算する。最後に、類似度の高い（予測対象プロジェクトと類似した）プロジェクトの工数を類似度で加重平均した値を、予測対象プロジェクトの工数とする。ケーススタディとして、株式会社 NTT データにおいて 1,081 件のソフトウェアプロジェクトから計測された 14 種類のメトリクス（約 60% の欠損値を含む）を用いて試験工数を予測した。その結果、提案方法は従来方法（欠損値処理法を用いたステップワイズ重回帰分析）よりも高い精度を示し、予測試験工数の相対誤差の平均値（1 プロジェクトあたり）が 22.11 から 0.79 に改善された。

Software Development Effort Prediction Based on Collaborative Filtering

MASATERU TSUNODA,[†] NAOKI OHSUGI,[†] AKITO MONDEN,[†]
KEN-ICHI MATSUMOTO[†] and SHIN-ICHI SATO^{††}

To predict software development effort, this paper proposes an effort prediction method based on the *Collaborative Filtering (CF)* which uses as input various software metrics recorded in past software development projects. The CF has an advantage that it can conduct a prediction using “defective” input data containing a large amount of missing values. There are, however, no researches which propose a method for applying the CF to Software effort prediction. Our proposal consists of three steps. In the first step, we normalize values of metrics to equalize their value range. In the next step, we compute the similarity between target (current) project and past (completed) project using normalized values. In the last step, we estimate the effort of target project by computing the weighted sum of efforts of high-similarity projects (that are similar to the target project) using the similarity of each project as a weight. In a case study to evaluate our method, we predicted the test process effort using 1,081 software projects including 14 metrics whose missing value rate is 60%, which have been recorded at NTT DATA Corporation. As a result, the accuracy of our method showed better performance than conventional methods (stepwise multiple regression models); and, the average accuracy per project was improved from 22.11 to 0.79.

1. はじめに

ソフトウェアプロジェクト管理において、納期の遅れ、コストの超過といったプロジェクトの失敗を未然に防ぐためには、工数予測を行うことが重要である¹³⁾。そのために、工数予測モデルが数多く提案されてき

た^{1),3),7),17),18)}。工数予測モデルとは、プロジェクトの特性値（開発規模、欠陥数など）を説明変数とし、目的変数である工数との関係を数学的に表すものである。プロジェクトの特性値はソフトウェアメトリクスによって計測、数値化され、過去のプロジェクトで計測されたメトリクス値に基づいてモデルの構築（パラメータ推定）が行われる。一般に広く用いられている重回帰分析やニューラルネットなどの多変数モデルでは、どの特性値を説明変数として用いるかはあらかじめ定められていない^{5),8),25)}。推定精度の高いモデルを構築する 1 つの方法は、できるだけ多数の特性値を説

[†] 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute
of Science and Technology

^{††} 株式会社 NTT データ
NTT DATA Corporation

明変数の候補とすることであり、そのために、多数のメトリクスによる特性値計測が行われる。

ただし、ソフトウェア開発プロジェクトを対象とした多数のメトリクスによる計測には、欠損値の問題がある。欠損値の問題とは、過去のプロジェクトにおいて計測されるべきメトリクス値に未計測のもの（欠損値）が数多く存在するために、工数予測モデルを構築しにくくなることである²³⁾。一般に、多数のプロジェクトから多種類のメトリクスを計測して得られるデータセットには欠損値が含まれる⁵⁾。我々は、株式会社 NTT データと奈良先端科学技術大学院大学の産学連携研究を通じて、1,081 件のプロジェクトから 14 種類のメトリクス値を計測したが、その約 60% の値が欠損値であった（3.3 節参照）。欠損値の生じる理由は様々である。企業内の部署ごとに計測対象のメトリクスが異なる場合には、部署間のデータを結合した際に欠損値が生じる。また、同一の部署でも、時期によって計測対象のメトリクスが異なる場合があり、欠損値の原因となる。さらに、メトリクス計測時の時間的制約により、未計測の値が生じることも多い^{6),19)}。一方、不必要に多くのメトリクス値を計測することは、プロジェクトのコストの増大を招くため、多くの開発現場において許容されない。以上のような理由から、長年にわたって多数のプロジェクトから多種類のメトリクス値を過不足なく計測することは容易でないといえる。

本論文では、欠損値の有無や多寡にかかわらず工数予測を行うための方法として、協調フィルタリングにより工数予測を行う方法を提案する。協調フィルタリングは、欠損値を大量に含んだデータを入力した場合でも予測が行えるという特長があり、これまで情報検索の分野においてさかんに研究されてきた。たとえば、amazon.com では、読書後の書籍に対する各ユーザの 5 段階評価（好き 5, ..., 嫌い 1）に基づいて、協調フィルタリングによりユーザが好む書籍の予測を行うが、各ユーザが評価した書籍が amazon.com に登録されている全書籍の 1% に満たない（すなわち、残りの 99% 以上が欠損値となる）場合でも予測を行える。協調フィルタリングをソフトウェア工数予測に適用することで、未計測のメトリクス値が多数含まれる場合でも予測が可能になると期待されるが、工数予測に適用する具体的な方法はこれまで提案されていない。

提案方法では、まず、入力となるメトリクス値を正規化し、値域を揃える。次に、正規化したメトリクス値を用いて、予測対象（開発中）のプロジェクトと、過去に行われた各プロジェクトとの間の類似度を計算する。類似度計算は、2 つのプロジェクトの両方で計

測された（すなわち、欠損していない）メトリクス値のみを用いて行われる。最後に、類似度の高い（予測対象プロジェクトと類似した）プロジェクトの工数を類似度で加重平均した値を、予測対象プロジェクトの工数とする。

従来の予測モデルでは、欠損値を含んだデータセットを直接用いてモデルを構築することはできず、欠損値処理法を併用する必要があった¹¹⁾。欠損値処理法とは、与えられたデータセットから欠損値を含むプロジェクトを除外したり、欠損値を何らかの値で補完したりすることによって、欠損値を含まないデータセットを作成する方法である。ただし、欠損値の多寡や分布にかかわらず特定の欠損値処理方法を採用することは望ましくない^{12),19)}。一方、協調フィルタリングに基づく提案方法は、欠損値を含まないデータセットだけでなく、多数の欠損値（たとえば 50% 以上）が含まれたデータセットを用いた場合でも、欠損値の除外や補完を行うことなしに工数予測を行える点異なる。

本論文では、提案方法を用いたケーススタディについても報告する。ケーススタディで用いたデータは、株式会社 NTT データにおいて約 10 年間にわたって 1,081 件のソフトウェアプロジェクトから計測されたメトリクス値の集合であり、約 60% の欠損値が含まれていた。提案方法の性能を評価するために、試験工数の予測を行った。また、欠損値処理法を用いたステップワイズ重回帰分析と提案方法との性能比較を行った。

本論文は、産業界における欠損値の問題に対し、企業の実データを用いて研究を進め、アカデミックな 1 つの解決策を与えたという点において、産学連携研究の成功事例の 1 つであると考える。

以降、2 章では協調フィルタリングを用いた予測方法を提案する。3 章ではケーススタディの方法と手順について説明する。4 章ではケーススタディの結果と、結果に対する考察および本論文の限界について議論する。5 章では関連研究を説明する。最後に 6 章でまとめと今後の課題について述べる。

2. 協調フィルタリングを用いた予測方法

2.1 従来の協調フィルタリングとその工数予測への応用

協調フィルタリングは、多量に存在するアイテムの中からユーザの好みに合うと予測されるアイテム情報を選出して推薦するシステム（以降、推薦システム）の基盤技術として用いられている。ここでのアイテムとは、記事、ウェブページ、書籍、楽曲、映画作品など、ユーザによって好みの度合いが異なるものである。

これまで、数多くの推薦システム、および、協調フィルタリングのアルゴリズムが提案されている^{4),9),16)}。

協調フィルタリングに基づく推薦システムは、典型的に、(1) ユーザ間の類似度の計算、(2) 類似度に基づく予測値計算、という2つの手順を含む。この2つの手順に従って推薦を行うシステムとして、Resnickら¹⁴⁾のGroupLensがあげられる。GroupLensは、Usenetにある多数のニュース記事から、ユーザの好みに合うと予測される記事を選び出して推薦するシステムである。GroupLensを利用するすべてのユーザは、読後の記事に対して、5段階（好き5、…、嫌い1）の評価を行う。システムは、その評価を基にユーザ間の類似度を求める。ユーザが高い評価を行った記事は、類似度の高い他のユーザにも推薦される。すなわち、「ある記事に対して同様の評価をするユーザ同士は、他の記事に関しても同様の評価をするであろう」と仮定して、自分と類似した評価をしているユーザが高い評価をした記事の推薦を行うのである。

本論文では、協調フィルタリングに基づく推薦システムの考え方をソフトウェアプロジェクトの工数予測に適用し、「類似するプロジェクトどうしは、工数も類似しているであろう」と仮定し、プロジェクト間の類似度を計算し工数予測を行う。

提案方法では、ユーザの評価値の代わりに、プロジェクトのメトリクス値を用いる。ただし、ユーザの評価値は[1, 5]などの一定の値域を持つのに対し、ソフトウェアメトリクスはその種類ごとに値域が異なる。そのため、類似度計算を行う際に何らかの補正が必要となる。また、ユーザの評価値は、ユーザごとの平均値のばらつきが小さい（どのユーザも評価値の平均は3に近い値となる）のに対し、ソフトウェアプロジェクトの規模は多様である（分散が著しく大きい）ため、プロジェクトごとのメトリクスの平均値のばらつきがきわめて大きくなる。我々の予備調査では、この特徴がソフトウェア工数の予測値に影響を及ぼし、従来の協調フィルタリングの予測値計算方法では十分な精度が得られなかった。本論文では、これらの点を考慮した工数予測方法を提案する。

協調フィルタリングをソフトウェアの工数予測に適用できれば、欠損値を含んだデータセットを用いた場合にも予測値の計算が可能となる。協調フィルタリングでは、欠損値を非常に多く含むデータを用いて、ユーザの嗜好を予測することが前提となっている。一般的な推薦システムでは、ほぼすべてのユーザは全アイテムの1%未満しか評価していない。これは存在するアイテムの量が莫大であることによる（たとえば、

	m_1	m_2	...	m_j	...	m_b	...	m_n
p_1	$v_{1,1}$	$v_{1,2}$...	$v_{1,j}$...	$v_{1,b}$...	$v_{1,n}$
p_2	$v_{2,1}$	$v_{2,2}$...	$v_{2,j}$...	$v_{2,b}$...	$v_{2,n}$
...
p_i	$v_{i,1}$	$v_{i,2}$...	$v_{i,j}$...	$v_{i,b}$...	$v_{i,n}$
...
p_a	$v_{a,1}$	$v_{a,2}$...	$v_{a,j}$...	$v_{a,b}$...	$v_{a,n}$
...
p_m	$v_{m,1}$	$v_{m,2}$...	$v_{m,j}$...	$v_{m,b}$...	$v_{m,n}$

図1 予測に用いる $m \times n$ 行列

Fig.1 $m \times n$ table used for prediction.

あるユーザがGroupLensにあるUsenetのニュースすべて（またはその大部分）を評価することは事実上不可能である）。そのため、協調フィルタリングでは、欠損値を考慮した類似度計算が行われる。具体的には、2人のユーザの両方が評価したアイテムの評価値のみを用いてユーザ間の類似度を計算する。同様に、本論文では、2つのプロジェクト間で共通に計測されているメトリクスのみを用いて類似度を計算することで、欠損値を非常に多く含むデータセットに対する予測を可能とする。

2.2 提案する工数予測方法

提案する予測方法では、図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。図中、 $p_i \in \{p_1, p_2, \dots, p_m\}$ は i 番目のプロジェクトを表し、 $m_j \in \{m_1, m_2, \dots, m_n\}$ は j 番目のメトリクスを表す。すなわち、図1の行がプロジェクト、列がメトリクスを表している。 $v_{i,j} \in \{v_{1,1}, v_{1,2}, \dots, v_{m,n}\}$ はプロジェクト p_i で計測されたメトリクス m_j の値を表す。メトリクス $v_{i,j}$ が未計測（欠損値）の場合、 $v_{i,j} = \varphi$ と記す。

提案する予測方法は、工数に限らず任意のメトリクス値の予測が可能である。ここでは、 a 番目のプロジェクト p_a を予測対象のプロジェクトとし、このプロジェクトにおける b 番目のメトリクス値 $v_{a,b}$ を予測することを目的とし、 $v_{a,b}$ の予測値を $\hat{v}_{a,b}$ と記す。以下、 $\hat{v}_{a,b}$ を予測するための3つのステップについて述べる。

ステップ1（メトリクスの正規化）：プロジェクトに含まれる各メトリクスは値域が異なるため、最初にメトリクス値の正規化を行い、値域を $[0, 1]$ にする。（プロジェクト p_i の）メトリクス値 $v_{i,j}$ の正規化された値 $normalized(v_{i,j})$ を次式によって求める：

$$normalized(v_{i,j}) = \frac{v_{i,j} - \min(P_j)}{\max(P_j) - \min(P_j)} \quad (1)$$

ここで、 P_j はメトリクス m_j が計測されているプロジェクトの集合を表し、 $\max(P_j)$ と $\min(P_j)$ はそれぞれ $\{v_{x,j} | p_x \in P_j\}$ の最大値、最小値を表す。

ステップ 2 (プロジェクト間の類似度計算): このステップにおいて、予測対象プロジェクト p_a と他のプロジェクト p_i との類似度 $\text{sim}(p_a, p_i)$ を求める。従来の協調フィルタリングの研究において、相関係数を用いた類似度計算アルゴリズム¹⁴⁾ や、ベクトルを用いた類似度計算アルゴリズム⁴⁾ など、数多くのアルゴリズムが提案されている^{9),16)}。我々の予備実験において、類似度計算アルゴリズムを様々に交換して工数予測を行った結果、ベクトルを用いたアルゴリズムの精度が最も高かった。そこで、本論文では、このアルゴリズムを採用する。

ベクトルを用いた類似度計算アルゴリズムは、2つの文書間の類似度を計算するために用いられている¹⁵⁾。文書間の類似度計算では、2つの文書に現れる単語の頻度を要素として2つのベクトルを作成し、ベクトルのなす角のコサインを用いて類似度を計算する。Breese ら⁴⁾ は文書をユーザ、単語をアイテム、単語の頻度をアイテムの評価値と見なし、このアルゴリズムに基づく推薦システムを提案した。本論文では、ユーザをプロジェクト、アイテムをメトリクス、アイテムの評価値をメトリクス値と見なし、プロジェクト間の類似度計算に適用する。

予測対象プロジェクト p_a と他のプロジェクト p_i との類似度 $\text{sim}(p_a, p_i)$ を次式のように定義する:

$$\begin{aligned} \text{sim}(p_a, p_i) &= \frac{\sum_{j \in M_a \cap M_i} (\text{norm}(v_{a,j}) \times \text{norm}(v_{i,j}))}{\sqrt{\sum_{j \in M_a \cap M_i} (\text{norm}(v_{a,j})^2)} \sqrt{\sum_{j \in M_a \cap M_i} (\text{norm}(v_{i,j})^2)}} \end{aligned} \quad (2)$$

ここで M_a と M_i はそれぞれプロジェクト p_a と p_i で計測された(未欠損の)メトリクスの集合を表す。類似度 $\text{sim}(p_a, p_i)$ の値域は $[0, 1]$ である。この計算式では、未欠損の値のみを用いて類似度を計算するため、欠損値を補完する必要がない。なお、この類似度計算式は、他のアルゴリズム(相関係数を用いるものなど¹⁴⁾)にも交換可能である。

ステップ 3 (予測値計算): このステップでは前のス

テップで求めた類似度 $\text{sim}(p_a, p_i)$ を用いて、予測対象プロジェクト p_a のメトリクス m_b の予測値 $\hat{v}_{a,b}$ を計算する。予測値計算アルゴリズムは、従来の協調フィルタリングの研究において数多く提案されている^{4),9),14),16)}。

我々の予備実験において、従来の予測値計算アルゴリズムを工数予測に適用した結果、十分な予測精度が得られなかった。その原因は、ステップ 2 の類似度計算ではプロジェクトの規模が考慮されず(ベクトルのなす角や相関係数などを用い、ベクトルの大きさを用いないため)、規模の異なるプロジェクトどうしの類似度が高くなる場合があったためである。この問題は、ユーザの評価値を用いた従来の協調フィルタリングでは顕在化しなかったが、工数予測においては、プロジェクトの規模を何らかの方法で評価し、予測値計算に反映させる必要がある。

そこで、本論文では、予測値計算において、類似度 $\text{sim}(p_a, p_i)$ を重みとして、プロジェクト p_a と類似したプロジェクトのメトリクス値 $v_{i,b}$ に、プロジェクトの規模を補正する *amplifier* $\text{amp}(p_a, p_i)$ を乗じた値で加重平均を行う。予測値 $\hat{v}_{a,b}$ を次式により計算する:

$$\begin{aligned} \hat{v}_{a,b} &= \frac{\sum_{i \in k\text{-nearestProjects}} (v_{i,b} \times \text{amp}(p_a, p_i) \times \text{sim}(p_a, p_i))}{\sum_{i \in k\text{-nearestProjects}} \text{sim}(p_a, p_i)} \end{aligned} \quad (3)$$

ここで $k\text{-nearestProjects}$ は、プロジェクト p_a との類似度が高い k 個のプロジェクトの集合を表す(以下、類似プロジェクトと呼ぶ)。類似プロジェクトは、メトリクス m_b が測定されているもののみとする。一般に、この類似プロジェクト数 k は予測精度に影響を与える。

$h = |M_a \cap M_i|$ ($M_a \cap M_i$ の積集合の要素数)、 $r_j = \frac{\text{norm}(v_{a,j})}{\text{norm}(v_{i,j})}$ とおき、 $\text{amp}(p_a, p_i)$ を次式のように定義する:

$$\begin{aligned} \text{amp}(p_a, p_i) &= \begin{cases} r_n & h = \text{奇数} (2n - 1) \\ \frac{r_n + r_{n+1}}{2} & h = \text{偶数} (2n) \end{cases} \end{aligned} \quad (4)$$

amplifier は、プロジェクト p_a の規模が p_i の

規模のおよそ何倍になっているかを、正規化されたメトリクスの比 r_j の中央値により求めている。この式は、工数をはじめとする多くのソフトウェアメトリクスが、プロジェクトの規模と相関が高いことを利用している。この amplifier により、多様な規模のプロジェクトを含むデータセットを用いた場合にも工数予測が可能となる。なお、より単純な方式として、プロジェクトの規模を表すメトリクス（プロジェクト終了時のソースコードの行数など）を直接 amplifier の計算に用いることも考えられるが、工数予測を行う時点において予測対象のプロジェクトの規模が確定している（計測されている）とは限らない。そのため、本論文では、間接的に規模の比を求め一手段として、やや乱暴ではあるが、 r_j の中央値を用いることとした。

従来の協調フィルタリングでは amplifier を用いずに、類似度を重みとした加重平均で予測値を計算している¹⁶⁾。他の方法として、予測対象ユーザの評価値の平均値からの偏差を用いて予測値計算を行う方法^{4),14)}も提案されているが、これらを用いて予備実験を行った際の工数の予測精度は amplifier を用いた場合よりも低かったため、本論文では採用しなかった（予測値計算のステップで重回帰モデルを利用する方法¹⁶⁾も提案されているが、その工数予測への応用は今後の課題である）。

3. ケーススタディ

3.1 目的

ケーススタディの目的は、予測に用いるデータに欠損値が多数含まれている場合における、提案方法の予測精度を評価することである。予測精度の評価基準は絶対誤差と相対誤差とした。評価基準の比較対象として、欠損値処理法を用いたステップワイズ重回帰分析による予測を用いた。

3.2 欠損値処理法を用いたステップワイズ重回帰分析

重回帰分析はソフトウェア開発における工数予測に広く用いられている^{5),25)}。重回帰分析では、予測対象の変数（目的変数）と、目的変数に対して影響すると考えられる複数の変数（説明変数）の関係を一次式で表したモデルを作成し、予測を行う。ソフトウェア開発における工数予測の場合、試験工数などが目的変数であり、それ以外のメトリクスが説明変数の候補となる。

候補の中から説明変数を選択する方法として、ステッ

プワイズ変数選択法がある。ステップワイズ変数選択法は以下の手順で行う²¹⁾。(1) 初期モデルを作成する、(2) 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、有意水準5%で棄却されなかった場合に変数を選択し、有意水準10%で棄却される場合には変数を除去する。ただし、多重共線性を回避するために、選択する変数の VIF (variance inflation factor, 分散拡大要因) が10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 VIF が10以上となる場合、その変数は選択しない、(3) 手順(2)ができなくなるまで繰り返す。ステップワイズ変数選択法を用いた重回帰分析をステップワイズ重回帰分析と呼ぶ。

データに欠損値を含んだままでは、ステップワイズ重回帰分析を行うことはできない。欠損値を含むデータを用いる場合、欠損値処理法を用いてステップワイズ重回帰分析を行う。欠損値処理法として以下の3つが広く用いられている¹¹⁾。

リストワイズ除去法 欠損値を1つでも含んでいる

プロジェクトをデータから除去し、欠損値をまったく含まないプロジェクトのみでモデル作成を行う。

ペアワイズ除去法 モデル作成に必要な相関係数を求める際、ペアにした変数に欠損値が含まれていないプロジェクトは計算に用い、欠損値が含まれているプロジェクトは用いない。

平均値挿入法 変数から欠損値を除いて平均値を計算し、計算した平均値を欠損値部分に挿入する。変数に欠損値が含まれていない状態にしてモデル作成を行う。欠損値部分に値を挿入する方法としては、もっと単純かつ一般的な方法である¹²⁾。

3.3 利用したデータ

ケーススタディでは、試験工数の予測を行った。表1にケーススタディで使用したデータに含まれるメトリクスと、各メトリクスの欠損値含有率を示す。データは株式会社 NTT データにおいて約10年間にわたって記録されたものであり、プロジェクト数は1,081件であった。収集すべきメトリクスの種類が年度や部署によって異なっていたため、データには欠損値が多数含まれている。

データ全体に含まれる欠損値の割合は60%であるが、メトリクスごとに欠損値を含む割合が大きく異なっている。メトリクス $m_2 \sim m_4$ には欠損値がほとんど含まれていないが、 $m_1, m_5 \sim m_{13}$ には少なくとも70%の欠損値が含まれている。説明変数の候補として、試験工程までに得られる $m_1 \sim m_{13}$ を用いる。目的変数は m_{14} である。

表 1 ケーススタディに用いたデータに含まれるメトリクス
Table 1 Metrics in case study data.

メトリクス	欠損値含有率	
m_1	汎用システム（汎用システムならば 1，それ以外ならば 0 とする）	76%
m_2	新規開発（新規開発ならば 1，それ以外ならば 0 とする）	7%
m_3	設計工数	0%
m_4	製造工数	0%
m_5	設計工数（正社員分）	87%
m_6	設計工数（派遣社員分）	87%
m_7	設計工数（外注分）	87%
m_8	製造工数（正社員分）	87%
m_9	製造工数（派遣社員分）	87%
m_{10}	製造工数（外注分）	87%
m_{11}	基本設計レビューで発見された欠陥（件/頁・発見数を仕様書の頁数で除算）	84%
m_{12}	詳細設計レビューで発見された欠陥（件/頁）	71%
m_{13}	プログラム設計レビューで発見された欠陥（件/頁）	80%
m_{14}	試験工数	0%

3.4 評価基準

本論文では、予測方法の評価基準として、一般的に用いられている以下の 5 つを用いる^{7),10),17),19),24)}。以降、実測値（記録されている値）を Y と表し、予測値を \hat{Y} と表す。また、メトリクスに含まれている Y の個数を t とおく。

絶対誤差の平均 (MAE):

$$MAE = \frac{\sum |\hat{Y} - Y|}{t} \quad (5)$$

絶対誤差の分散 (VAE):

$$VAE = \frac{\sum (|\hat{Y} - Y| - MAE)^2}{t} \quad (6)$$

相対誤差の平均 (MRE):

$$MRE = \sum \left| \frac{\hat{Y} - Y}{Y} \right| \times \frac{1}{t} \quad (7)$$

相対誤差の分散 (VRE):

$$VRE = \sum \left(\left| \frac{\hat{Y} - Y}{Y} \right| - MRE \right)^2 \times \frac{1}{t} \quad (8)$$

Pred25: 相対誤差 25%以下の \hat{Y} が、 t に含まれている割合。一般的に Pred25 が大きいほど、予測精度が高いと考えられる。

$$\text{Pred25} = \frac{\sum \text{isAccurate} \left(\frac{\hat{Y} - Y}{Y} \right)}{t} \quad (9)$$

$$\text{isAccurate} \left(\frac{\hat{Y} - Y}{Y} \right) = \begin{cases} 1 & \left| \frac{\hat{Y} - Y}{Y} \right| \leq 0.25 \\ 0 & \left| \frac{\hat{Y} - Y}{Y} \right| > 0.25 \end{cases}$$

3.5 ケーススタディの手順

ケーススタディでは、以下の手順で m_{14} を予測した。

- 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した。フィットデータは予測モデルの構築に利用し、テストデータは予測モデルの評価に用いる。フィットデータとテストデータのペアを 10 組作成した。

- ② ① で作成した 10 組のペアを用いて、2.2 節で述べた提案方法の予測精度を評価した。適切な類似プロジェクト数（式 (3) の k の値）を発見するため、フィットデータを用いて類似プロジェクト数を変更 ($k = 1$ から $k = 50$) し、最も MRE が小さくなった k を採用した。そして、採用した類似プロジェクト数でテストデータを用いた予測を行い、MAE, VAE, MRE, VRE, Pred25 を計算した。
- ③ ① で作成した 10 組のペアを用いて、3.2 節で述べた欠損値処理法を用いたステップワイズ重回帰分析でモデルを作成した。そして、テストデータの欠損値に各変数の平均値を代入した後、フィットデータより作成したモデルを用いて予測を行い、MAE, VAE, MRE, VRE, Pred25 を計算した。なお、作成したどのモデルにおいても、条件指標 (condition index) の最大値は 15 以下となった。一般的に条件指標が 30 以上の場合、多重共線性があると判断される²⁰⁾。
- ④ ②, ③で得られた評価基準を比較し、どの方法が最も予測精度が高いのかを結論付けた。また、得られた結果に対して、その意味と限界について考察を行った。

4. ケーススタディの結果と考察

4.1 予測精度の比較

提案手法を用いて予測した際、類似プロジェクト数 k を 21 とした。類似プロジェクト数 ($k = 1 \sim 50$) と

表 2 評価基準 (10 個のテストデータから得られた評価基準の平均値)
Table 2 Averaged evaluation criteria resulted with the 10 test dataset.

	MAE	VAE	MRE	VRE	Pred25
リストワイズ除去法	0.91	5.50	192.61	3.987E+06	8%
(P 値)	(0.0059)	(0.11)	(0.0020)	(0.0020)	(0.0020)
ペアワイズ除去法	1.76	181.20	223.95	3.566E+06	7%
(P 値)	(0.0020)	(0.0059)	(0.0020)	(0.0020)	(0.0020)
平均値挿入法	0.32	3.87	22.11	2.571E+04	12%
(P 値)	(0.0020)	(0.049)	(0.0020)	(0.0020)	(0.0020)
協調フィルタリング	0.19	1.31	0.79	2.82	37%

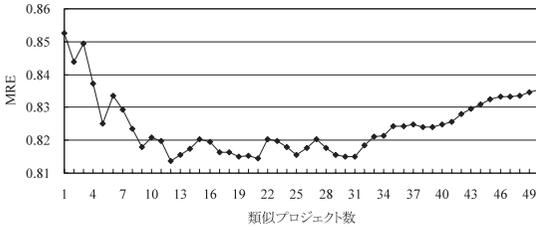


図 2 類似プロジェクト数と MRE の関係 (10 個のフィットデータから得られた MRE の平均値。k = 1 ~ 50)
Fig.2 The relation between neighborhood size and MRE (Averaged MRE resulted with the 10 fit dataset, from k = 1 up to k = 50).

MRE との関係を図 2 に示す。グラフにおいて、横軸は類似プロジェクト数を、縦軸は k に対応する MRE を示している。縦軸の MRE は、10 個のフィットデータから得られた MRE を平均したものである。MRE が最小になっている k をテストデータの予測に採用した。

ケーススタディに用いた予測方法を比較した結果、提案方法が最も高い予測精度となった。表 2 にケーススタディの結果から得られた評価基準を示す。表は 10 個のテストデータから得られた評価基準を平均したものである。MAE, VAE, MRE, VRE は値が小さいほど予測精度が高く、逆に Pred25 は値が大きいほど予測精度が高いことを示す。表を見ると、提案方法の MAE, VAE, MRE, VRE の値が最も小さく、Pred25 が最も大きいことが分かる。なお、欠損値処理法どうして評価基準を比較すると、平均値挿入法の予測精度が最も高く、逆にペアワイズ除去法の予測精度は最も低かった。

さらに、ステップワイズ重回帰分析と提案方法の評価基準に差があるかどうかを検定すると、VAE を除く評価基準で有意な差があった。検定の結果、有意差があったものを表 2 に斜体で示す。ケーススタディでは、同じフィットデータに対して提案方法とステップワイズ重回帰分析を適用しているが、各評価基準は正規分布していなかったため、ノンパラメトリックな検定である Wilcoxon の符号付順位和検定を行った²²⁾。

検定は提案方法と 3 つの欠損値処理法をそれぞれ対して行うため、1 つの評価基準について 3 回検定を行うことになる。多重比較を行うため、Bonferroni の方法²²⁾ により有意水準の調整を行い、 $0.05/3=0.0166$ を有意水準に設定した。

4.2 考 察

ケーススタディでは、欠損値処理法を用いたステップワイズ重回帰分析よりも、提案方法のほうが高い予測精度を示した。ステップワイズ重回帰分析の予測精度が低くなっているのは、欠損値の存在が予測モデルの性能を低下させているためであると考えられる。性能の低下した 1 個の予測式を使って、あらゆるプロジェクトを予測しなければならぬため、予測精度が低くなっていると考えられる。協調フィルタリングの場合、プロジェクトごとに予測式を作成するので、欠損値が少ないケースでは高い予測精度が得られ、欠損値が多いケースでもそれなりの精度が得られているものと考えられる。

本ケーススタディの限界は、1 つのデータセットを用いた評価しか行っていないことである。そのため、任意のデータセットに対して提案方法が有効であるかどうかは、現時点では定かでない。現実には、ソフトウェア開発企業で記録されるデータセットの状態は様々である(欠損値の量と分布が異なっている)。本論文の結論を強化し、信頼性を向上させるために、他のデータセットを用いたり、欠損値の量と分布を実験的に変化させたりしたケーススタディを行うことが課題となる。

5. 関連研究

ケーススタディで用いた欠損値処理法、特にリストワイズ除去法は多くの統計ソフトで欠損値処理法として初期設定されており¹²⁾、ソフトウェア工学の研究においても広く用いられている^{5),12),19)}。しかし、リストワイズ除去法は、欠損値が含まれる割合が高くなると、利用可能なデータを著しく減らしてしまう。また、欠損値を補完する方法(平均値挿入法など)は、デー

タの特性(分散など)が変わり、望ましいモデルが得られない場合がある。どの欠損値処理法を用いても問題がないのは、欠損値の量が少なく、かつ欠損値の分布がランダムである場合のみである¹⁹⁾。

本論文では、提案方法との比較を、リストワイズ除去法、ペアワイズ除去法、平均値挿入法の3つの欠損値処理法で行ったが、これら以外にも、あまりメジャーな方法ではないものの、欠損値処理法は存在する。たとえば、Strikeら¹⁹⁾はリストワイズ除去法、平均値挿入法、ホットデック挿入法の性能を比較した結果、ホットデック挿入法の性能が最も高いことを示している。Myrtveitら¹²⁾は、リストワイズ除去法、平均値挿入法、類似レスポンスパターン挿入法(SRPI)、完全情報最尤法(FIML)の性能を比較した結果、欠損値の分布がランダムでない場合は、FIMLを用いるのが適切であり、それ以外の欠損値処理法では適切なモデルが構築できないと結論付けている。

ただし、これらの従来研究では、本論文で用いたような欠損値が60%以上含まれるデータを用いた評価は行っていない。今後、提案方法とホットデック挿入法、SRPI、FIMLなどとの比較実験を行うことが課題としてあげられる。

協調フィタリングによく似た方法として、analogy(類似度)-based法が提案されている^{2),17),24)}。Analogy-based法は、提案方法と同様、予測対象のプロジェクトと類似したプロジェクトを選び出し、類似するプロジェクトの工数を元に、予測対象のプロジェクトの工数を予測する。

しかし、analogy-based法は、重回帰分析と同様、欠損値を含むデータセットに直接適用できず、欠損値処理法を用いて欠損値のないデータセットをあらかじめ作成することが求められている¹²⁾。その理由の1つは、類似度の計算方法にあると考えられる。代表的なanalogy-based法であるShepperdら¹⁷⁾の方法では、ベクトルのユークリッド距離を用いて類似度を計算し、類似プロジェクト($k = 1 \sim 3$)の工数の(類似度による)加重平均により予測工数を計算する。ここで、仮に欠損値を含んだまま2つのベクトル間のユークリッド距離を計算したとすれば、欠損値の多寡(すなわちベクトルの次元)に類似度の値が大きく影響されることとなり、十分な精度が得られない可能性がある。一方、本論文では、類似度の計算にベクトルのなす角を用いているため、ベクトルの次元にかかわらず類似度が $[0, 1]$ の値域に収まる点が異なる。なお、Sheppaerdらの方法では、予測値計算において、amplifierのような補正を行わないため、予測工数は、過去のプロジェ

クトで記録された工数を超える値をとることができず、規模の大きなプロジェクトの工数を過少予測する傾向がある。

欠損値の含むデータセットに適用できそうな類似度計算を採用したanalogy-based法も存在する。Walkerdenら²⁴⁾は、2プロジェクト間の各メトリクス値の差からメトリクスごとの類似度を求め、その平均値をプロジェクトの類似度としている。類似度の値域がメトリクスの個数に依存しないため、欠損値を含むデータセットに適用できるように類似度計算式を拡張することは可能と考えられる。一方、予測値計算では、類似プロジェクトの工数 ÷ 類似プロジェクトのFP(ファンクションポイント) × 予測対象プロジェクトのFP、により工数を求めている。この方法は、プロジェクトの規模(の指標の1つであるFP)により予測値の補正を行うという点において、提案方法のamplifierと似ている。ただし、提案方法と異なり、すべてのプロジェクトにおいてFPが計測されている(欠損値が存在しない)ことが求められる。

これらのanalogy-based法に用いられている類似度計算、および、予測値計算の一部は、欠損値を含むデータセットに適用できるように拡張できる可能性がある。拡張した計算式を提案方法のステップ2、および、3に適用し、その性能を評価することは、今後の課題の1つである。

6. おわりに

本論文では、過去のソフトウェア開発プロジェクトにおいて記録された多種類のソフトウェアメトリクス値を入力データとし、協調フィルタリングにより予測工数を求める方法を提案した。提案方法は、欠損値を数多く含む入力データに用いた場合においても、欠損値の除外や補完を行うことなしに工数予測が行える。60%の欠損値を含むデータを用いて予測を行った結果、提案方法は従来のステップワイズ重回帰分析よりも高い予測精度を示した。なお重回帰分析には、欠損値の割合が増加するに従い、精度が低下する傾向がある¹⁹⁾。提案手法にも同様の傾向が存在する可能性があり、欠損値の割合と予測精度の関係を明らかにすることは今後の課題である。

これまでソフトウェア工学の研究において、データ収集の際に起こる欠損値の問題は見落とされがちであった⁶⁾。欠損のないデータセットを用いることを前提とした予測モデルは非常に数多く提案されているのに対し、欠損値処理法に関する提案、評価を行った論文は数少ない。我々は産学連携研究を通じて、データ

にしばしば含まれる欠損値に着目した。そして、欠損値が存在することを前提とした工数予測方法を提案し、企業の実データを用いてその評価を行った。本論文の成果は、産業界の現状に即した問題に対して、産学が連携して解決策を得たという点において、産学連携研究の成功事例の1つであると考えられる。

提案方法は、改良の余地があると考えている。1つは、類似度計算、および、予測値計算の方法を改良することである。類似度計算においては、各メトリクスに対して何らかの重み付けを行うことが考えられる。現状では、すべてのメトリクスを $[0, 1]$ に正規化するため、同一の重み付けとなっている。予測値計算においては、式 (3) の amp を変更することで、予測精度が改善する可能性がある。今後、回帰モデルを利用する方法¹⁶⁾の適用について検討する予定である。もう1つは、類似度を計算する際に、変数選択を行うことである。今回のケーススタディでは、重回帰分析ではステップワイズ法による変数選択を行っているが、提案方法では、変数選択を行わずすべてのメトリクスを用いている。予測精度が最も高くなるようにメトリクスを選択できれば、精度が改善される可能性がある。

謝辞 本研究の一部は、文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託に基づいて行われた。

参 考 文 献

- 1) Albrecht, A. and Gaffney, J.: Software Function, Source Lines of Code and Development Effort Prediction, *IEEE Trans. Softw. Eng.*, Vol.9, No.6, pp.83–92 (1979).
- 2) Auer, M. and Biffi, S.: Increasing the Accuracy and Reliability of Analogy-Based Cost Estimation with Extensive Project Feature Dimension Weighting, *Proc. International Symposium on Empirical Software Engineering (ISESE'04)*, California, pp.147–155 (2004).
- 3) Boehm, B.W.: *Software Engineering Economics*, p.767, Prentice-Hall, Englewood Cliff, New Jersey (1981).
- 4) Breese, J.S., Heckerman, D. and Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, Wisconsin, pp.43–52 (1998).
- 5) Briand, L., Langley, T. and Wiczorrek, I.: A replicated assessment and comparison of common software cost modeling techniques, *Proc. 22nd IEEE International Conf. on Software Eng.*, Limerick, pp.377–386 (2000).
- 6) Cartwright, M.H., Shepperd, M.J. and Song, Q.: Dealing with Missing Software Project Data, *Proc. 9th International Software Symposium*, Sydney, pp.154–165 (2003).
- 7) Conte, S.D., Dunsmore, H.E. and Shen, V.Y.: *Software Engineering Metrics and Models*, p.396, The Benjamin/Cummings Publishing Company, Inc., Menlo Park, California (1986).
- 8) Finnie, G.R., Wittig, G.E. and Desharnais, J.M.: A comparison of software effort estimation techniques: Using function points with neural networks, casebased reasoning and regression models, *Journal of Systems and Software*, Vol.39, No.3, pp.281–289 (2000).
- 9) Goldberg, D., Nichols, D., Oki, B.M. and Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry, *Comm. ACM*, Vol.35, No.12, pp.61–70 (1992).
- 10) Khoshgoftaar, T.M., Munson, J.C., Bhattacharya, B.B., and Richardson, G.D.: Predictive Modeling Techniques of Software Quality from Software Measures, *IEEE Trans. Softw. Eng.*, Vol.18, No.1, pp.979–987 (1992).
- 11) Little, R. and Rubin, D.: *Statistical Analysis with Missing Data*, 2nd ed., p.408, John Wiley & Sons, New York (2002).
- 12) Myrtveit, I., Stensrud, E. and Olsson, U.H.: Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods, *IEEE Trans. Softw. Eng.*, Vol.27, No.11, pp.999–1013 (2001).
- 13) Ohsugi, N., Tsunoda, M., Monden, A. and Matsumoto, K.: Effort Estimation Based on Collaborative Filtering, *Proc. 5th International Conference on Product Focused Software Process Improvement (Profes2004)*, Kyoto, pp.274–286 (2004).
- 14) Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proc. ACM Conf. on Computer Supported Cooperative Work (CSCW'94)*, North Carolina, pp.175–186 (1994).
- 15) Salton, G. and McGill, M.: *Introduction to Modern Information Retrieval*, p.448, McGraw-Hill, New York (1983).
- 16) Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms, *Proc. 10th International World Wide Web Conference (WWW10)*, Hong Kong, pp.285–295 (2001).
- 17) Shepperd, M. and Schofield, C.: Estimating Software Project Effort Using Analogies, *IEEE Trans. Softw. Eng.*, Vol.23, No.12, pp.736–743

(1997).

- 18) Srinivasan, K. and Fisher, D.: Machine Learning Approaches to Estimating Software Development Effort, *IEEE Trans. Softw. Eng.*, Vol.21, No.2, pp.126–137 (1995).
- 19) Strike, K., El Eman, K. and Madhavji, N.: Software Cost Estimation with Incomplete Data, *IEEE Trans. Softw. Eng.*, Vol.27, No.10, pp.890–908 (2001).
- 20) Tabachnick, B.G. and Fidell, L.S.: *Using Multivariate Statistics*, 3rd Edition, p.880, Harper Collins College Publishers, New York (1996).
- 21) 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).
- 22) 田中 豊, 垂水共之 (編): Windows 版 統計解析ハンドブック ノンパラメトリック法, p.164, 共立出版, 東京 (1999).
- 23) 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ソフトウェア信頼性予測方法, 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, No.SS2003-27, pp.19–24 (2003).
- 24) Walkerden, F. and Jeffery, R.: An Empirical Study of Analogy-based Software Effort Estimation, *Empirical Software Engineering*, Vol.4, Issue 4, pp.135–158 (1999).
- 25) Walston, C. and Felix, C.: A Method of Programming Measurement and Estimation, *IBM Systems Journal*, Vol.1, pp.54–73 (1977).

(平成 16 年 9 月 1 日受付)

(平成 17 年 2 月 1 日採録)



角田 雅照

平成 9 年和歌山大学経済学部卒業。平成 16 年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修了。現在, 同大学院情報科学研究科博士後期課程在籍。ソフトウェアメトリクス, ソフトウェアプロセスに興味を持つ。電子情報通信学会, IEEE 各会員。



大杉 直樹 (学生会員)

平成 13 年奈良工業高等専門学校専攻科電子情報工学専攻修了。平成 16 年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修了。同年同大学産学官連携研究員。博士 (工学)。エンピリカルソフトウェア工学, 特に, ソフトウェア開発コスト/信頼性/リスク予測, ソフトウェアプロセス改善, 協調フィルタリングの研究に従事。電子情報通信学会, IEEE 各会員。



門田 暁人 (正会員)

平成 6 年名古屋大学工学部電気学科卒業。平成 10 年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修了。同年同大学同研究科助手。平成 16 年同大学助教授。平成 15~16 年 Auckland 大学客員研究員。博士 (工学)。ソフトウェアメトリクス, ソフトウェアセキュリティ, ヒューマンインタフェースの研究に従事。電子情報通信学会, 日本ソフトウェア科学会, 教育システム情報学会, IEEE, ACM 各会員。



松本 健一 (正会員)

昭和 60 年大阪大学基礎工学部情報工学科卒業。平成元年同大学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。平成 5 年奈良先端科学技術大学院大学助教授。平成 13 年同大学教授。工学博士。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。電子情報通信学会, ACM 各会員, IEEE Senior Member。



佐藤 慎一 (正会員)

平成 6 年大阪大学基礎工学部情報工学科卒業。平成 8 年同大学大学院博士前期課程修了。同年株式会社 NTT データ入社。ソフトウェアプロセス改善, エンピリカルソフトウェア工学, ソフトウェアメトリクスに興味を持つ。