# 工数予測における類似性に基づく欠損値補完の効果

**Effect of Similarity-Based Missing Data Imputation on Effort Estimation**

**田村 晃一**[*] **柿元 健**[†] **戸田 航史**[‡] **角田 雅照**[§] **門田 暁人**[¶]
**松本 健一**[‖] **大杉 直樹**[**]

**あらまし**
開発中，もしくは将来のプロジェクトの計画，管理を目的として開発工数の予測が行われている．工数予測では重回帰モデルが広く用いられている．重回帰モデルは過去のソフトウェア開発プロジェクトで計測・収集されたソフトウェアメトリクス値に基づいて構築される．構築に用いる過去のデータセットに未記録の値（欠損値）が存在している場合，モデル構築を行う前に，欠損値を含むプロジェクトを除外したり，欠損値を当該メトリクスの平均値で補完するなどの欠損値処理が行われる．ただし，データセットに含まれる欠損値の割合が高い場合，従来の欠損値処理では性能のよいモデルの構築は期待できない．本稿では，欠損値が多い場合にもそれなりの予測性能をもったモデルを構築するために，欠損値を単に平均値で埋めるのではなく，プロジェクト間の類似性に基づいて欠損値を推定し，補完を行う．プロジェクト間の類似性に基づいて欠損値を推定する方法は，協調フィルタリングを応用したアルゴリズムを用いた．類似性に基づく欠損値補完の効果を評価するために，工数を予測する実験を行い，従来の欠損値処理法と予測精度を比較した．その結果，類似性に基づく欠損値補完を行った場合，相対誤差平均が 0.415 から 0.220 に改善した．

## 1 はじめに

ソフトウェア開発プロジェクトにおける工数予測は，プロジェクト完遂に必要な資源，及びスケジュール管理を行う上で重要である．必要な工数を過不足なく予測することで，納期遅れ，コスト超過といったプロジェクトの失敗を防ぐことができる．そのため，工数予測に関する数多くの研究が行われている [1] [7] [8] [11]．

工数予測モデルを作成するために，過去のプロジェクトで収集されたデータを用いる必要があるが，過去のプロジェクトのデータには欠損値が数多く含まれている．一方で，重回帰分析などの工数予測手法は，モデルを構築するためのデータセットに欠損値が含まれていないことが前提となっている．そこで，モデル構築に先立って，欠損値を含むプロジェクトを除外したり，欠損値を当該メトリクスの平均値で補完することが行われる．しかし，データセットに含まれる欠損値の割合が 30 % を越えることもしばしばあり，そのような場合には，たとえ欠損値を補完したとしても，性能のよいモデルの構築は期待できない [5]．

本稿では，データセットに含まれる欠損値を補完する方法として，プロジェクト間の類似性に基づいて欠損値を推定し，補完を行う手法を提案する．プロジェクト間の類似性に基づいて欠損値を推定する方法は，協調フィルタリングを応用したア

[*]Koichi Tamura，奈良先端科学技術大学院大学
[†]Takeshi Kakimoto，奈良先端科学技術大学院大学
[‡]Koji Toda，奈良先端科学技術大学院大学
[§]Masateru Tunoda，奈良先端科学技術大学院大学
[¶]Akito Monden，奈良先端科学技術大学院大学
[‖]Ken-ichi Matsumoto，奈良先端科学技術大学院大学
[**]Naoki Ohsugi，株式会社 NTT データ

ルゴリズム [11] を用いた．協調フィルタリングの特徴として，欠損値が多いデータセットを入力とした場合でも予測が行える特徴があり，情報検索の分野においてさかんに研究が行われてきた．協調フィルタリングを過去に行われたソフトウェア開発プロジェクトの欠損値補完に利用することで，より適切な欠損値補完が行えるようになり，高い精度で工数予測が可能になると期待される．

本稿では，提案手法の有効性を実データを用いて実験的に評価し明らかにする．実験では，実データの欠損値を提案手法によって補完し，補完したデータセットを用いてプロジェクトの総工数を予測する．また，比較対象として，従来の欠損値処理手法である，ペアワイズ除去法，平均値挿入法を用いた予測も行う．それぞれの欠損値処理を行ってから予測モデルを構築した際の予測精度を比較することにより評価を行う．なお，本稿ではもともと欠損の含まれるデータセットを用いるために，欠損値の真の値は不明である．そのため，補完値そのものの正確さの評価（真の値との比較）は行っていない．

本稿では，筆者らの予備実験 [10] で用いたデータセットを見直し，より妥当性の高い実験を行った．予備実験では，説明変数として有力な変数（要件定義工数など）に欠損が少なく，説明変数として有力でない変数（FP 計測時の外部参照数など）に欠損が多いデータセットを用いてテスト工数を予測していた．また，開発期間などのよく用いられる変数が説明変数に含まれていなかった．そのため，欠損値補完の効果を（工数予測により）評価するためのデータセットとして適切とはいえず，類似性に基づく欠損値補完の効果は小さいという結果が得られている．本稿で用いたデータセットでは，説明変数として従来よく用いられている変数を用い，かつ，それらがある程度の欠損を含んでいる．(ただし，データセットの選定が恣意的にならないように十分配慮した．)

以降，2 章では，本稿の実験で用いた工数予測手法であるステップワイズ重回帰分析について述べ，3 章では，提案手法であるプロジェクト間の類似性に基づく欠損値補完，および，従来の欠損値処理について述べる．4 章では，提案手法の有効性を示すための評価実験について説明し，5 章で評価実験の結果について述べる．6 章で関連研究について述べ，最後に 7 章で本稿の結論について述べる．

## 2　ステップワイズ重回帰分析による工数予測

重回帰分析は多変量解析の一手法であり，ソフトウェア開発に要する工数を予測するために広く用いられている．本稿では，工数予測手法として重回帰分析の一手法であるステップワイズ重回帰分析を用いた．

重回帰分析では，予測対象の変数 (目的変数) と，目的変数に影響を与える複数の変数 (説明変数) との関係を表した一次式 (回帰式) を作成する．回帰式中の各係数と定数は，予測値の絶対誤差（残差）の 2 乗和が最小になるように決定される．作成された回帰式に，現行プロジェクトで計測した説明変数を与えることで，目的変数を予測することが可能となる．

重回帰分析では，予測精度を向上させるために，多数の説明変数候補の中から，予測精度の向上に寄与すると予測される変数を選択して回帰式を作成する方法がとられる．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採用する変数を決定し，重回帰分析を行う手法である．ステップワイズ変数選択は次の手順で行われる．

**手順 1**. 変数を全く含まないモデルを初期モデルとして作成する．

**手順 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，指定した有意水準 (本稿の評価実験では，偏 F 値の有意水準を $p_{in} = 0.05$, $p_{out} = 0.1$ とした) で棄却されない場合に変数を採択する．ただし，多重共線性を回避するために，採択する変数の分散拡大要因 (VIF) が一定値以上の場合，またはその変数を採択することによって，他の変数の VIF が一定値以上となる

場合，その変数は採択しない．

**手順 3**．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　メトリクスの欠損値の補完

### 3.1　提案手法

本稿では，プロジェクト間の類似性に基づく予測手法を，データセットに含まれる欠損値の補完に適用することを提案する．プロジェクト間の類似性に基づく予測手法は，メトリクス値が類似した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を予測する．プロジェクト間の類似性に基づく予測手法は，欠損値が多数含まれるデータセットを用いても，高い精度で工数を予測できることが報告されている．欠損値の補完においても，メトリクス値が類似したプロジェクトの値を用いることで，より適切な値の補完が可能であると考えられる．

プロジェクト間の類似性に基づく予測手法は 3 つの手順（正規化，類似度計算，補完値計算）から構成される [11]．各手順の詳細とアルゴリズムについて以下で述べる．

**手順 1．メトリクス値の正規化**各メトリクスは値域に大きなばらつきがあるため，値域をそろえるための正規化を行い，値域を [0,1] にする．ここで，$p_i$ は $i$ 番目のプロジェクト，$m_j$ は $j$ 番目のメトリクスと定義すると，プロジェクト $p_i$ のメトリクス $m_j$ の値 $v_{ij}$ を正規化した値 $v'_{ij}$ は式 (1) で定義される．

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

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，$\max(P_j)$ 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．

**手順 2．プロジェクト間の類似度計算** メトリクス値を補完するプロジェクトと類似した他の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算出する．メトリクス値を補完するプロジェクト $p_a$ と他の各プロジェクト $p_i$ との類似度 $sim(p_a, p_i)$ は式 (2) で定義される．

$$sim_{p_a,p_i} = \frac{\sum_{j \in M_a \cap M_i} \left(v'_{aj} - md(m'_j)\right)\left(v'_{ij} - md(m'_j)\right)}{\sqrt{\sum_{j \in M_a \cap M_i} \left(v'_{aj} - md(m'_j)\right)^2} \sqrt{\sum_{j \in M_a \cap M_i} \left(v'_{ij} - md(m'_j)\right)^2}} \tag{2}$$

ここで，$M_a$ と $M_i$ はそれぞれプロジェクト $p_a$ と $p_i$ に関して記録されている（欠損していない）メトリクスの集合を表し，$md(m'_j)$ は $j$ 番目のメトリクスの中央値を表す．

$v_{ij}$ から $md(m'_j)$ を減算することで，中央値よりも大きなメトリクス値は正の値をとり，小さい値は負の値をとるようになる．類似度の計算例を図 1 に示す．図 1 の左側の図はメトリクス値から中央値を減算して類似度を計算した場合，右側の図はメトリクス値をそのまま使って類似度を計算した場合を示している．この計算により，類似度 $sim(p_a, p_i)$ の値域が [-1,1] をとるようになり，大きくメトリクス値が離れたプロジェクト間の類似度が小さくなる．

**手順 3．類似度に基づく補完値の算出** 補完対象となる欠損値について，その補完値の算出に類似したプロジェクトの対応するメトリクスの実測値を用いる．手順 2. の類似度計算では，（ベクトルのなす角を用い，ベクトルの大きさを用いないため）規模が異なるが傾向が似ているプロジェクト同士は類似度が高いとみなしている．そこで，補完値の算出において，類似度 $sim(p_a, p_i)$ 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 $v_{ib}$ に，プロジェクトの規模を補正する $amp(p_a, p_i)$ を乗じた値で加重平均を行う．プロジェクト $p_a$

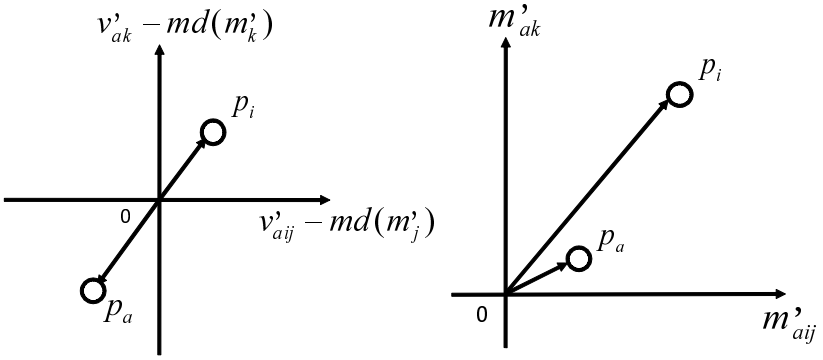


**図 1　プロジェクト間の類似度計算例**

のメトリクス $m_b$ の補完値 $\hat{v}_{ab}$ は式 (3) で定義される．

$$\hat{v}_{ab}=\frac{\sum_{i\in k\text{ - }nearestProjects}(v_{ib}\times amp(p_a,p_i)\times sim(p_a,p_i))}{\sum_{i\in k\text{ - }nearestProjects} sim(p_a,p_i)} \tag{3}$$

ここで，$k$-$nearestProjects$ は，メトリクス $m_b$ が欠損しておらず，かつ，プロジェクト $p_a$ と類似度の高い上位 $k$ 個のプロジェクトの集合を表す．$k$ の値は実験的に別途求める必要がある．

また，$amp(p_a,p_i)$ は式 (4) で定義される．

$$amp(p_a,p_i)=\begin{cases} r_n \quad \cdots \quad h=\text{奇数} \\ (r_1\le r_2\le\cdots\le r_n\le\cdots\le r_{2n-1}) \\ \dfrac{r_n+r_{n+1}}{2} \quad \cdots \quad h=\text{偶数} \\ (r_1\le r_2\le\cdots\le r_n\le\cdots\le r_{2n}) \end{cases} fd \tag{4}$$

ここで，$h=|M_a\cap M_i|$，$r_i=\frac{v'_{aj}}{v'_{ij}}$ である．

$amp$ 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．これは，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 $amp$ 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補完が可能となる．

### 3.2　欠損値処理

欠損値を含むデータセットに対してステップワイズ重回帰分析を適用する手法として，欠損値処理が従来使われている．欠損値処理とは多変量解析を可能とするために，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する，もしくは欠損値を何らかの値で補完する，といった前処理を行う方法のことである．重回帰分析に対しては，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種類の手法が広く用いられる [5] [9]．

**リストワイズ除去法**　欠損値を１つでも含むプロジェクトを全て除去する．

**ペアワイズ除去法**　重回帰分析に特化した手法で，重回帰分析の過程においてメトリクス間の相関を求める際に，相関を求めるメトリクスのいずれかが欠損しているプロジェクトを除外して相関を求める手法である [9]．

**表 1　実験に用いたデータセットに含まれるメトリクス**

| 名称 | 変数の内容 | 欠損率 (%) |
|---|---|---|
| Functional Size | FP 数 | 0 |
| Project Elapsed Time | プロジェクトの全開発期間 (単位:月) | 8.8 |
| Effort Plan | 計画工数 | 77.0 |
| Effort Specify | 要件定義工数 | 71.3 |
| Effort Build | コーディング工数 | 66.0 |
| Lines of Code | コードの行数 | 92.0 |
| Summary Work Effort | プロジェクトの総工数 (単位:時間) | 0 |

**平均値挿入法**　欠損値に対して，当該メトリクスの平均値を挿入することで，欠損値を補完する．

## 4　評価実験

### 4.1　実験用データセット

実験で利用したデータセットは，ISBSG(International Software Benchmarking Standards Group) が収集した，20ヶ国のソフトウェア開発企業の実績データ [3] である．データセットに含まれるメトリクスのうち Summary Work Effort を目的変数とし，6 個のメトリクスを説明変数として用いた．工数予測は開発工程の早い段階で行うべきであるが，実験で用いたデータセットでは予測時期を開発の初期段階とすると説明変数が少なくなるため，コーディング終了時を予測時期として想定した．総工数の予測値とコーディング終了時における工数の実測値との差を取ることにより，後行程（試験工程）の工数を知ることができる．7 種類のメトリクスの詳細について表 1 に示す．開発期間はプロジェクトの開始時点で決定していると想定し，プロジェクトの全開発期間を説明変数に含めた．この ISBSG データセットから，FP 計測手法が IFPUG であり，開発形態が新規開発，かつ Summary Work Effort（目的変数）が欠損していない 849 件（欠損率 45%）のプロジェクトを実験に用いた．

### 4.2　評価基準

予測精度の評価基準として一般的に用いられている，絶対誤差，相対誤差それぞれの平均値，中央値，及び Pred(25) の 5 種類の評価基準を用いて評価した．Pred(25) は予測値の相対誤差が 25%以下となったプロジェクトの割合を示す．Pred(25) は値が大きいほど予測精度が高いことを表わし，その他の評価基準は値が小さいほど予測精度が高いことを表す．

それぞれの評価基準は次の式 (5)〜(9) により計算される．ここで，M 件のプロジェクトがあるとする．また，実測値と予測値をそれぞれ $X_i$，$\hat{X}_i (i = 1 \sim M)$ とし，$A_i = \left| \hat{X}_i - X_i \right|$, $R_i = \frac{\left| \hat{X}_i - X_i \right|}{X_i}$ とおく．

**絶対誤差平均値 (MAE)**

$$MAE = \frac{\sum_{i=1}^{M} A_i}{M} \tag{5}$$

**絶対誤差中央値 (MdMAE)**

$$MdMAE = \begin{cases} \underset{(A_1 \leq A_2 \leq \cdots \leq A_n \leq \cdots \leq A_{2n-1})}{A_n} & M = \text{奇数} \\ \underset{(A_1 \leq \cdots \leq A_n \leq A_{n+1} \leq \cdots \leq A_{2n})}{\frac{A_n + A_{n+1}}{2}} & M = \text{偶数} \end{cases} \tag{6}$$

**相対誤差平均値 (MMRE)**

$$MMRE = \frac{\sum_{i=1}^{M} R_i}{M} \tag{7}$$

**相対誤差中央値 (MdMRE)**

$$MdMRE = \begin{cases} \underset{(R_1 \leq R_2 \leq \cdots \leq R_n \leq \cdots \leq R_{2n-1})}{R_n} & M = \text{奇数} \\ \underset{(R_1 \leq \cdots \leq R_n \leq R_{n+1} \leq \cdots \leq R_{2n})}{\frac{R_n + R_{n+1}}{2}} & M = \text{偶数} \end{cases} \tag{8}$$

**Pred(25)**

$$Pred(25) = \frac{\sum_{i=1}^{M} isAccurate(R_i)}{M} \tag{9}$$

$$isAccurate(R) = \begin{cases} 1 & R \leq 0.25 \\ 0 & R > 0.25 \end{cases}$$

### 4.3 実験手順

評価実験は次の手順で行った.

1. 4.1 で述べたデータセットを，欠損値を含むプロジェクトのみのデータセット (プロジェクト数 813 件（欠損率 47%）) と欠損値を含まないプロジェクトのみのデータセット (プロジェクト数 36 件) の 2 つのデータセットに分割した. 前者を予測モデルを作成するデータセット（以降ラーニングデータと呼ぶ）とし，後者をラーニングデータを用いて実際に予測を行うデータセット（以降テストデータと呼ぶ）とした. 欠損値を含まないデータセットをテストデータとしたのは，欠損値によって予測精度が影響されることを防ぐためである.
2. ラーニングデータに対し，提案手法，及び，比較対象であるペアワイズ除去法，平均値挿入法によって欠損値処理を行った. 本稿の評価実験では，欠損値処理を施すデータセット（ラーニングデータ）が欠損値を含むプロジェクトのみで構成されており，リストワイズ除去法では全てのプロジェクトが除去されるため，ペアワイズ除去法および平均値挿入法を用いた. 提案手法における式 (3) の *k-nearestProjects* は，予備実験において MRE が最小となった 16 を用いた.
3. それぞれの手法で欠損値を補完あるいは削除したテストデータに対して，ステップワイズ重回帰分析で予測を行い，各評価基準を算出した. 予測対象のメトリクスを Summary Work Effort とし，テストデータの Summary Work Effort は未知数として予測を行った. ステップワイズ変数選択における分散拡大要因 (VIF) は 10 とした.

**表 2　ステップワイズ重回帰分析に各欠損値処理を適用した際の予測精度**

| | MAE | MdMAE | MMRE | MdMRE | Pred(25) |
|---|---|---|---|---|---|
| 提案手法 | 4071 | 1213 | 0.220 | 0.185 | 64% |
| ペアワイズ除去法 | 5854 | 2250 | 0.415 | 0.378 | 33% |
| 平均値挿入法 | 5044 | 1538 | 0.492 | 0.279 | 36% |

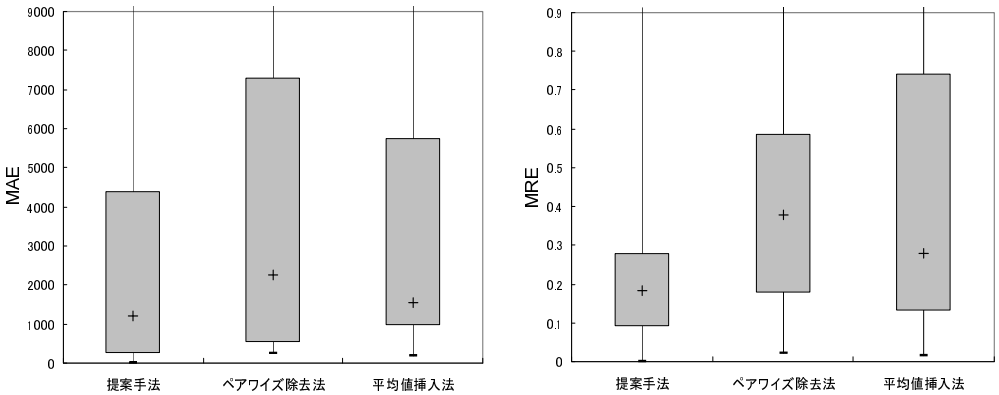


**図 2　各手法ごとの予測精度の箱ひげ図（拡大）**

## 5　実験結果

提案手法及び従来法によって欠損値処理を行い，ステップワイズ重回帰分析で予測した時の各評価基準の値を表 2 に示す．

表 2 より，提案手法を用いて欠損値を補完した場合，評価基準の 4 種類の誤差の値が最も小さく，Pred(25) の値が最も大きい．このことから，提案手法を用いて欠損値を補完した場合が最も高い精度で予測可能であると言える．また，ペアワイズ除去法と平均値挿入法を比較すると，絶対誤差平均値，絶対誤差中央値及び相対誤差中央値では平均値挿入法が，相対誤差平均値ではペアワイズ除去法の方が精度が高い．

各手法の絶対誤差，相対誤差の箱ひげ図を拡大したものを図 2 に示す．グラフの縦軸は予測誤差を示し，箱の下端は第 1 四分位，上端は第 3 四分位，箱中の+は中央値，線分（ひげ）の下端の横線は最小値を表す．グラフの上方を省略しているため最大値は示されていない．

図 2 の箱ひげ図より，提案手法を用いて欠損値を補完した場合，箱が一番下に来ており，全体的に精度が高いと言える．また，最小値，第 1 四分位，第 3 四分位において，比較対象の手法よりも値が小さくなっていることがわかる．さらに，提案手法の方が他の手法よりも箱が小さい，すなわち誤差のばらつきが小さくなっていることもわかる．

各手法の絶対誤差，相対誤差の箱ひげ図の全体を図 3 に示す．グラフの上端の横線は最大値を表す．図 3 の箱ひげ図により，誤差の最大値を比較した場合でも，提案手法がもっとも精度が高いと言える．また，相対誤差の最大値が提案手法においては大幅に改善されていることがわかる．

これらの評価実験の結果から，プロジェクト間の類似性に基づく予測手法によって欠損値を補完する提案手法は，比較対象の手法であるペアワイズ除去法，平均値挿入法によって欠損値処理を行った場合よりも高い精度で予測でき，より適切に欠損値の補完を行えていると言える．

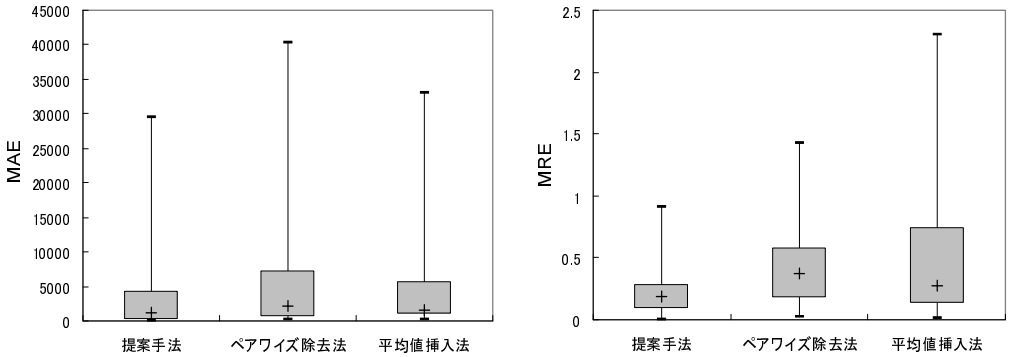


**図 3　各手法ごとの予測精度の箱ひげ図（全体）**

## 6　関連研究

提案手法のように，データセット中の非欠損部分の値を用いて欠損値を補完する方法は，memory based，あるいは likelihood な手法と呼ばれる．memory based な手法として，これまでにも k-nn 法 [2] [4] や similar response pattern imputation (SRPI) [6], HotDeck 法 [9] などを用いた手法などが提案されている．

従来の研究では，欠損値を正確に補完することに焦点があてられており，欠損のないデータセットに対して意図的に欠損値を設けてから補完を行い，真の値との比較を行っている．従来の結果からは，ランダムに値を欠損させた場合には，補完の精度が高いことが示されている．しかし，現実のデータセットの欠損値の発生は，ランダムというよりはむしろバースト的であり，欠損値の分布に大きな偏りがある．本稿では，欠損を多く含んだ現実のデータセットに対して補完を行ったことと，その効果を工数予測の精度として評価した点が特色である．

## 7　おわりに

本稿では，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て，プロジェクト間の類似性に基づいて欠損値を推定，補完を行う手法を提案した．評価実験の結果，提案手法によって欠損値を補完することで，従来手法であるペアワイズ除去法，平均値挿入法よりも高い精度で予測でき，欠損値を適切な値で補完できることが確認できた．

今後は，欠損値をより適切な値で補完できるように手法の改善を行う予定である．また，k-nn 法などの提案手法以外の memory based な欠損値補完法と比較する予定である．

## 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] B.W. Boehm, Software engineering economics, Prentice Hall, New Jersey, 1981.
[2] M. Cartwright, M.J. Shepperd, and Q. Song, ”Dealing with Missing Software Project Data,” Proc. 9th IEEE International Software Metrics Symposium (Metrics'03), pp.154-165, Sydney, Australia, 2003.
[3] ”ISBSG Estimating, Benchmarking and Research Suite Release 9,” International Software Benchmarking Standards Group, 2004, http://www.isbsg.org/

[4] P. Jonsson and C. Wohlin, "An evaluation of k-nearest neighbour imputation using likert data," Proc of the 10th IEEE International Software Metrics Symposium (Metrics'04), pp.108-118, Chicago, Illinois, 2004.
[5] J. Kromrey, and C. Hines, "Nonrandomly missing data in multiple regression: An empirical comparison of common missing-data treatments," Educational and Psychological Measurement, vol.54, no.3, pp.573-593, 1994.
[6] I. Myrtveit, E. Stensrud, and U.H. Olsson, "Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods," IEEE Trans. Software Eng., vol.27, no.11, pp.999-1013, 2001.
[7] M. Shepperd, and C. Schofield,"Estimating software project effort using analogies," IEEE Trans. Software Eng., vol.23, no.12, pp.736-743, 1997.
[8] 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.
[9] K. Strike, K. El Eman, and N. Madhavji, "Software cost estimation with incomplete data," IEEE Trans. Software Eng., vol.27, no.10, pp.890-908, 2001.
[10] 田村晃一，柿元健，戸田航史，角田雅照，門田暁人，松本健一, "プロジェクト間の類似性に基づくソフトウェアメトリクスの欠損値の補完," ソフトウェア信頼性研究会 第 4 回ワークショップ, pp.17-23, 2007.
[11] 角田雅照, 大杉直樹, 門田暁人, 松本健一, 佐藤慎一, "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.