

# 企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積り



大杉 直樹† 角田 雅照† 門田 暁人† 松村 知子† 松本 健一† 菊地 奈穂美†

プロジェクトの工数見積りのための基礎データとして、複数の異なる組織で収集されたソフトウェア開発プロジェクトのデータを利用できれば、蓄積データの少ない組織においても精度の高い見積りができると期待される。

本論文では、1つのケーススタディとして、独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センターが収集した多数のソフトウェア開発プロジェクトのデータを用いて工数見積りを行い、その精度を評価した。見積り手法として、重回帰分析、対数重回帰分析、ニューラルネットワーク、協調フィルタリングを用いた。評価の結果、協調フィルタリングは他の手法よりも高い精度を示し、実績工数に対する見積り値の相対誤差の平均値が0.642、Pred25（相対誤差が0.25以下のプロジェクトの割合）が30.1%であった。

## Using Cross-company Data to Estimate the Effort of Software Development Projects

Naoki Ohsugi†, Masateru Tsunoda†, Akito Monden†, Tomoko Matsumura†, Ken-ichi Matsumoto†, and Nahomi Kikuchi††

To predict software development effort, it would be useful to employ cross-company data collected outside a company, especially for the company who owns few project data. In this paper, as a case study, we conducted an effort prediction experiment using project data collected at the Software Engineering Center, Information Technology Promotion Agency, Japan. In the experiment we used linear regression, log-linear regression, neural network and collaborative filtering (CF) as prediction methods. As a result, the CF showed the best prediction performance, and its average MRE (Mean Relative Error) was 0.642 and the average Pred25 was 30.1%.

### 1 はじめに

ソフトウェアプロジェクト管理において、スケジュール管理及び要員配置を適切に行うためには、工数の見積りも高い精度で行うことが重要である[BOEHM1984]。そのために、数多くの工数見積り手法が提案されている[CONTE1986], [OHSUGI2004], [SHEPPERD1997],

[TAKADA1994], [WALSTON1977]。現在進行中のプロジェクト（現行プロジェクト）の完遂に要する工数を正確に見積ることができれば、プロジェクト管理者は、人的資源を効率的に配置し、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ことが可能となる。一般に、現行プロジェクトの工数を見積り際には、過去に実施されたプロジェクト（過去プロジェクト）のデータを見積りの根拠とする[CONTE1986]。プロジェクト

の特性を表す変数（特性変数）として、プロジェクトの完遂に要した工数や日数、要員のスキルや人数、開発に用いた技術や手法、開発された成果物の規模（ファンクションポイントやコード行数等）、発見されたバグの数等を、プロジェクトごとに記録し、蓄積する。蓄積した過去プロジェクトのデータから見積りモデルを作成し、現行プロジェクトから計測したデータをモデルに当てはめることで見積りを行う。

精度の高い（正確な）見積り結果を得るためには、大量のプロジェクトから、同種の特性変数をできるだけ多く、かつ一貫した計測方法により収集する必要がある。プロジェクトごとに収集されている特性変数の種類や、計測方法が異なる場合、プロジェクト間、並びに特性変数間の因果関係を正確に分析することができないため、見積り精度が低下する。そのため、データ収集を始める前に、収集する特性変数、収集の時期（開発のどの段階でデータを収集するのか）、収集に用いるツール、収集の手順等について、社内で統一した取り決めを行い、その取り決めに従ってデータを収集することが重要となる。

しかし、多くの企業において、首尾一貫した方法で大量のデータを収集することは容易ではない。一般に、企業全体で一貫したデータ収集の体制を整えるためには長い年月を要する[PAULK1993]。米Software Engineering Institute (SEI) は、準備を整えるまでに平均43ヶ月を要すると報告している[CMUSE2005]<sup>\*1</sup>。また、大量のデータを蓄積するためには、長期にわたってデータ収集を継続する必要がある[BRIAND2000]。収集活動に大きな工数を要する一方で、データの蓄積が少ない間は見積り精度が低く、得られる利益も小さい。早期に大きな利益を生みやすい活動に大きな工数を捻出することは、多くの会社にとって容易ではない。

蓄積データが少ない組織において、高い精度で工数見積りを行う1つの方法は、他の組織で収集されたソフトウェア開発プロジェクトのデータを見積りの根拠として利用することである。本論文では、その1つのケーススタディとして、多くの日本のソフトウェア開発企業の協力のもとにSECが収集したデータ（SECデータ）を用いて工数見積りを行い、その精度を評価する。SECデータ

には、2005年3月現在、日本のソフトウェア開発企業15社から収集されたプロジェクト1,009件分のデータが含まれている。プロジェクトごとに、要した工数や日数等、約490の特性変数が記録され、蓄積されている[IPASEC2005]。

ただし、複数の異なる組織からデータを収集した場合、計測された特性変数が組織ごとに異なり、欠損値（収集されていない値）が多く含まれる可能性が高くなる[MENDES2004]。SECデータにおいても、全体の87.3%が欠損値であった。一般に、30%を超える欠損値が含まれる場合、見積り精度が著しく低下するといわれている[KROMREY1994]。SECデータを用いる際には、この問題を解決し、多くの欠損値が含まれるデータを用いた場合でも、精度が低下しにくい見積り手法を適用することが望ましい。

現在までに、欠損値の問題を考慮した特性見積りの研究がいくつか行われている[OHSUGI2004], [STRIKE2001], [TSUNODA2005]。Strikeら[STRIKE2001]は、各種見積り手法を適用する際に欠損値の問題を解決する欠損値処理法（リストワイズ除去法、平均値挿入法等）を紹介し、その有効性を調査した。Strikeらは、代表的な見積り手法である重回帰分析に対して欠損値処理法を用い、欠損値が全体の15%以下の場合にはリストワイズ除去法、15%を超える場合は平均値挿入法等が有効であることを示した。Ohsugiら[OHSUGI2004]及び角田ら[TSUNODA2005]は、欠損値が多く含まれるデータに適した見積り手法として、協調フィルタリングに基づく見積り手法を提案した。そして、全体の60%が欠損値であるデータに提案手法を適用し、欠損値処理法を用いた重回帰分析よりも高い精度で見積りを行えることを確認した。

本論文では、見積り手法として、重回帰分析、対数重回帰分析[STRIKE2001]、ニューラルネットワーク[TAKADA1994]、協調フィルタリング[OHSUGI2004]を用い、SECデータにそれらを適用した場合の見積り精度の評価実験について報告する。協調フィルタリング以外の手法については、欠損値処理法として平均値挿入法を適用し、さらにステップワイズ法により、見積りに用いる特性変数を取捨選択した。一方、協調フィルタリングでは、欠損値を含む特

\*1 プロセス改善モデルとして著名なソフトウェア成熟度モデル（SW-CMM）によると、成熟度レベル3を達成することにより組織全体で一貫したデータを収集するための基盤が確立される。SEIは、成熟度レベル1から2への移行に平均23ヵ月、レベル2から3への移行に平均20ヶ月を要すると報告している。



性変数をありのまま用い、また、変数選択を行わず、すべての特性変数を用いた。実験では、相互検証法によって各手法の精度を評価し、精度に統計的な有意差があるかどうか、また、精度のばらつきはどの程度かを調査した。SECデータにおける各手法の見積精度を明らかにすることができれば、今後企業がSECデータを利用する際の指針になると期待される。

以降、2章では、本論文で用いた見積手法について説明する。3章では、精度評価実験の目的、評価指標、実験の手順について説明する。4章では、評価実験の結果と、結果に対する考察を述べる。5章で関連研究を紹介し、6章でまとめと今後の課題を述べる。

## 2 見積手法

### 2.1 重回帰分析

重回帰分析は多変量解析に基づく線形見積手法であり、ソフトウェア開発プロジェクトの工数や信頼性を見積もる際に広く利用されている[BRIAND 2000], [WALSTON 1977]。重回帰分析を適用する場合、特性変数間の依存関係を分析し、次の式(1)のような回帰式を作成する。

$$\hat{U}_{a,b} = C_1 U_{a,1} + C_2 U_{a,2} + \dots + C_n U_{a,n} + E \quad (1)$$

ただし、ここでは図1のような表形式のデータを用いて見積りを行うこととする。表中、 $p_1, p_2, \dots, p_m$ は各プロジェクト、 $m_1, m_2, \dots, m_n$ は各特性変数を表す。また、 $v_{i,j}$  ( $i=1, 2, \dots, m, j=1, 2, \dots, n$ )はプロジェクト $p_i$ について記録された変数 $m_j$ の値(特性値)を表す。式(1)で、 $\hat{U}_{a,b}$ はプロジェクト $p_a$ における変数 $m_b$ の見積値、 $U_{a,j}$ は見積りに影響する変数(説明変数)の値、 $c_j$ は係数、 $E$

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

図1 見積りに用いた  $m$  行  $n$  列の表

は定数項を表す。例えば、プロジェクト完遂に要する工数を  $m_b$  として見積もる場合、ソフトウェアの規模、要員のスキル等が説明変数の候補(説明変数候補)となる。

本論文では、説明変数候補から説明変数を選択する際に、ステップワイズ変数選択を行った。この方法では、次のような説明変数の採用と除去を、回帰式の変化がなくなるまで繰り返す[TANAKA1995]。まず、目的変数と最も相関が高い説明変数候補を1つ採用し、説明変数とする。次に、残る説明変数候補から、式の寄与度を最も大きく増加させるものを採用し、回帰式を作成する。作成された回帰式中の各説明変数の係数が0かどうか検定し、有意水準10%で棄却できる場合に当該変数を除去する。

評価実験を行う際には、平均値挿入法により欠損値の問題に対処した。平均値挿入法は、欠損値を各変数の平均値で埋める方法である[LITTLE1987]。欠損値を埋める方法としては、最も単純かつ一般的な方法である[SALTON1983]。

### 2.2 対数重回帰分析

対数重回帰分析は、重回帰分析を拡張した見積手法であり、プロジェクトの工数や信頼性を見積りに広く利用されている[BOEHM1984]。対数重回帰分析を適用する場合、次の式(2)のような回帰式を作成する。

$$\hat{U}_{a,b} = e^{(c_1 \log v_{a,1} + c_2 \log v_{a,2} + \dots + c_n \log v_{a,n} + E)} \quad (2)$$

対数重回帰分析を行う際は、特性値全てを底  $e$  ( $= 2.71828182845904$ ) で対数変換して新たなデータを作成し、そのデータに重回帰分析を適用すればよい。評価実験を行う際には、重回帰分析と同様の変数選択法と欠損値処理法を適用した。

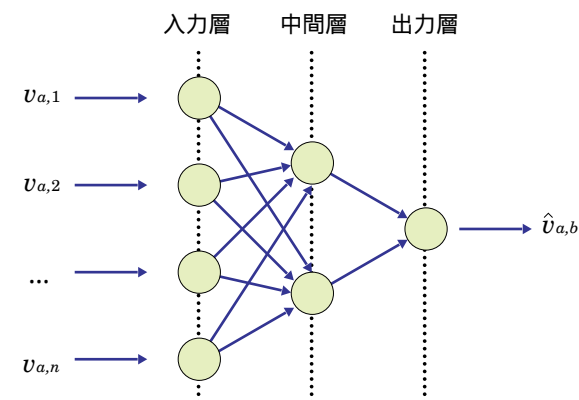


図2 見積りに用いたニューラルネットワーク

### 2.3 ニューラルネットワーク

ニューラルネットワークは、人工知能の分野で研究されてきた非線形見積手法である[FUNAHASHI1989], [ROSENBLATT1958], [RUMELHART1986]。重回帰分析は、説明変数が互いに独立であり、かつ、それらの線形結合によって目的変数との関係を表現できることを前提としている。しかし、プロジェクトの工数や信頼性の予測においては、説明変数は相互に影響を及ぼしあっている場合がほとんどであり、また、説明変数と目的変数の関係が線形であるという理論的根拠もない。一方、ニューラルネットワークは、説明変数の独立性は前提とされておらず、かつ説明変数間の非線形な関係を捉えることができるため、重回帰分析よりも高い精度で見積りを行える可能性がある[TAKABAYASHI1999], [TAKADA1994]。

評価実験で使用したニューラルネットワークは図2に示すように入力層、中間層、出力層からなる。各層の処理ユニットは、隣接する層の処理ユニットと重み付きリンクで接続されている。入力層から説明変数の値が入力され、出力層から見積値が出力される。入力層の各処理ユニットに入力された説明変数の値は、リンクの重みが乗算され、中間層の処理ユニットに送られる。中間層では、送られてきた値が和算された後、シグモイド関数 ( $g(\theta) = 2 / (1 + \exp(-\theta)) - 1$ ) により非線形変換される。変換された値は、再びリンクの重みが乗算され、出力層に送られる。出力層に送られた値の和がニューラルネットワークの出力となる[TAKADA1994]。

リンクの重みの決定には、様々な学習アルゴリズムが提案されている。本論文では、米Integral Solutions社の統計処理ソフトウェアClementine 8.6に実装されている誤差逆伝播アルゴリズム[RUMELHART1986]を使用した。この方法では、既知のデータを使用し、次のようなリンクの重みの更新を繰り返す。まずニューラルネットワークの出力と見積値との間の残差平方和を計算する。次に、リンクの重みに対する残差平方和の勾配を計算し、残差平方和を減少させる方向にリンクの重みを変化させる[TAKADA1994]。

学習に影響を与えるパラメータとして、中間層に含まれる処理ユニットの個数及び重みを更新する回数(学習回数)を調整する必要がある。処理ユニットの個数を増やすと、より複雑な関係を表現できる。しかし、リンクの重みが多くなるため、学習に使用するデータの量が少

なければ、見積精度が低下する[FUNAHASHI1989]。また、処理ユニットの個数の増加に応じて、より多くの学習回数を設定する必要がある。評価実験を行う際には、中間層の処理ユニットの個数を2と3に変動させ、学習回数を10,000と50,000に変動させた。評価実験を行う際には、重回帰分析と同様の変数選択法と欠損値処理法を適用した。

### 2.4 協調フィルタリング

協調フィルタリングは、情報検索の分野で研究されてきた見積手法である[GOLDBERG1992], [RESNICK1994], [SARWAR2001]。協調フィルタリングを用いることで、過去に手がけた類似の案件の実績データをもとに、新規プロジェクトの工数を見積もる「類推見積り」を、系統的に行うことができる。重回帰分析やニューラルネットワークでは、ただ1つの予測式が作成されるのに対し、協調フィルタリングは予測対象プロジェクトごとに個別に予測式を構築するため、プロジェクトの個性をより強く反映した予測が行える。また、欠損値の量や分布(欠損値の偏り)が変化しても精度は低下しにくいという特徴がある[KAKIMOTO2004]。SECデータには多くの欠損値が含まれるため、重回帰分析やニューラルネットワークよりも高い精度で見積りを行える可能性がある。

協調フィルタリングは、特性値の正規化、プロジェクト間の類似度計算、類似度に基づく見積値計算という3つの手順で実行される。各手順について、複数のアルゴリズムが提案されており、データの性質に応じて用いるアルゴリズムを変更する必要がある。精度評価実験では、手順(1)と(2)で用いるアルゴリズムを1つに固定し、とくに精度への影響が大きい手順(3)のアルゴリズムを変動させた。各手順で用いたアルゴリズムについて、以降で説明する。

(1) 特性値の正規化: データ中の各特性変数は値域が互いに異なる。各変数が見積りに対して与える影響を均一にするため、次式により、値域  $[0.0, 1.0]$  に正規化した特性値  $v_{i,j}'$  を求めた。

$$v_{i,j}' = \frac{v_{i,j} - \min(m_j)}{\max(m_j) - \min(m_j)} \quad (3)$$

ただし、ここでは図1のような表形式のデータを用いて見積りを行うこととする。 $v_{i,j}$  はプロジェクト  $p_i$  につ



いての特性変数 $m_j$ の値,  $\max(m_j)$ は変数 $m_j$ の最大値,  $\min(m_j)$ は変数 $m_j$ の最小値を表す.

□ プロジェクト間の類似度計算: 見積り対象のプロジェクト $p_a$ と見積りの根拠として用いる他の各プロジェクト $p_i$ との類似度 $\text{sim}(p_a, p_i)$ を求める. 実験の際には, 代表的なアルゴリズムであるAdjusted Cosine Similarity (式□)とCorrelation Coefficient (式□)を適用した[RESNICK1994].

$$\text{sim}(p_a, p_i) = \frac{\sum_{j \in M_a \cap M_i} \left\{ (v_{a,j'} - \overline{m_j'}) \times (v_{i,j'} - \overline{m_j'}) \right\}}{\sqrt{\sum_{j \in M_a \cap M_i} (v_{a,j'} - \overline{m_j'})^2} \sqrt{\sum_{j \in M_a \cap M_i} (v_{i,j'} - \overline{m_j'})^2}} \quad \square$$

$$\text{sim}(p_a, p_i) = \frac{\sum_{j \in M_a \cap M_i} \left\{ (v_{a,j'} - \overline{p_a'}) \times (v_{i,j'} - \overline{p_i'}) \right\}}{\sqrt{\sum_{j \in M_a \cap M_i} (v_{a,j'} - \overline{p_a'})^2} \sqrt{\sum_{j \in M_a \cap M_i} (v_{i,j'} - \overline{p_i'})^2}} \quad \square$$

ただし,  $M_a$ と $M_i$ はそれぞれプロジェクト $p_a$ と $p_i$ について記録された特性変数の集合を表す. また,  $\overline{m_j'}$ と $\overline{p_i'}$ は特性変数 $m_j'$ とプロジェクト $p_i'$ の正規化された特性値の平均値を表す. 類似度 $\text{sim}(p_a, p_i)$ の値域は[-1.0, 1.0]である. 式□, □では欠損していない変数のみを用いるため, 欠損値処理法を用いる必要はない.

□ 類似度に基づく見積値計算: 手順□で求めた類似度 $\text{sim}(p_a, p_i)$ を用いて, プロジェクト $p_a$ の特性変数 $m_b$ の見積値 $\hat{v}_{a,b}$ を計算する. 実験では, 式□に示すAmplified Weighted Sumを適用して見積値計算を行った.

$$\hat{v}_{a,b} = \frac{\sum_{j \in k\text{-nearest Projects}} (v_{a,j'} \times \text{amplifer}(p_a, p_i) \times \text{sim}(p_a, p_i))}{\sum_{j \in k\text{-nearest Projects}} \text{sim}(p_a, p_i)} \quad \square$$

$$\text{amplifer}(p_a, p_i) = \frac{\sum_{j \in M_a \cap M_i} \left( \frac{v_{a,j'}}{v_{i,j'}} \times \text{correl}(m_b, m_j) \right)}{\sum_{j \in M_a \cap M_i} \text{correl}(m_b, m_j)} \quad \square$$

ただし, k-nearestProjectsは, 特性変数 $m_b$ の値が収集されており, かつプロジェクト $p_a$ と類似度が高い $k$ 個のプロジェクトの集合を表す. 類似プロジェクト数 $k$ は見積精度に影響を与えるため, 評価実験では,  $k$ を3から21の間で3ずつ変動させ, 最も精度が高くなる $k=15$ を採用した. また,  $\text{correl}(m_b, m_j)$ は特性変数 $m_b$ と $m_j$ の相関係数を表す. ただし, 相関係数が負の値になる場合は0.0と

して計算した.

### 3 精度評価実験

#### 3.1 目的

今後多くの企業がSECデータを利用する際の1つの指針として, SECデータにおける各手法の見積精度を明らかにすることである. 本実験では, 代表的な見積手法である重回帰分析, 対数重回帰分析, ニューラルネット及び近年有望視されている協調フィルタリングを評価対象とした. ただし, ステップワイズ重回帰分析とニューラルネットについては, データに欠損値が含まれる場合に見積りが行えないため, 欠損値処理法(平均値挿入法)を適用した上で, ステップワイズ法による変数選択を行った. 精度の評価指標は絶対誤差及び相対誤差を用いた. 相互検証法(ホールドアウト法)によって10回の試行を行い, 各手法の精度及び精度のばらつきを評価, 比較した.

#### 3.2 利用したデータ

評価実験ではSECデータを用いて, プロジェクトの完遂に要する工数(人時)の見積りを行った. 表1に実験で用いたデータに含まれる特性変数を示す. 見積対象の変数は $m_0$ であり, 見積りに用いた変数は $m_1 \sim m_{97}$ である. 本来, SECデータには約490の変数が含まれている[IPASEC2005].

ここでは, 詳細設計完了時に変数 $m_0$ を見積る場合を想定し, その時点で収集が終了していると思われる97個の特性変数を選出した. なお, 総開発期間(月数)の実績数 $m_1$ は, 実際には開発が終了しないと確定しないが, ここでは, 開発期間(納期)が開発開始時にあらかじめ決定されることを想定し, 特性変数に含めている.

選出した変数には, 量的変数と質的変数が含まれている. 前者は数値の大小に意味があり, 加減算が可能な変数である. 例えば, 工数200人時は150人時に対して50人時の差がある. 一方, 後者は数値の大小に意味がない変数である. 例えば, プロジェクト種別1(新規開発)とプロジェクト種別4(拡張)に3の差があるわけではない. 選出した変数のうち $m_0 \sim m_{29}$ は量的変数,  $m_{30} \sim m_{97}$ は質的変数である. ここでは, 質的変数をダミー変数化して共分散分析を行うことはせず, 量的変数 $m_1 \sim m_{29}$ のみ

を用いて重回帰分析を行った.

見積精度を評価する際, 見積もった値(見積値)と実際に収集された $m_0$ の値(実測値)を比較するため, SECデータに含まれるプロジェクト全1,009件から, 変数 $m_0$ が収集されているプロジェクト378件のデータを抽出し, 実験データとして使用した. 実験データには数社のプロジェクトが混在していた. 全ての企業から収集された特性変数もあるが, いくつかの企業では収集されなかった変数もある. このため, データには多くの欠損値が含まれていた.

表1に実験データに含まれる欠損値の割合(欠損率)を特性変数ごとに示す. 各特性変数の欠損率は, 実験データ全378件のプロジェクト中, 当該変数を記録しなかったプロジェクトの割合と等しい. 実験データ全体の欠損率は約67%と非常に大きく, その分布にも大きな偏りがある. 見積りに用いた変数 $m_1 \sim m_{97}$ の内, 欠損率が20%以下のものは全体の約9%しかない. 一方で, 欠損率が80%以上の変数は全体の約42%に及ぶ. また, 変数 $m_4, m_{30}$ 等は欠損値を全く含まない. 一方で, 変数 $m_2, m_3$ 等は全体の90%以上が欠損値である.

#### 3.3 評価指標

実験では, 見積精度を評価する基準として, 次の5つの評価指標を用いた. ただし, ここでは図1のような表形式のデータを用いて見積りを行うこととする.  $\hat{v}_{a,b}$ はプロジェクト $p_a$ における特性変数 $m_b$ の見積値,  $v_{a,b}$ はその実測値を表す. また,  $m$ は見積もったプロジェクトの数を表す.

絶対誤差平均(MAE): 実測値と見積値の誤差の平均. 値域は[0.0,  $\infty$ )であり, 値が小さいほど見積精度が高いことを示す.

$$\text{MAE} = \frac{1}{m} \sum_{a=1}^m \left| \hat{v}_{a,b} - v_{a,b} \right| \quad \square$$

相対誤差平均(MRE): 誤差が実測値の何倍であるかを表す相対誤差の平均. 値域は[0.0,  $\infty$ )であり, 値が小さいほど見積精度が高いことを示す.

$$\text{MRE} = \frac{1}{m} \sum_{a=1}^m \left| \frac{\hat{v}_{a,b} - v_{a,b}}{v_{a,b}} \right| \quad \square$$

Pred25: 見積値全体に占める, 相対誤差0.25以下の見積

値の割合. 値域は[0.0, 1.0]であり, 値が大きいほど, 見積精度が高いことを示す.

$$\text{Pred25} = \frac{1}{m} \sum_{a=1}^m \text{isAccurate} \left( \frac{\hat{v}_{a,b} - v_{a,b}}{v_{a,b}} \right) \quad \square$$

$$\text{isAccurate}(x) = \begin{cases} 1 & |x| \leq 0.25 \\ 0 & |x| > 0.25 \end{cases} \quad \square$$

#### 3.4 手順

実験では, 相互検証法(ホールドアウト法)によって各手法の見積精度を評価した. 2節で説明した見積手法ごとに, 次の手順で評価指標を計算した.

Es1. 実験データを無作為に半数ずつに分割し, 一方を学習データ, もう一方を試験データとする.

Es2. 学習データを用いて見積モデルを構築する.

Es3. Es2で作成した見積モデルを用いて試験データ中に含まれるプロジェクトの特性値 $\hat{v}_{a,b}$ ( $a=0, 1, \dots, m$ )を見積もる. ただし, この際に試験データ中の実測値 $v_{a,0}$ は見積りに使用しない.

Es4. Es3で得た各見積値 $\hat{v}_{a,b}$ と, 試験データ中の各実測値 $v_{a,0}$ を比較し, 評価指標を計算する.

Es5. 実験結果の信頼性を向上させるため, Es1からEs4を1回の試行として10回繰り返す. 各手法について, 10回の試行から得た評価指標の平均値を算出する.

各手法について算出した評価指標の平均値の差の検定を行い, 精度に統計的な有意差があるかどうかを調査した. また, 10回の試行で得た各評価指標について箱ひげ図(boxplot)を描き, 各手法を用いた場合の精度のばらつきを比較した. 得られた結果に基づき, その意味と限界について考察を行った.

## 4 実験の結果と考察

評価実験の結果, 10回の試行から得られた各評価指標の平均値を表2に示す. 表2では, 各列に評価指標を記し, 各行にはMREの値が小さい(精度が高い)ものから順に見積手法を記した. 行と列が交差するセルには, 各評価指標の平均値と共に( )内に直下のセルに記した平均値との差の検定によって算出した $p$ 値を記した. また, 各セルに記した値が, 直下のセルに記した値よりも有意水準5%で有意に精度が高い場合, そのセルの値を斜体と下線によって強調した.

表1 評価実験に用いたデータに含まれる特性値と各特性の欠損率

特性値の種類	欠損率	特性値の種類	欠損率
<i>m</i> <sub>0</sub> 実績工数#総計人時#_プロジェクト全体	0.00%	<i>m</i> <sub>9</sub> アーキテクチャ	0.00%
<i>m</i> <sub>1</sub> 月数#実績#プロジェクト全体	42.86%	<i>m</i> <sub>10</sub> 開発対象プラットフォーム	33.60%
<i>m</i> <sub>2</sub> 利用者数	97.88%	<i>m</i> <sub>11</sub> Web技術の利用	44.97%
<i>m</i> <sub>3</sub> 利用拠点数	97.62%	<i>m</i> <sub>12</sub> オンライントランザクション処理	96.83%
<i>m</i> <sub>4</sub> FP実測値_調整前	0.00%	<i>m</i> <sub>13</sub> 主開発言語	21.16%
<i>m</i> <sub>5</sub> 改修FP値_母体FP	89.15%	<i>m</i> <sub>14</sub> DBMSの利用	47.88%
<i>m</i> <sub>6</sub> ILF実績値_FP	60.32%	<i>m</i> <sub>15</sub> 開発ライフサイクルモデル	0.00%
<i>m</i> <sub>7</sub> EIF実績値_FP	61.90%	<i>m</i> <sub>16</sub> 運用ツール利用	91.27%
<i>m</i> <sub>8</sub> トランザクションファンクション実績値_機能数	74.07%	<i>m</i> <sub>17</sub> 類似プロジェクト_有無	91.53%
<i>m</i> <sub>9</sub> トランザクションファンクション実績値_FP	65.34%	<i>m</i> <sub>18</sub> プロジェクト管理ツール_利用	65.34%
<i>m</i> <sub>10</sub> データファンクション実績値_機能数	73.28%	<i>m</i> <sub>19</sub> 構成管理ツール利用	69.05%
<i>m</i> <sub>11</sub> データファンクション実績値_FP	63.23%	<i>m</i> <sub>20</sub> 設計支援ツール利用	69.05%
<i>m</i> <sub>12</sub> 設計書文書量要件定義書	81.22%	<i>m</i> <sub>21</sub> ドキュメント作成ツール利用	69.05%
<i>m</i> <sub>13</sub> 設計書文書量基本設計書	81.22%	<i>m</i> <sub>22</sub> デバッグ_テストツール利用	69.05%
<i>m</i> <sub>14</sub> 設計書文書量詳細設計書	80.69%	<i>m</i> <sub>23</sub> 上流CASEツール利用	97.09%
<i>m</i> <sub>15</sub> 他規模指標_DBテーブル数	82.80%	<i>m</i> <sub>24</sub> 統合CASEツール利用	97.09%
<i>m</i> <sub>16</sub> 他規模指標_画面数	81.22%	<i>m</i> <sub>25</sub> コードジェネレータ利用	95.77%
<i>m</i> <sub>17</sub> 他規模指標_帳票数	83.07%	<i>m</i> <sub>26</sub> 開発方法論利用	95.24%
<i>m</i> <sub>18</sub> 月数#実績#要件定義	76.72%	<i>m</i> <sub>27</sub> 要求仕様_明確度合	52.12%
<i>m</i> <sub>19</sub> 月数#実績#基本設計	77.25%	<i>m</i> <sub>28</sub> ユーザ担当者_要求仕様関与	52.65%
<i>m</i> <sub>20</sub> 月数#実績#詳細設計	66.93%	<i>m</i> <sub>29</sub> ユーザ担当者_システム経験	69.84%
<i>m</i> <sub>21</sub> 実績工数#開発#システム化計画	83.86%	<i>m</i> <sub>30</sub> 要求仕様_ユーザ承認有無	97.35%
<i>m</i> <sub>22</sub> 実績工数#開発#要件定義	37.83%	<i>m</i> <sub>31</sub> ユーザ担当者_設計内容理解度	97.35%
<i>m</i> <sub>23</sub> 実績工数#開発#基本設計	21.43%	<i>m</i> <sub>32</sub> 設計_ユーザ承認有無	97.35%
<i>m</i> <sub>24</sub> 実績工数#開発#詳細設計	26.19%	<i>m</i> <sub>33</sub> ユーザ担当者_受け入れ試験関与	69.05%
<i>m</i> <sub>25</sub> 外注実績#工数#要件定義	92.06%	<i>m</i> <sub>34</sub> 要求レベル_信頼性	67.72%
<i>m</i> <sub>26</sub> 外注実績#工数#基本設計	83.07%	<i>m</i> <sub>35</sub> 要求レベル_使用性	97.09%
<i>m</i> <sub>27</sub> 外注実績#工数#詳細設計	87.30%	<i>m</i> <sub>36</sub> 要求レベル_性能・効率性	61.64%
<i>m</i> <sub>28</sub> レビュー指摘件数基本設計	98.94%	<i>m</i> <sub>37</sub> 要求レベル_保守性	97.09%
<i>m</i> <sub>29</sub> レビュー指摘件数詳細設計	99.47%	<i>m</i> <sub>38</sub> 要求レベル_移植性	97.35%
<i>m</i> <sub>30</sub> 開発プロジェクト種別	0.00%	<i>m</i> <sub>39</sub> 要求レベル_ランニングコスト要求	97.35%
<i>m</i> <sub>31</sub> 母体システム安定度	70.11%	<i>m</i> <sub>40</sub> 要求レベル_セキュリティ	67.99%
<i>m</i> <sub>32</sub> 開発プロジェクト形態	0.00%	<i>m</i> <sub>41</sub> 法的規制有無	69.58%
<i>m</i> <sub>33</sub> 受託開発作業場所	64.02%	<i>m</i> <sub>42</sub> PMSスキル	67.72%
<i>m</i> <sub>34</sub> 新規顧客	86.77%	<i>m</i> <sub>43</sub> 要員スキル_業務分野経験	61.90%
<i>m</i> <sub>35</sub> 新規業種・業務	86.77%	<i>m</i> <sub>44</sub> 要員スキル_分析・設計経験	67.99%
<i>m</i> <sub>36</sub> 新規協力会社	88.62%	<i>m</i> <sub>45</sub> 要員スキル_言語・ツール利用経験	61.90%
<i>m</i> <sub>37</sub> 新技術利用	87.83%	<i>m</i> <sub>46</sub> 要員スキル_開発プラットフォーム使用経験	61.90%
<i>m</i> <sub>38</sub> 役割分担_責任所在	67.46%	<i>m</i> <sub>47</sub> 主なFP計測手法	4.76%
<i>m</i> <sub>39</sub> 達成目標_優先度_明確度合	67.99%	<i>m</i> <sub>48</sub> FP計測手法実績値の純度	5.03%
<i>m</i> <sub>40</sub> 作業スペース	67.99%	<i>m</i> <sub>49</sub> FP計測_支援技術	47.62%
<i>m</i> <sub>41</sub> プロジェクト環境_騒音	67.99%	<i>m</i> <sub>50</sub> テスト体制	71.16%
<i>m</i> <sub>42</sub> 業種	29.63%	<i>m</i> <sub>51</sub> 定量的出荷品質基準_有無	93.92%
<i>m</i> <sub>43</sub> 業務種類	57.41%	<i>m</i> <sub>52</sub> SLOC実測値_コメント行取り扱い	85.19%
<i>m</i> <sub>44</sub> システム用途	95.24	<i>m</i> <sub>53</sub> SLOC実測値_空行取り扱い	85.19%
<i>m</i> <sub>45</sub> 利用形態	0.00%	<i>m</i> <sub>54</sub> 品質保証体制_基本設計	91.53%
<i>m</i> <sub>46</sub> システム種別	0.00%	<i>m</i> <sub>55</sub> 品質保証体制_詳細設計	90.74%
<i>m</i> <sub>47</sub> 業務パッケージ_利用有無	29.89%	<i>m</i> <sub>56</sub> 品質保証体制_結合テスト	89.95%
<i>m</i> <sub>48</sub> 処理形態	61.64%	<i>m</i> <sub>57</sub> 品質保証体制_総合テスト(ベンダ確認)	91.01%

表2 10回の試行から得た評価指標の平均値 .MAE ,MREは値が小さいほど精度が高いことを示す .Pred25は値が大きいほど精度が高いことを示す ( )内は直下のセルの値との差の検定により算出したp値を示す .

	MAE	MRE	Pred25
協調フィルタリング ( Adjusted Cosine Similarity )	<u>5637.185</u> (0.017)	<u>0.642</u> (0.000)	0.301 (0.069)
協調フィルタリング ( Correlation Coefficient )	<u>5863.709</u> (0.002)	<u>0.771</u> (0.417)	<u>0.286</u> (0.001)
対数重回帰分析	8102.984 (0.439)	<u>0.855</u> (0.014)	0.180 (0.129)
ニューラルネット ( 3ユニット , 50,000回 )	9096.571 (0.151)	<u>1.281</u> (0.222)	0.221 (0.261)
ニューラルネット ( 2ユニット , 50,000回 )	<u>8020.968</u> (0.010)	<u>1.428</u> (0.339)	0.218 (0.369)
ニューラルネット ( 2ユニット , 10,000回 )	8135.367 (0.640)	<u>1.473</u> (0.595)	0.209 (0.426)
ニューラルネット ( 3ユニット , 10,000回 )	<u>10279.519</u> (0.014)	<u>1.584</u> (0.002)	<u>0.201</u> (0.000)
重回帰分析	12315.063	6.247	0.101

結果として、実験データに対しては協調フィルタリングの見積精度が最も高いことがわかった。対数重回帰分析、ニューラルネットは評価指標によって精度の優劣が異なり、また統計的な有意差もMREを除いて観察できなかった。また、重回帰分析は最も見積精度が低いことが分かった。協調フィルタリングを適用する場合には、Adjusted Cosine Similarityを類似度計算アルゴリズムとして用いた場合の方がCorrelation Coefficientを用いた場合よりも高い精度を示した。ニューラルネットを適用する場合には、中間層のユニット数を3、学習回数を50,000回とした場合が最も高い精度を示した。

評価実験の結果、10回の試行から得られた各評価指標のばらつきを図3、図4、図5にboxplotとして記す。図中では、ニューラルネットをNN、協調フィルタリングをCFと略記した。図では、横軸に各手法を列挙し、縦軸が各評価指標の値を表す。各手法について、最小値、第1四分位数、中央値、第3四分位数、最大値を求め、対応する軸に記した。図によると、協調フィルタリング、とくにAdjusted Cosine Similarityを類似度計算アルゴリズムとして用いた場合に、ばらつきなく高い精度で見積りを行えたことがわかる。さらに、対数重回帰分析はMREのばらつきがニューラルネットよりも小さく、ニューラルネットはMAEやPred25のばらつきが対数重回帰分析より

も小さいことがわかる。また、重回帰分析については、他の手法より精度が低く、ばらつきも大きいことがわかる。

評価実験では、協調フィルタリングが他の見積手法よりも高い見積精度を示した。これは、過去の類似のプロジェクトのデータに基づいて類推見積りを行うという協調フィルタリングのアプローチが有効であったことに加え、実験データに含まれる欠損値が他の手法の精度を低下させたためと考えられる。重回帰分析やニューラルネットを用いた場合、学習データに基づいて見積り対象の特性変数(今回の場合はプロジェクト完遂に要する工数)に大きな影響を与える変数(今回の場合はプロジェクトの月数等)を調べ、数学的な見積モデルを作成する。見積り対象のプロジェクトで、見積りに大きな影響を与える特性変数が欠損している場合、欠損値処理法によって挿入された平均値などに基づいて見積りが行われるため、精度が著しく低下する。一方、協調フィルタリングの場合には、欠損していない変数だけを用いて見積りを行うため、欠損値が多いプロジェクトに対してもある程度高い見積精度を示したものと考えられる。

今回の実験において最も精度が高かった協調フィルタリングでは、MREが0.642、Pred25が30.1%であった。これらの数値は、1つの企業内だけで収集されたデータに基づく工数予測事例と比べても、遜色のない精度である。



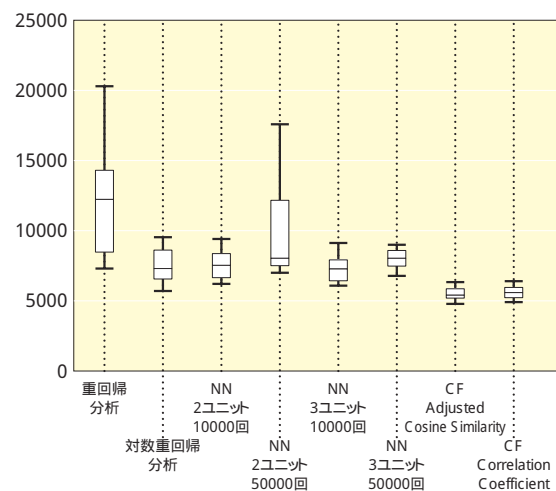


図3 MAEのboxplot

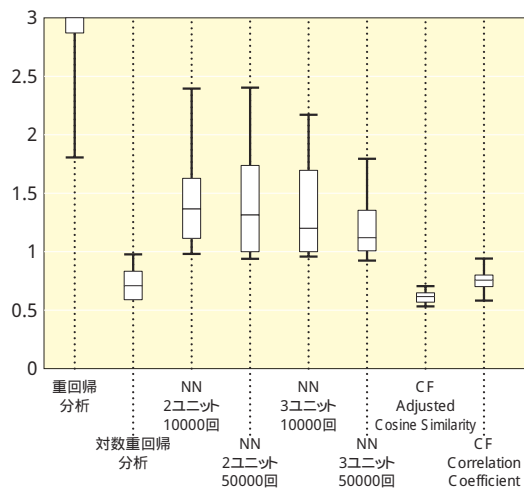


図4 MREのboxplot

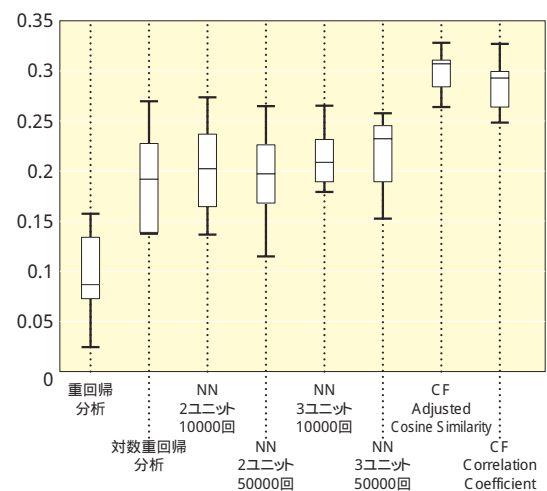


図5 Pred25のboxplot

参考となる事例として、文献[TSUNODA2005]では企業内データを用いて、試験工程開始までに得られるプロジェクト特性値に基づいて試験工数の見積りを行っており、そのMREが0.79, Pred25が37%であった。実験条件は異なるものの、本実験ではより早い工程（詳細設計完了時）での見積りを行ったにもかかわらず、遜色のない精度が得られたといえる。

ただし、MREが0.642であるということは、工数の見積り値が平均して64.2%の誤差を含むということであり、現場へ適用するには注意が必要である。今後、さらなる分析が必要であるが、相対誤差が大きいプロジェクトの多くは、規模が小さなものであった。ある程度以上の規模を持つプロジェクトであれば、より精度の高い見積り値が得られる可能性がある。

また、今回の実験は、あくまでも1つのケーススタディであり、SECデータから抽出した378件のプロジェクトの特性値を用いた場合の結果である点に注意する必要がある。SECと同様、多数の異なる企業からプロジェクトのデータを収集している事例として、海外ではISBSGが知られている[ISBSG]。これら他のデータセットを用いて本論文と同様の評価実験を行うことが今後の課題である。

## 5 関連研究

単一企業から収集したデータを用いた場合と、複数企業から収集したデータを用いた場合の見積精度を比較する研究が行われている。複数企業からデータを収集した方が多くのデータを利用できる反面、データの一貫性が低下しやすく、見積精度が低下する可能性が高い[BRIAND2000]。今回、実験に用いたデータには複数企業のデータが含まれていたが、単一企業のデータのみを用いることで見積精度を改善できる可能性がある。複数企業のデータを用いた場合の見積精度と、単一企業のデータを用いた場合の見積精度を比較調査することは、今後の課題の1つである。

本論文で取り上げた見積手法以外にも、ソフトウェア開発プロジェクトの特性を見積もる手法が提案されている。Amasakiら[AMASAKI2003]は、重回帰分析等の見積手法を用いる場合に問題となる欠損値の問題を解決するため、ベイジアンネットを用いた特性の見積手法を提案し

た。Amasakiらは実プロジェクトから収集したデータに対して提案手法を適用し、ソフトウェアの品質見積りに利用可能であることを確認した。SECデータには非常に多くの欠損値が含まれるため、ベイジアンネットによって高い精度で見積りを行える可能性がある。ベイジアンネットをSECデータに適用した場合の精度を評価することは、今後の課題の1つである。

また、本論文で取り上げた欠損値処理法以外にも、いくつかの欠損値処理法が提案されている。Strikeら[STRIKE 2001]は、データの欠損率を変化させ、平均値挿入法を含む3種類の欠損値処理法を重回帰分析と共に適用した場合の精度を調査した。他の欠損値処理法をSECデータに適用し、見積精度を評価することも、今後の課題の1つである。

## 6 おわりに

本論文では、企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積りの1つのケーススタディとして、SECデータを用いて複数の手法により工数の見積りを行い、その精度を評価した。その結果、協調フィルタリングは他の3つの手法（重回帰分析、対数重回帰分析、ニューラルネット）よりも高い精度を示し、実績工数に対する見積り工数の相対誤差の平均値が0.642, Pred25が30.1%となった。

今後は、ISBSGなど他のデータセットを用いた場合との比較、単一企業のデータのみを用いた場合との比較、ベイジアンネット等の他の見積手法の評価等を行ってきたい。

### 謝辞

本研究の一部は、SECとの共同研究の成果並びに文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託により行われた研究成果に基づく。本論文の執筆に際し、見積手法や評価実験に関するご意見をいただいた奈良先端科学技術大学院大学情報科学研究科 柿元健氏に深く感謝する。また、統計手法全般にわたってご助言をいただいた同志社大学文化情報学部 宿久洋博士に深く感謝する。

### 参考文献

- [AMASAKI2003] Amasaki, S., Mizuno, O., Kikuno, T., and Takagi, Y.: A Bayesian Belief Network for Predicting Residual Faults in Software Products, In Proc. of the 14th Int'l Symposium on Soft. Reliability Eng., pp.215-226, 2003
- [BOEHM1984] Boehm, B.W.: Software Engineering Economics, IEEE Trans. on Soft. Eng., vol.10, no.1, 4-21, 1984
- [BRIAND2000] Briand, L.C., Langley, T., Wiecek, I.: A Replicated Assessment of Common Software Cost Estimation Techniques. In Proc. of the 22nd Int'l Conf. on Soft. Eng., pp.377-386, 2000
- [CMUSEI2005] Carnegie Mellon University and Software Engineering Institute: Process Maturity Profile: Software CMM2004 Year End Update, 2005
- [CONTE1986] Conte, S.D., Dunsmore, H.E., and Shen, V.Y.: Software Engineering Metrics and Models. The Benjamin/Cummings Publishing Company, Inc., Menlo Park, California, 1986
- [FUNAHASHI1989] Funahashi, K.: On the Approximate Realization of Continuous Mappings by Neural Networks, Neural Networks, vol.2, no.3, pp.183-192, 1989
- [GOLDBERG1992] Goldberg, D., Nichols, D., Oki, B.M., and Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry, Comm. of the ACM, vol.35, no.12, pp.61-70, 1992
- [IPASEC2005] 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター: ソフトウェア開発データ白書2005 - IT企業1000プロジェクトの定量データを徹底分析 -, 日経BP社, 2005
- [ISBSG] Software Benchmarking Standards ISBSG Homepage, <http://www.isbsg.org/>
- [KAKIMOTO2004] 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りのロバスト性評価, 第11回ソフトウェア工学の基礎ワークショップ, 2004
- [KROMREY1994] Kromrey, J., and Hines, C.: 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
- [LITTLE1987] Little, R., and Rubin, D.: Statistical Analysis with Missing Data, John Wiley & Sons, Inc., 1987
- [MENDES2004] Mendes, E., and Kichenham, B. A.: Further Comparison of Cross-Company and Within Company Effort Estimation Models for Web Applications, In Proc. of the 10th Int'l Soft. Metrics Symposium, pp.348-357, 2004
- [OHSUGI2004] Ohsugi, N., Tsunoda, M., Monden, A., and Matsumoto, K.: Applying Collaborative Filtering for Effort Estimation with Process Metrics, In Proc. of the 5th Int'l Conf. on Product Focused Soft. Process Improvement, Springer, Berlin Heidelberg, pp.274-286, 2004
- [PAULK1993] Paulk, M., Curtis, B., Chrissis, M., and Weber, C.: Capability Maturity Model for Software (Version 1.1), CMU/SEI-93-TR-024, 1993
- [RESNICK1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews, In Proc. of the ACM Conf. on Computer Supported Cooperative Work, 175-186, 1994
- [ROSENBLATT1958] Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review, vol.65, No.6, pp.386-408, 1958
- [RUMELHART1986] Rumelhart, D.E., Hinton, G.E., and Williams, R.J.: Learning Representations by Back-propagating Errors, Nature, vol.323, pp.533-536, 1986
- [SALTON1983] Salton, G., and McGill, M.: Introduction to Modern Information Retrieval, p.448, McGraw-Hill, New York, 1983
- [SARWAR2001] Sarwar, B.M., Karypis, G., Konstan, J.A., and Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms, In Proc. of the 10th Int'l World Wide Web Conf., pp.285-295, 2001
- [SHEPPERD1997] Shepperd, M., and Schofield, C.: Estimating Software Project Effort Using Analogies, IEEE Trans. on Soft. Eng., vol.23, no.12, pp.76-743, 1997
- [STRIKE2001] Strike, K., El Eman, K., and Madhavji, N.: Software Cost Estimation with Incomplete Data, IEEE Trans. on Soft. Eng., vol.27, no.10, pp.890-908, 2001
- [TAKABAYASHI1999] Takabayashi, S., Monden, A., Sato, S., Matsumoto, K., Inoue, K., and Torii, K.: The Detection of Fault-Prone Program Using a Neural Network, In Proc. of the Int'l Symposium on Future Software Technology '99, pp.81-86, 1999
- [TAKADA1994] 高田義広, 松本健一, 鳥居宏次: ニューラルネットを用いたソフトウェア信頼性予測モデル, 電子情報通信学会論文誌D-I, vol.J77-D-1, no.6, pp.454-461, 1994
- [TANAKA1995] 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
- [TSUNODA2005] 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, 情報処理学会論文誌, vol.46, no.5, pp.1155-1164, 2005
- [WALSTON1977] Walston, C., and Felix, C.: A Method of Programming Measurement and Estimation, IBM Systems Journal, vol.1, pp.54-73, 1977