

## 協調フィルタリングを用いたプロジェクトコスト超過の予測

本村 拓也<sup>†</sup> 柿元 健<sup>†</sup> 角田 雅照<sup>†</sup> 大杉 直樹<sup>†</sup> 門田 暁人<sup>†</sup> 松本 健一<sup>†</sup>

<sup>†</sup>奈良先端科学技術大学院大学情報科学研究科 〒630-0192 奈良県生駒市高山町 8916-5

E-mail: †{takuy-mo, takesi-k, masate-t, naoki-o, akito-m, matumoto}@is.naist.jp

**あらまし** 本稿では、ソフトウェア開発コストの超過を、開発プロジェクトの初期段階に予測することを目的とする。その方法として、コスト超過に影響する多数の要因(リスク項目)の評価値を説明変数とし、協調フィルタリングによって超過コストの大きなプロジェクトを判別する。45 件のプロジェクトのデータを用いた評価実験の結果、ユーザベース協調フィルタリングによる判別精度は適合率 73%, 再現率 100%となり、従来の線形判別分析と比較して、それぞれ 11%, 27%高くなった。

**キーワード** リスク管理, リスク評価, リスク検出, 判別分析

## Prediction of Project Cost Overrun Based on Collaborative Filtering

Takuya MOTOMURA<sup>†</sup> Takeshi KAKIMOTO<sup>†</sup> Masateru TSUNODA<sup>†</sup>

Naoki OHSUGI<sup>†</sup> Akito MONDEN<sup>†</sup> and Ken'ichi MATSUMOTO<sup>†</sup>

<sup>†</sup>Graduate School of Information Science, Nara Institute of Science and Technology  
8916-5 Takayama, Ikoma, Nara, 630-0192 Japan

E-mail: †{ takuy-mo, takesi-k, masate-t, naoki-o, akito-m, matumoto }@is.naist.jp

**Abstract** In this paper, we evaluate estimation methods of software development cost overrun in an early development phase. To discriminate cost overrun projects, we used collaborative filtering technique whose predictor variables are the evaluation values of risk factors which influence the excess of cost. Data of 45 projects were used in an evaluation experiment. The result showed that precision and recall of user-based collaborative filtering was higher than liner discriminant analysis.

**Keyword** risk management, risk rating, risk detection, discrimination analysis

### 1. まえがき

ソフトウェア開発企業にとって、開発コストの超過(コスト超過)による損益は死活問題である。1つのプロジェクトにおける損益のために、他のプロジェクトで得られた利潤が全て失われるケースもある。2002 年度に全米で実施されたソフトウェア開発プロジェクトを対象にした Standish Group の調査によると、プロジェクト実施コストに対する超過コストの割合は平均 43%であった[11]。つまり、大規模プロジェクトにコスト超過が発生すれば、小規模プロジェクトで得た利潤は容易に失われてしまうことになる。

コスト超過を回避するためには、開発プロジェクトの初期段階でコスト超過を正確に予測することが重要となる[6]。予測が正確であれば、人員を再配置する、あるいは、開発対象のソフトウェアの要求を見直すなど、何らかの対策を効率よく講じることができる。また、対策を講じる時期が早期であれば、多くの選択肢が残されているだけでなく、その効果も大きい。

一般に、コスト超過を予測するための根拠として、定量化したリスク要因を用いる[7]。リスク要因とは、「機能要求の明確さ」や「顧客との関係」など開発コストに影響を与える要因を洗い出して列挙したものである。各要因は、プロジェクト管理者などが定性的に一定段階の数値などで評価して定量

化する。定量化したリスク要因と、コスト超過が発生したか否かをデータベースに蓄積し、将来のプロジェクトでコスト超過を予測するための根拠として用いる。

しかし、リスク要因には数値が記録されていない穴あきになった値(欠損値)が多く含まれるため、予測の正確さ(精度)が著しく低下する可能性が高い。リスク要因は人間の手作業により定量化されるため、様々な理由で欠損値が混入するからである。例えば、組織内で統一的な取り決めがない場合、部署毎に洗い出したリスク要因が異なってしまう。あるいは、従来の取り決めを見直して新たなリスク要因を追加した場合、過去のプロジェクトについては、新たに追加された要因全てが欠損値になってしまう。もしくは、リスクを評価する担当者が一部のリスク要因を評価し忘れるかもしれない。

そこで本稿では、協調フィルタリング(以降 CF と略記する)[1][5][8][9]を用いて、開発プロジェクトのコスト超過を予測する方法を提案する。CF は予測の根拠となるデータに、欠損値が多く含まれる場合でも、ある程度高い精度で予測できるという特長があり、これまで情報検索の分野で盛んに研究されてきた。例えば amazon.com では、読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、CF によってユーザが好むと思われる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

	説明変数 (リスク項目)							目的変数 (コスト超過評価値)	
	$m_1$	$m_2$	...	$m_j$	...	$m_b$	...	$m_n$	$C$
$p_1$	$r_{11}$	$r_{12}$	...	$r_{1j}$	...	$r_{1b}$	...	$r_{1n}$	$c_1$
$p_2$	$r_{21}$	$r_{22}$	...	$r_{2j}$	...	$r_{2b}$	...	$r_{2n}$	$c_2$
...	...	...	...	...	...	...	...	...	...
$p_i$	$r_{i1}$	$r_{i2}$	...	$r_{ij}$	...	$r_{ib}$	...	$r_{in}$	$c_i$
...	...	...	...	...	...	...	...	...	...
$p_a$	$r_{a1}$	$r_{a2}$	...	$r_{aj}$	...	$r_{ab}$	...	$r_{an}$	$c_a$
...	...	...	...	...	...	...	...	...	...
$p_m$	$r_{m1}$	$r_{m2}$	...	$r_{mj}$	...	$r_{mb}$	...	$r_{mn}$	$c_m$

図 1. 予測に用いる行列

いる全書籍の1%に満たない(即ち、残り99%以上が欠損値となる)場合でも、ある程度高い精度で予測できる[14]。CFをコスト超過予測に適用することで、多くのリスク要因が記録されていない場合でも予測が可能になると期待されるが、その具体的方法は提案されていない。

提案方法では2種類の方法で予測を試みる。第1の方法では、まず、予測対象(実施中)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、リスク項目の類似性に基づいて計算する。次に、予測対象プロジェクトのコスト超過を類似するプロジェクトから類推する。第2の方法では、まず、コスト超過が発生するか否かと各リスク項目との間の類似度を、相関係数に基づいて計算する。次に、類似するリスク項目が、予測対象プロジェクトでどのような値かに基づいて、コスト超過が発生するか否かを予測する。いずれの方法でも、欠損値の部分は計算に使わず、非欠損値を可能な限り用いて類似度計算を行う。従来方法のように、欠損値を何らかの値で埋めたり、欠損値を含むプロジェクトやリスク要因を削ったりしなくて良いため、予測精度を改善できると考えられる[14]。

以降、2章でCFの予測手順を説明する。3章で提案方法の予測手順を説明する。4章で提案方法の有効性を評価するためのケーススタディについて報告する。5章でケーススタディの結果と考察を述べる。6章で関連研究を紹介し、最後に、7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

協調フィルタリング(CF)は、非常に多くのアイテム(書籍、映画、音楽など)の中からユーザの好みに合うと思われるアイテムを予測し、推薦するシステムに用いられてきた。CFを用いたシステムは、推薦手順に応じてユーザベース手法とアイテムベース手法の2つに分類できる。

ユーザベース手法はResnickら[8]によって提案された手法である。次の手順で推薦が行われる。

### us1. データの収集

各ユーザが、使用したことがある各アイテムに対して、好みの度合いを数段階の数値(例えば、5段階評価)で評価する。

### us2. 類似度計算

収集したデータに含まれる推薦対象のユーザ(対象ユーザ)とその他のユーザとの間の類似度を算出し、評価の傾向が対象ユーザと似ているユーザ(類似ユーザ)を探し出す。

### us3. 予測値計算

対象ユーザがまだ評価していないアイテム全てに対する評価を、類似ユーザが与えた評価に基づいて予測する。

### us4. 推薦作業

対象ユーザが高い評価を与えるであろうと予測されたアイテムを推薦する。

アイテムベース手法はSarwarらによって提案された手法である[9]。次の手順で推薦が行われる。

### is1. データの収集

us1と同様の処理を行う。

### is2. 類似度計算

収集したデータに含まれる推薦対象のユーザ(対象ユーザ)とその他のユーザとの間の類似度を算出し、評価の傾向が対象ユーザと似ているユーザ(類似ユーザ)を探し出す。

### is3. 予測値計算

対象ユーザがまだ評価していないアイテム全てに対する評価を、類似ユーザが与えた評価に基づいて予測する。

### is4. 推薦作業

us4と同様の処理を行う。

### 3. 提案方法

提案方法では、図 1 に示す  $m$  行  $n$  列の表データを予測の根拠として用いる。図 1 で、 $p_i \in \{p_1, p_2, \dots, p_m\}$  は  $i$  番目の開発プロジェクトを表し、 $m_j \in \{m_1, m_2, \dots, m_n\}$  は  $j$  番目のリスク項目を表す。表中の各セル  $r_{ij} \in \{r_{11}, r_{12}, \dots, r_{mn}\}$  は、プロジェクト  $p_i$  のリスク項目  $m_j$  に対する評価値が記録される。評価値  $r_{ij}$  が未評価の場合、当該セルは欠損値となる。さらに、 $c_i \in \{c_1, c_2, \dots, c_m\}$  はプロジェクト  $p_i$  のコスト超過の有無を表す評価値（コスト評価値）が記録される。プロジェクト  $p_i$  の実施中に許容できない程度に大きなコスト超過が発生した場合は  $c_i$  として 1 が、発生しなかった場合は 0 が記録される。以降、ユーザベース手法による予測手順と、アイテムベース手法による予測手順について説明する。

#### ユーザベース手法による予測手順

1. 予測対象プロジェクト  $p_a$  とその他のプロジェクト  $p_i$  との間の類似度  $sim(p_a, p_i)$  を次式で計算する。

$$sim(p_a, p_i) = \frac{\sum_{j \in M_a \cap M_i} (w_j r_{aj} - \bar{m}_j)(w_j r_{ij} - \bar{m}_j)}{\sqrt{\sum_{j \in M_a \cap M_i} (w_j r_{aj} - \bar{m}_j)^2} \sqrt{\sum_{j \in M_a \cap M_i} (w_j r_{ij} - \bar{m}_j)^2}} \quad (1)$$

ただし、 $M_a$  と  $M_i$  は各々、プロジェクト  $p_a$  と  $p_i$  で評価されたリスク項目の集合を現す。 $\bar{m}_j$  は、リスク項目  $m_j$  に対する評価値の中央値を表す。 $w_j$  はリスク項目  $m_j$  に対する重みであり、次式で計算する。

$$w_j = \frac{\sum_{i \in P_a \cap P_j} (r_{ib} \times r_{ij})}{\sqrt{\sum_{i \in P_a \cap P_j} (r_{ib})^2} \sqrt{\sum_{i \in P_a \cap P_j} (r_{ij})^2}} \quad (2)$$

ただし、 $P_a$  と  $P_j$  は各々、リスク項目  $m_a$  と  $m_j$  を評価したプロジェクトの集合を表す。

2. 予測対象プロジェクト  $p_a$  に対するコスト評価値  $c_a$  の予測値  $\hat{c}_a$  は、式(1)によって計算した類似度を用い、次式(3)で計算する。計算結果に対してひとつの閾値を設定し、予測値  $\hat{c}_a$  が閾値より大きければ 1 (許容できないコスト超過が発生する)、小さければ 0 (許容できないコスト超過は発生しない)と予測する。

$$\hat{c}_a = \frac{\sum_{j \in k\text{-nearest Projects}} c_j \times sim(p_a, p_i)}{\sum_{j \in k\text{-nearest Projects}} sim(p_a, p_i)} \quad (3)$$

ただし、 $k\text{-nearestProjects}$  は、予測対象プロジェクト  $p_a$  との類似度が高い上位  $k$  件の類似プロジェクトの集合を表す。類似プロジェクト数  $k$  の値は予測精度に影響を与える。4 章で説明するケーススタディでは、最も精度が高

くなる  $k$  の値を採用した。

#### アイテムベース手法による予測手順

1. 対象プロジェクト  $P_a$  についてコスト評価値  $c_a$  を予測するため、コスト超過の有無を表す評価項目（予測対象項目） $C$  と各リスク項目  $m_j$  との間の類似度を次式で計算する。

$$sim(C, m_j) = \frac{\sum_{i \in P_b \cap P_j} (r_{ib} - \bar{m}_i) \times (r_{ij} - \bar{C})}{\sqrt{\sum_{i \in P_b \cap P_j} (r_{ib} - \bar{m}_i)^2} \sqrt{\sum_{i \in P_b \cap P_j} (r_{ij} - \bar{C})^2}} \quad (4)$$

ただし、 $P_a$  と  $P_j$  は各々、リスク項目  $m_a$  と  $m_j$  を評価したプロジェクトの集合を表す。また  $\bar{m}_i$  は、リスク項目  $m_i$  に対する評価の平均値を表す。

2. 予測対象プロジェクト  $p_a$  に対するコスト評価値  $c_a$  の予測値  $\hat{c}_a$  は、式(4)によって計算した類似度を用いて次式で計算する。計算結果に対してひとつの閾値を設定し、予測値  $\hat{c}_a$  が閾値より大きければ 1 (許容できないコスト超過が発生する)、小さければ 0 (許容できないコスト超過は発生しない)と予測する。

$$\hat{c}_a = \frac{\sum_{j \in k\text{-nearestRisks}} (m_{ij} - \bar{p}_i) \times sim(C, m_j)}{\sum_{j \in k\text{-nearestRisks}} sim(C, m_j)} + \bar{p}_a \quad (5)$$

ここで、 $k\text{-nearestRisks}$  は、予測対象項目  $C$  と類似度が高い上位  $k$  件のリスク項目の集合を表す。類似リスク項目数  $k$  の値は予測精度に影響を与える。4 章で説明するケーススタディでは最も精度が高くなる  $k$  の値を採用した。 $\bar{p}_i$  はプロジェクト  $p_i$  に対する評価の平均値を表す。

## 4. ケーススタディ

### 4.1. 目的とアプローチ

ケーススタディの目的は、多くの欠損値を含むデータに対する提案方法の有効性を評価することである。そのためのアプローチとして、実際のソフトウェア開発プロジェクトに対するリスク項目の評価データを用い、提案方法と代表的な従来方法（判別分析）の予測精度を評価した。ただし、判別分析に関してはデータに欠損値が含まれる場合に予測ができないため、欠損値処理法（平均値挿入法）を適用し、その上でステップワイズ法による変数選択を行った。精度の評価指標は適合率、再現率、F1 値を用いた。相互検証法（ホールドアウト法）によって評価を行い、各手法の精度を評価、比較した。

表1. リスク項目の分類と各分類の欠損率

	基本	提案時 要求関係	提案時 見積関係	受注時 要求関係	受注時 見積関係	受注時 契約関係	全項目
リスク 項目数	44	16	49	16	49	25	199
最大 欠損率	0.00%	91.11%	93.33%	84.44%	86.67%	97.78%	97.78%
最小 欠損率	0.00%	35.56%	33.33%	6.67%	6.67%	6.67%	0.00%
平均 欠損率	0.00%	52.32%	63.61%	30.61%	45.57%	43.70%	42.28%
欠損率 標準偏差	0.00%	24.49%	26.44%	33.27%	35.57%	37.87%	35.83%

#### 4.2. 実験データ

ケーススタディでは、日本のあるソフトウェア開発企業で実施されたソフトウェア開発プロジェクトに対するリスク項目の評価データ(実験データ)を用いた。実験データに含まれるプロジェクトは45件、リスク項目は199項目であった。各プロジェクトに対して、十分訓練を積んだプロジェクト管理者により、各リスク項目が0(コスト超過に全く影響を与えない)から3(コスト超過に大きな悪影響を与える)の4段階で評価された。さらに、各プロジェクトについて、予測対象項目Cの値が0(許容できないコスト超過は発生しない)、あるいは、1(許容できないコスト超過が発生する)として記録された。

表1に、リスク項目の分類と各分類の欠損率(全プロジェクト中、当該リスク項目が欠損値であるプロジェクトの割合)を示す。リスク項目数が多いため、表中の各列に記した規準に基づいて項目を分類し、各分類について基本統計量を示した。分類の規準は次の通りである。

- **基本**: プロジェクト開始時に評価可能な基本的項目。
- **提案時要求関係**: プロジェクトで開発するソフトウェアの要求に関する項目。顧客に対して案件を提案した時点での評価値。
- **提案時見積関係**: プロジェクト完遂に要する期間、人員などに関する項目。顧客に対して案件を提案した時点での評価値。
- **受注時要求関係**: プロジェクトで開発するソフトウェアの要求に関する項目。顧客と契約を締結した時点での評価値。
- **受注時見積関係**: プロジェクト完遂に要する期間、人員などに関する項目。顧客と契約を締結した時点での評価値。
- **受注時契約関係**: 顧客との契約に関するリスク項目。顧客と契約を締結した時点での評価値。顧客と契約を締結した時点での評価値。

実験データ全体の欠損率は平均42.28%と大きく、その分布にも大きな偏りがある。基本分類に含まれる全項目には欠損値が存在しない。一方で、その他の分類に含まれるリスク項目の平均値は30%から60%と様々である。また、受注時要求関係や受注時見積もり関係に含まれる項目の最小欠損値は10%未満である。一方で、提案時見積関係や受注氏契約関係などに含まれる項目の最大欠損率は90%を超える。欠損率や、その分布の偏りが大きな理由として、プロジェクトが実施された時期や部署が様々であることが挙げられる。実施時期や部署が異なる場合、評価されるリスク項目が異なり、欠損値が生じ易い。また、実際の開発現場では時間的制約などに起因し、リスク項目を評価し忘れるなど、人的要因によって欠損値が生じることもあった[2][10]。

#### 4.3. 評価基準

ケーススタディでは、予測精度の評価基準として適合率、再現率、F1値を用いた。これらの評価基準は、CFを用いたシステムの精度評価基準として広く用いられている[3]。適合率と再現率を求める際は、次の2つの集合に着目する。

$A$ : コスト評価値の予測値  $\hat{c}_i$  が1と計算されたプロジェクトの集合。

$R$ : 実際のコスト評価値  $c_i$  が1だったプロジェクトの集合。

##### 適合率

集合  $A$  に含まれる要素の内、集合  $R$  に含まれる要素と合致した割合。即ち、予測値が1のときに実際の値も本当に1だった割合。適合率(precision)は次式で計算する。

$$precision = \frac{|A \cap R|}{|R|} \quad (6)$$

##### 再現率

集合  $R$  に含まれる要素の内、集合  $A$  に含まれる要素と合致した割合。即ち、実際の値が1のときに漏れなく1と予測できた割合。再現率(recall)は次式で計算する。

$$recall = \frac{|A \cap R|}{|A|} \quad (7)$$

### F1値

適合率と再現率のバランスの良さを表す値。適合率と再現率はトレードオフの関係である。予測値  $\hat{c}_a$  に対する閾値を変化させることで、両者は様々な値に変化する。極端な場合、閾値を 0 とすれば、全プロジェクトが集合  $A$  に含まれ、適合率は著しく低下するものの、再現率は 100%となる。F1値は次式で計算する。

$$F1 = \frac{(1+b^2) \times precision \times recall}{b \times (precision + recall)} \quad (8)$$

ただし、式(8)中の  $b$  は適合率、再現率のどちらを重視するかを指定するためのパラメータである。本ケーススタディでは  $b = 1$  とし、どちらか一方を重視することなく、平等にバランスを取ることにした。

ケーススタディでは、閾値を0から1の間で0.1ずつ変化させ、F1値が最も高くなる時の適合率と再現率を当該手法の精度として算出した。

### 4.4. 実験手順

ユーザベース手法の CF、アイテムベース手法の CF、判別分析[12]、各々を用いて予測を行った。精度評価手順は次のとおり。

1. 実験データを無作為に2等分し、一方を予測の根拠とするフィットデータ、もう一方を予測対象とするテストデータとした。
2. 予測値の計算に用いる類似要素数(3章で説明した  $k$ -nearestProjects 及び  $k$ -nearestRisks の  $k$ )を決定し、ユーザベース手法の CF、アイテムベース手法の CF を用いてコスト評価値の予測値  $\hat{c}_a$  を算出した。
3. 算出した予測値  $\hat{c}_a$  に対して、閾値を 0 から 1 まで 0.1 ずつ変化させ、評価基準を計算するために必要な集合  $A$  を決定した。
4. 実際のコスト評価値  $c_a$  に基づき、評価基準を計算するために必要な集合  $R$  を決定した。
5. 集合  $A$  と  $R$  から、各評価基準を計算し、比較した。

### 5. 結果と考察

ケーススタディの結果得られた評価基準の値を表 2 に示す。表中、各行に各予測方法を、各列に各評価基準を記した。表の各セルには、各手法を用いた場合の評価基準の値を記した。いずれの評価基準も、値が高いほど精度が高いことを示す。

ケーススタディの結果から、ユーザベース手法の CF を用

表2. ケーススタディの結果:各評価基準の値

	適合率	再現率	F1 値
ユーザベース CF	0.73	1.00	0.84
アイテムベース CF	0.58	0.64	0.61
判別分析	0.62	0.73	0.67

いた予測の精度が最も高いことがわかる。ユーザベース手法の CF を用いた場合、予測対象プロジェクトのコスト超過有無は、類似プロジェクトから類推される。即ち、ケーススタディで用いたデータに関して、リスク項目に対する評価の傾向が似たプロジェクトは、コスト超過の有無に関しても似た傾向を示した可能性が高いと考えられる。データ中に多くの欠損値が含まれる場合にも比較的高い精度で予測できるという CF の性質も、判別分析と比較して予測精度が高い理由の一つと考えられる。以上のことから、本ケーススタディで用いたデータを用いた場合、ユーザベース手法の CF は、プロジェクトコスト超過予測にある程度有効であると考えられる。

アイテムベース手法の CF は、ケーススタディで用いた方法の中で最も精度が低い。コスト超過の有無と、単独で高い相関があるリスク項目が極めて少なかった可能性が高い。一方で判別分析の精度がアイテムベース手法の CF よりも高いことから、コスト超過の有無と、その他のリスク項目の間にある程度の重相関関係があることが考えられる。

### 6. 関連研究

水野らは、過去の開発プロジェクトから収集された特性値からベイジアンネットを用いた混乱予測モデルを作成し、混乱状態(プロジェクトの進捗状況の把握が困難となった状態)に陥る可能性のあるプロジェクトを予測することを提案している[4]。

ベイジアンネットを用いた予測では、予測のためのモデル作成が必要である。予測者が立てた仮定を用いてあらかじめ作成した特定の予測モデルを基に、予測が行われる。予測モデルが適切であれば精度は高くなるが、入力となる予測対象のデータセット毎の細かな特徴を考慮した予測は難しい。CFによる予測では、あらかじめ予測モデルを作成するといった事前の作業を必要としなくとも高い精度で予測可能である。

### 7. むすび

本稿では、「あるリスク項目に対して同様の評価値が与えられているプロジェクト同士は、コストの超過に関しても同様の結果を示すであろう」という仮定のもと、協調フィルタリング(CF)を用いてプロジェクトのコスト超過の予測を行い、その予測精度を評価した。実験の結果、CF のユーザベース手法による予測では、判別分析を用いた予測よりも F1 値で 0.17 高いという結果が得られた。

今後の課題として、プロジェクトのフェーズの推移に伴っ

て予測精度がどのように変化するかを明らかにし、予測を行うための最適なフェーズを特定することが考えられる。

また、CF における類似度計算の妥当性を向上させ予測精度を高めるために、予測の前作業としてコストの評価値への影響が低いリスク項目を特定し、特定した項目を除去した上で予測を行うことを考えている。

予測用データに含まれるプロジェクトの件数を増やし、より多数、かつ、多様なプロジェクトが含まれるデータセットを用いての予測実験と評価を行うことも今後の課題である。

## 謝 辞

本研究を進めるにあたり、貴重なご意見を頂き、データをご提供頂いた、株式会社 日立製作所 情報・通信グループ 生産技術本部 福地豊氏、米光哲哉氏、及び、同 プロジェクトマネジメント本部 星幸雄氏に感謝致します。

## 参 考 文 献

- [1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," *Comm. of the ACM*, Vol.35, No.12, pp.61-70, 1992.
- [2] 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.
- [3] J. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. on Information Systems*, Vol.22, No.1, pp.5-53, 2004.
- [4] 水野修, 天壽聡介, 山之内太, 菊野亨, 高木徳生, "混乱プロジェクトの予測へのベイジアンネットの適用," *ソフトウェアシンポジウム 2003 論文集*, pp.193-199, July 2003.
- [5] N. Ohsugi, M. Tsunoda, A. Monden, and K. Matsumoto, "Applying Collaborative Filtering for Effort Estimation with Process Metrics," *Proc. of the 5th Int'l Conf. on Product Focused Soft. Process Improvement*, Springer, Berlin Heidelberg, pp.274-286, 2004.
- [6] J. D. Procaccion, J. M. Verner, S. P. Overmyer, and M. E. Darter, "Case study : factors for early prediction of software development success," *information and software technology*, Vol.44, No.1, pp.53-62, 2002.
- [7] Project Management Institute, *A Guide to the Project Management Body of Knowledge 2000 Edition*, Project Management Institute, 2000.
- [8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proc. ACM Conf. on Computer Supported Cooperative Work*, pp.175-186, Chapel Hill, North Carolina, U.S.A, Oct 1994.
- [9] B. M. Sarwar, G. Karypis, J. A. Konstan, and J.T. Riedl, "Item-Based Collaborative Filtering

Recommendation Algorithms," *Proc. 10th International World Wide Web Conference*, pp.285-295, Hong Kong, May 2001.

- [10] K. Srinivasan, and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort," *IEEE Trans. on Software Eng.*, Vol.21, No.2, pp.126-137, 1995.
- [11] The Standish Group International, Inc., "2003 CHAOS Chronicles Report," 2003.
- [12] 田中豊, 垂水共之, *統計解析ハンドブック 多変量解析*, 共立出版, 東京, 1998.
- [13] 外山久, 南野猛, "「リスク早期抽出自己診断表」の開発と応用," *プロジェクトマネジメント学会誌*, Vol.4, No.4, Aug 2002.
- [14] 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.