欠損率の高いプロジェクトデータを利用したプロジェクトの成否予測

出張純也[†1]　菊野亨[†1]
菊地奈穂美[†2]　平山雅之[†3]

ソフトウェア開発プロジェクトから収集したデータを利用して，品質やコストなどを予測する研究が多く行われている．本研究では，通常のプロジェクトから収集される欠損の多いデータを利用して，プロジェクトの成否の予測を試みる．

欠損率が高いので，2 段階の方法を提案する．最初に，未記入項目の多いメトリクスを削除し，次に予測に影響を与えると考えられるメトリクスだけに絞り込む．メトリクスの絞り込みには相関ルールマイニングを適用する．

適用実験として，IPA/SEC のデータ白書として公開されているプロジェクトデータを利用して，プロジェクトの成否を設計工程の終了時に予測した．まず，設計工程終了時点では未だ値が定まらないメトリクスを削除した．その時点でのデータの欠損率は 43.8%になった．提案法を適用した結果，メトリクスを 7 個にまで絞り込みを行って，予測精度 82.8% が達成できた．

On Prediction of Project Success Using Incomplete Project Data

Junya Debari ,[†1] Tohru Kikuno ,[†1] Nahomi Kikuchi [†2]
and Masayuki Hirayama[†3]

Many researches tried to predict quality and cost using project data set. Note that project data set is usually assured to be complete in the sense that all metrics data is filled out. But actually we are facing with public project data set which contain many incomplete data. In this paper we try to predict, after design phase, if a project will finish successfully or not based on such a public project data set.

We propose two phases of refinements upon data set: (1) reduction of incomplete data and (2)extraction of meaningful metrics. The first reduction is just deletion of such metrics that contain many missing data. We then apply association rule mining for metrics extraction. For prediction of a project,we employ Bayesian Classifier as usual.

We conducted an experimental evaluation on IPA/SEC data set which is collected from Japanese companies. The IPA/SEC data set consists of 237 projects and 69 metrics,and contains 43.8% of missing data. By applying the proposed method, 82.8% of accuracy was finally realized with only 7 metrics.

1. はじめに

ソフトウェア開発プロジェクトから収集したプロジェクトデータを対象として，品質，コスト，成否などを予測する研究が多く行われてきている．これまでのプロジェクトデータは周到に計画・設計されたプロジェクトから収集されたデータ (ケース 1) か，あるいはよく整備された開発組織におけるプロジェクトから収集されたデータ (ケース 2) が利用されていた．

しかし現実には，こうした条件を満たさない多くのプロジェクトでも様々なメトリクスデータが収集されている．例えば，International Software Benchmarking Standards Group (ISBSG) や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 (IPA/SEC) がソフトウェア開発プロジェクトのデータを収集している [6, 16]．こうしたデータには欠損が多く含まれている．IPA/SEC は 2008 年の時点で国内の企業 20 社からソフトウェア開発プロジェクトのデータを収集している．本研究では，こうしたデータをケース 3 と呼び，その活用について一つの提案を行う．ケース 1，2 とケース 3 の違いはデータの欠損率に特徴的に現れる．ケース 1 での欠損率は 0%，ケース 2 では 10% 未満であるのに対して，ケース 3 では 30%以上になる．

ケース 1 やケース 2 の研究では，予測精度を向上させるためにメトリクスの絞り込みを行うものが多く存在する．本研究においても，予測精度の向上のためにメトリクスの絞り込みを行う．ただし，ケース 3 では欠損率が高いためケース 1 やケース 2 の方法とは異なるアプローチをとる．提案法の主な特徴は次の 3 つである．

(1) まず未記入項目の多いデータを削除して欠損率をある程度改善させておいて，メトリクスの絞り込みを行う．
(2) 未記入項目の多いデータの削除は，未記入項目の多い順番に機械的に行う．
(3) メトリクスの絞り込みには，欠損のあるデータにも適用可能な相関ルールマイニング法を適用する．

†1 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
†2 沖電気工業株式会社
Oki Electric Industry Co., Ltd
†3 日本大学 大学院理工学研究科
Graduate School of Science and Technology, Nihon University

本研究では IPA/SEC のデータ白書として公開されているプロジェクトデータを利用して，プロジェクトの成否を設計工程の終了時に予測することを試みる．そのため，提案法のフェーズ 1 として，設計工程が終了した段階で未だ入手が不可能なメトリクスを削除した．こうして求まったプロジェクトデータの欠損率は 43.8 %になっていた．適用実験の結果，未記入項目データの削除の操作とメトリクスの絞り込みを行う事によって，予測精度は最も高い場合で 7 つのメトリクスによって 82.8%が達成できることを確認した．

2. 従来の研究

プロジェクトデータを利用して種々の予測を行う研究は，概ね，次の 2 種類に分類することができる．

(1) 可能な限り注意深く設計された開発プロジェクトから，事前に選定されたメトリクスのデータを収集して，それらを用いた科学的，あるいは統計的な分析を行う．データには欠損がほとんどない．

(2) 特に設計されたわけではない，通常の開発プロジェクトから事前に選定されたメトリクスのデータを収集して，それらに種々の統計的な分析を行う．

文献 [3, 7, 11] の研究は，欠損のないプロジェクトデータを対象として行われた研究である．つまり典型的なケース 1 の研究である．文献 [11] では，複数のデータセットに対して，類似度に基づいた工数見積を適用した．その結果，全てのデータセットにおいて高い精度で見積りが可能であることがわかった．文献 [3] では，COCOMO 法でコストを見積る際に，見積りに役に立たないメトリクスを削除することで，コスト見積り精度が上昇することが明らかにされた．文献 [7] は，CoBRA 法に基づいて品質予測を行っているが，品質予測のためのパラメータ決定を行い，予測モデルを構築している．この手法では，予測に用いる特徴の決定を専門家の知見と統計手法に基づいて行っている．

一方，文献 [1, 13, 18] の研究は，少量の欠損を含むプロジェクトデータを対象として行われた研究で，ケース 2 に属している．文献 [13] では，プロジェクトマネージャからのアンケート結果に対してロジスティック回帰分析を適用し，失敗プロジェクトの予測を試みている．この研究では，欠損 (アンケートに回答がないこと) を潜在的なリスク要因であると考えて欠損を補完している．文献 [1] では，ベイズ識別器をソフトウェア開発データに適用して，ソフトウェアプロジェクトの最終状態を予測している．ベイズ識別器は欠損のあるデータに対しても適用可能な手法であるため，欠損の削除や補完は行っていない．文献 [18] では開発の現場から得られたアンケートの回答に対して相関ルールマイニングを適用して，ソフトウェアプロジェクトが混乱するリスク要因の特定を行っている．

欠損を扱う方法についての研究としては文献 [2, 8, 12, 14] がある．文献 [12] では，無欠損のデータに人工的に欠損を発生させて，欠損データの削除法，欠損データの補完法について評価実験を行っている．その結果，欠損が少ない場合には欠損データを削除するのが最も効率が良いと結論している．文献 [2] では，工数見積りに使うデータについて，K 近傍法と中央値法を比較し，K 近傍法で補完する場合に精度が最も高くなることを明らかにした．文献 [14] では，複数の欠損補完法を比較した結果，多重補完法が最も優れていると報告している．

上述の研究以外に特徴のある研究として文献 [17, 19] がある．文献 [17] では協調フィルタリングを用いてソフトウェア開発工数を予測している．60%の欠損があるデータに対して協調フィルタリングを適用することで，欠損の除去や補完をした上で重回帰分析を行う場合よりも高い精度が達成できることを明らかにした．しかし，[17] では全てのメトリクスを用いた分析を行っており，変数の選択を今後の課題としている．文献 [19] ではプロジェクトの類似性に基づいて工数を予測している．この研究では，ISBSG のデータなどを利用しているが，ユークリッド距離に基づいて類似性を計算しているため，欠損のないデータを作成して利用している．欠損のあるデータを利用する場合については述べられていない．

3. 本研究での予測問題

3.1 定　　義

本研究では IPA/SEC のデータ白書として公開されているプロジェクトデータ [16] を利用して，プロジェクトの成否を設計工程の終了時に予測することを試みる．

このデータは国内の企業 20 社から収集されたものである．2007 年までに終了したエンタープライズ系のソフトウェア開発プロジェクトのうち，1397 プロジェクト[⋆1]から 633 種類のメトリクスを収集している．データの欠損率は 83.8%となっている．ここで，プロジェクトの成否の判断にはメトリクス「実績の評価 (品質)」を利用する．このメトリクスの値は，稼働後 6ヶ月の不具合数の実績値が計画値に対してどの程度大きかったかを示す．a, b, c, d, e の 5 種類の値をとり，a が「稼働後不具合数が計画値より 20%以上少ない」，b が「稼働後不具合数が計画値以下であり，そのずれが 20%未満である」，c が「稼働後不具合数が

⋆1 実際には 2056 件のプロジェクトから収集されているが，IPA/SEC やデータ提出企業が「信頼性が低い」と判断しているプロジェクトのデータについては削除している．

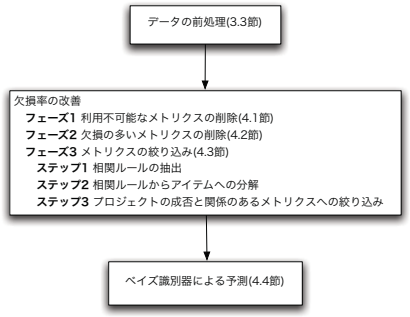

図 1　提案法の構成

計画値を超過している．超過は 50%以内である」，d が「稼働後不具合数が計画値を超過している．超過は 50%より大きく，100%以内である」，e が「稼働後不具合数が計画値を超過している．超過は 100%より大きい」という意味である．a，b を稼働後不具合数が計画値よりも少なかったため「成功」，c，d，e を稼働後不具合数が計画値よりも多かったため「失敗」と考える．

3.2　解決の指針

データに欠損が多すぎる場合には分析の結果が不正確になる可能性があるため，欠損のあるデータを削除する．しかし，データに欠損が多いため，データの欠損を全て削除することができない．そのため，本研究では，2 段階で (1) 機械的に欠損の多いメトリクスを削除する，(2) プロジェクトの成否と関係のあるメトリクスだけに絞り込みを行う，という方法を採用する．この提案法の概要を図 1 に示す．

図 2 は M 種類のメトリクス，N 件のプロジェクトからなるデータのイメージ図である．3.1 節の IPA/SEC データの場合，$M = 633$，$N = 1397$ となる．x とマークしてある箇所はその値が欠損しているとする．$j(1 \leq j \leq M)$ 番目のメトリクスに着目すると，6 個の欠損がある．$i(1 \leq i \leq N)$ 番目のプロジェクトには 9 個の欠損がある．このような欠損を削除するには (1) 欠損の多いメトリクスを削除する，(2) 欠損の多いプロジェクトを削除する，の 2 種類の方法が考えられる．本研究では，(1) の方法を採用する．これは，分析に利用するプロジェクト数が少なくなると統計的な分析が正確に行えない可能性があるからである．

メトリクスの絞り込みの方法としては，ロジスティック回帰分析や相関ルールマイニングなどが考えられる．ロジスティック回帰分析は欠損のないデータへの適用を前提としているため，本研究では利用せず，欠損のあるデータに対しても適用可能な相関ルールマイニングを利用する．

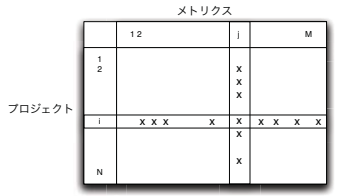

図 2　データ欠損の説明図

プロジェクトの予測に関しても，欠損のあるデータに対して利用可能な方法を利用する必要がある．そのため，本研究ではベイズ識別器を利用する．

3.3　前　処　理

IPA/SEC のデータに収集されているメトリクスには，開発プロジェクトの特徴を示すメトリクスと，規模・工期・工数・信頼性などの実績を示すメトリクスが含まれているが，生産性を示すメトリクスが含まれていない．そのため，生産性に関するメトリクスは，含まれているメトリクスをもとに計算して追加する必要がある．例えば,「月あたりの工数」,「月あたりの SLOC」,「月あたりの FP」などである．ここでは,「月あたりの SLOC」＝「SLOC 実績値」÷「実績月数_プロジェクト全体」[⋆1] とする．

また，相関ルールマイニングおよびベイズ識別器は連続値を取り扱うことができないため，連続値のメトリクスは離散値に変換する必要がある．例えば,「実績開発工数」は中央値以上の場合に”High”，中央値未満の場合に”Low” とした．他の連続値のメトリクスについても同様に中央値で 2 分割した．同じ前処理を行ったメトリクスとしては,「SLOC 実績値」,「実績月数_プロジェクト全体」,「月あたりの工数」,「月あたりの SLOC」,「月あたりの FP」などがある．

相関ルールマイニングでは，メトリクスの属性値ごとの件数が少ない場合には，その属性値がルールに現れないという問題がある．メトリクスの属性値の種類が多い場合には属性値ごとの件数が少なくなるため，いくつかの属性値をまとめる必要がある．例えば,「開発対象プラットフォーム」というメトリクスは「Windows95/98/Me 系」,「Windows NT/2000/XP 系」

⋆1 「SLOC 実績値」,「実績月数_プロジェクト全体」は，いずれもデータ白書に含まれているメトリクスである．

など 17 種類の属性値をとり，少ないものでは 1 件しかないものもある．そこで,「Windows クライアント」「Windows サーバ」「UNIX 系」「Linux 系」「その他」という 5 種類の属性値にまとめた．同じ前処理を行ったメトリクスとしては,「Web 技術の利用」,「オンライントランザクション処理」,「主開発言語」,「DBMS の利用」がある．

4. 提 案 手 法

提案手法は，図 1 のような構成になっている．まず，データに対する前処理 (3.3 節で述べた) を行う．次に，データセットの欠損率の向上を実現する．最後に，予測を行う．欠損率の改善は次の 3 つのフェーズからなる．

4.1 フェーズ 1(利用不可能なメトリクスの削除)

与えられるデータ D0 は M0 個のメトリクスと N0 件のプロジェクトからなるとする．本研究ではプロジェクトの設計終了段階での予測を想定している．プロジェクトの設計が終了した段階で未だ入手が不可能なメトリクスを利用不可能なメトリクスと呼び，それらをすべて削除する．ただし，見積り値がその時点で求まるもの (規模，工期，工数，生産性など) については削除しない．さらに，メトリクス「実績の評価 (品質)」の値が欠損しているプロジェクトをすべて削除する．この結果，N1 プロジェクト，M1 メトリクスからなるデータ D1 が準備される．

4.2 フェーズ 2(欠損の多いメトリクスの削除)

データに含まれる欠損が多いメトリクスから順番に削除していき，欠損率が $n\%$ になった時点で削除をやめる．n の値については事前に確定できないので，例えば $n = 20\%, 15\%, \cdots$ を考える．なお，欠損率が同じメトリクスが複数存在する場合は，同時に削除する．メトリクスの削除が終了した時点でのデータを D2 とすると，そのプロジェクト数は N1 に変化はなく，メトリクス数は減少して M2(M2 < M1) となる．

4.3 フェーズ 3(メトリクスの絞り込み)

D2 からプロジェクトの成否に関係のあるメトリクスに絞り込みを行う．絞り込んだ後のメトリクス数は M3 となる．プロジェクト数は N1 のまま変化しない．絞り込みを行った後のデータを D3 とする．フェーズ 3 は次のステップ 1 ～ ステップ 3 からなる．

ステップ 1 相関ルールの抽出

データ D2 に対して相関ルールマイニングを行う．相関ルールマイニングとは，相関ルールと呼ばれるデータセットを抽出するデータマイニング手法の一つである [5]．相関ルールは，$X_1 \wedge \cdots \wedge X_k \Rightarrow Y$ のような形で表現される．各 $X_i (1 \leq i \leq k)$ はアイテム (「メトリクス」=「属性値」) で表現されている．本研究では，プロジェクトの成否と関係のあるメトリクスを選択するために，Y を「実績の評価 (品質)」=「成功」に限定している．マイニングの際には，信頼度，支持度と呼ばれるパラメータを変化させてマイニングを行う．

表 1　データの例

ID	M_1	M_2	M_3	M_4	品質
1	a	m	p	x	成功
2	a	n	q	y	成功
3	a	m	p	x	成功
4	b	m	p	y	成功
5	b	m	q	x	失敗
6	b	m	q	y	失敗
7	a	m	p	x	失敗

ここでは表 1 に示す簡単な例を用いてステップ 1～ ステップ 3 の説明を行う．表 1 のデータに対して，最低信頼度 0.7，最低支持度 0.4 として相関ルールマイニングを適用すると，次の 3 つのルールが抽出される．

- R_1 :「$M_1 = a$」 $\Rightarrow$「品質=成功」
- R_2 :「$M_3 = p$」 $\Rightarrow$「品質=成功」
- R_3 :「$M_2 = m$」 $\wedge$「$M_3 = p$」 $\Rightarrow$「品質=成功」

ステップ 2 相関ルールからアイテムへの分解

抽出した相関ルールの前提部を全てアイテムに分解する．ここで得られるアイテムの集合を I とする．

例えば，ステップ 1 の例で得られた R_1，R_2，R_3 から得られる I は {「$M_1 = a$」,「$M_2 = m$」,「$M_3 = p$」} となる．

ステップ 3 プロジェクトの成否と関係のあるメトリクスへの絞り込み

最後に，目的変数との関係がより強いアイテムに絞り込む．成功プロジェクトと失敗プロジェクトについて，I に含まれる各アイテムの条件が満たされる割合を計算する．あるアイテム i について，成功プロジェクトで条件が満たされる割合を成功寄与率，失敗プロジェクトで条件が満たされる割合を失敗寄与率と呼ぶ．成功寄与率と失敗寄与率の差を寄与率とし，寄与率が大きいアイテムをプロジェクトの成否と関係の強いアイテムと考える．この成否と関連の強いアイテムに含まれるメトリクスを，プロジェクトの成否と関係のあるメトリクスとする．

例えば，ステップ 2 の例で得られた「$M_1 = a$」の成功寄与率は $3 \div 4 = 0.75$ となり，失敗寄与率は $1 \div 3 = 0.33$ となる．寄与率は $0.75 - 0.33 = 0.42$ となる．同様に計算する

と,「$M_2 = m$」の寄与率は -0.25,「$M_3 = p$」の寄与率は 0.42 となる．よって，ここでは「$M_1 = a$」と「$M_3 = p$」が成否と関連の強いアイテムであるとし，M_1 と M_3 を成否と関連の強いメトリクスとする．

4.4　ベイズ識別器による予測

D3 に対してベイズ識別器 [4] を利用して予測モデルを構築する．ベイズ識別器を用いる主な理由としては，ベイズ識別器が確率として予測結果を示すこと，欠損のあるデータに対しても適用可能であること，そして，先行研究 [1] から判断すると，ある程度の適用可能性が期待できること，が挙げられる．

5. 適 用 実 験

3.1 節で説明したように，本研究では IPA/SEC のデータ白書として公開されているプロジェクトデータ [16] を利用してプロジェクトの成否を設計工程の終了時に予測する．以下，5.1 節でデータセットが変化する様子を詳しく説明する．次に，5.2 節でその結果の分析を行う．

5.1　適 用 結 果

まず，フェーズ 1 について説明する．前述の通りプロジェクトの成否を予測するので，メトリクス「実績の評価 (品質)」に注目する．分析対象となるプロジェクトの数は 237 となった．237 プロジェクトのうち，186 プロジェクトが成功プロジェクト，51 プロジェクトが失敗プロジェクトであった．プロジェクトの設計段階で予測するので，その時点までに (見積りなども含めて) 情報が入手できないメトリクスを削除する．その結果，メトリクス数が 633 から 69 に変化した．

この 69 個のメトリクス，237 件のプロジェクトからなるデータを D1 とする．なお，D0 の欠損率は 83.8%であったが，D1 の欠損率は 43.8%であった．

次に，フェーズ 2 について説明する．欠損率が 20% 15%，10%，5%，0%となる 5 種類のデータセット D2 を作成した．その結果，残ったメトリクスの数はそれぞれ 32，27，24，22，11 となった．次に，それぞれの D2 から，学習用データと評価用データを作成した．具体的には，D2 をランダムに 4 分割し，そのうち 3 つを学習用データとし，1 つを評価用データとした．この結果，1 つの D2 から学習用データと評価用データの組み合わせが 4 組できるので，それらをデータ 1，2，3，4 と呼ぶ．このとき，学習用データと評価用データにおいて，そこに含まれるデータがすべて成功プロジェクトあるいは失敗プロジェクトとならないように注意した．

表 2　絞り込まれたメトリクスの数

	欠損 20%	欠損 15%	欠損 10%	欠損 5%	欠損 0%
データ 1	**6**	**7**	**7**	7	2
データ 2	4	7	4	4	0
データ 3	9	7	9	**6**	**2**
データ 4	6	7	5	7	2

表 3　求まった予測精度 (D3)

	欠損 20%	欠損 15%	欠損 10%	欠損 5%	欠損 0%
データ 1	**0.776**	**0.800**	**0.828**	0.700	0.783
データ 2	0.717	0.729	0.717	0.768	—
データ 3	0.738	0.700	0.738	0.763	0.780
データ 4	0.690	0.707	0.690	**0.793**	**0.793**
平均値	0.730	0.734	0.743	0.756	0.785

次に，フェーズ 3 について述べる．学習用データ 1，2，3，4 に対して，相関ルールマイニングを適用した．最低信頼度は 0.9 と設定した．最低支持度は各学習用データの成功プロジェクトの 1/3 で成立するルールが抽出されるように設定した．相関ルールマイニングには R [10] を使用した．次に，得られた相関ルールをアイテムに分解し，得られたアイテムに基づいてメトリクスを絞り込んだ．最後に，絞り込まれたメトリクスに基づいて D3 を作成した．絞り込んだ結果のメトリクスの数を表 2 にまとめた．例えば，欠損 10%の学習用データ 1，2，3，4 からそれぞれ 63，9，101，43 個のルールが抽出された．ルールを分解して得られたアイテムはそれぞれ 17，12，16，14 個であった．最終的に絞り込まれたメトリクスの数はそれぞれ 7，4，9，5 個であった．

最後に，予測精度を計測した．予測精度は，評価用データのうち，成功と予測して実際に成功していたプロジェクトと，失敗と予測して実際に失敗していたプロジェクトの和をプロジェクトの総数で割った値とする．予測には Weka [15] を使用した．その結果が表 3 である．表 3 の —は，絞り込みの結果としてメトリクスが 0 個となったために予測を行っていないことを表している．また，それぞれの欠損率について，最も高い精度を太字で表している．全てのデータのうち最も精度が高かったのは欠損率が 10%の場合で，7 個のメトリクスによって 82.8%を達成している．

5.2　分　　析

表 3 より，いずれの欠損率を利用する場合でも，高い精度で予測ができていることがわかる．このうち，最も高い予測精度となっていた欠損 10%のデータ 1 における (絞り込まれた)7 つのメトリクスについて分析する．

絞り込まれたメトリクスは，プロジェクトの性質を示すものと，文献 [9] で中核メトリクスと呼ばれているものから構成されている．「稼働後品質の目標が妥当か」，「アーキテクチャ」，「主開発言語」，「DBMS の利用」がプロジェクトの性質を示している．これらのメトリクスは，いずれも見積り値ではなく，設計が終了した時点で計測が可能なものである．

次に，残りの 3 つ「SLOC(機能量に該当)」，「月数_プロジェクト全体 (開発期間に該当)」，「月あたりの SLOC(生産性に該当)」が文献 [9] に挙げられる中核メトリクスにあてはまるものである．これらのメトリクスは，いずれも設計が終了した時点で見積りが可能なものである．本研究では，これらのメトリクスの値を二値化しているため，見積りの誤差による影響は少ないと考えている．

6. ま と め

本研究では，現実に企業で収集されているソフトウェア開発プロジェクトのデータを利用してプロジェクトの成否を設計段階で予測するための方法を提案した．

IPA/SEC が収集しているソフトウェア開発プロジェクトのデータを利用して，提案法の適用実験を行った．その結果，最も予測精度が高い場合でメトリクスは 7 個まで絞り込まれ，精度は 82.8%となった．

今後の課題としては，他のデータに対する提案手法の適用が考えられる．

謝辞　この研究の一部は，日本学術振興会科学技術研究費補助金基盤研究 (C)(課題番号：21500035)，及び日本学術振興会科学技術研究費補助金特別研究員奨励費 (課題番号：21・3963) の助成を受けている．

参 考 文 献

1) Abe, S., Mizuno, O., Kikuno, T., Kikuchi, N. and Hirayama, M.: Estimation of Project Success Using Bayesian Classifier, *Proc. of 28th International Conference on Software Engineering (ICSE2006)*, pp.600–603 (2006). Shanghai, China.
2) Cartwright, M.H., Shepperd, M.J. and Song, Q.: Dealing with Missing Software Project Data, *Software Metrics, IEEE International Symposium on*, Vol.0, p.154 (2003).
3) Chen, Z., Boehm, B., Menzies, T. and Port, D.: Finding the Right Data for Software Cost Modeling, *IEEE Software*, Vol.22, pp.38–46 (2005).
4) Dura, R.O., Hart, P.E. and Stork, D.G.: *Pattern Classification*, John Wiley & Sons, Inc. (2001).
5) Han, J. and Kamber, M.: *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers (2001).
6) Internatinal Software Benchmarking Standards Group: ISBSG Estimating, Benchmarking and Research Suite Release 11, http://www.isbsg.org/ (2009).
7) Kläs, M., Nakao, H., Elberzhager, F. and Münch, J.: Predicting Defect Content and Quality Assurance Effectiveness by Combining Expert Judgment and Defect Data - A Case Study, *Proceedings of the 2008 19th International Symposium on Software Reliability Engineering*, Washington, DC, USA, IEEE Computer Society, pp.17–26 (2008).
8) 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 Transactions on Software Engineering*, Vol.27, pp.999–1013 (2001).
9) Putnam, L.H. and Myers, W.: *Five Core Metrics: The Intelligence Behind Successful Software Management*, Dorset House Publishing Company, Incorporated (2003).
10) R Development Core Team: *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria (2010).
11) Shepperd, M. and Schofield, C.: Estimating Software Project Effort Using Analogies, *IEEE Transactions on Software Engineering*, Vol.23, pp.736–743 (1997).
12) Strike, K., Emam, K.E. and Madhavji, N.: Software Cost Estimation with Incomplete Data, *IEEE Transactions on Software Engineering*, Vol.27, pp.890–908 (2001).
13) Takagi, Y., Mizuno, O. and Kikuno, T.: An Empirical Approach to Characterizing Risky Software Projects Based on Logistic Regression Analysis, *Empirical Software Engineering*, Vol.10, No.4, pp.495–515 (2005).
14) Twala, B., Cartwright, M. and Shepperd, M.: Comparison of various methods for handling incomplete data in software engineering databases, *Empirical Software Engineering, International Symposium on*, Vol.0, p.10 pp. (2005).
15) Weka Machine Learning Project: Weka, URL http://www.cs.waikato.ac.nz/~ml/weka.
16) (独) 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（編）：ソフトウェア開発データ白書 2008，日経 BP 社 (2008).
17) 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).
18) 浜野康裕，天嵜聡介，水野　修，菊野　亨：相関ルールマイニングによるソフトウェア開発プロジェクト中のリスク要 因の分析，コンピュータソフトウェア， Vol.24, No.2, pp.79–87 (2007).
19) 瀧　進也，戸田航史，門田暁人，柿元　健，角田雅照，大杉直樹，松本健一：プロジェクト類似性に基づく工数見積もりに適した変数選択法，情報処理学会論文誌，Vol.49, No.7, pp.2338–2348 (2008).