Vol.2010-SE-170 No.1
2010/11/11

情報処理学会研究報告
IPSJ SIG Technical Report

ソフトウェアプロジェクト予測に用いるメトリクスの削減

尾形憲一[†1] 出張純也[†1] 菊野亨[†1]
菊地奈穂美[†2,†3] 平山雅之[†2]

ソフトウェア開発プロジェクトを管理するためには，プロジェクトの初期段階においてプロジェクトの成功を予測する事が有効である．

そうしたプロジェクトの成功予測における課題の 1 つに予測にかかるコストの削減がある．本研究では，その手段として，相関ルールマイニング法を用いて予測時に必要とするメトリクスの絞り込みを行う．なお，実用的データ環境での評価を行うために，IPA/SEC で収集されたデータを利用した．このデータの場合には記入率が低かったため，記入率を 73.2%まで上げる前処理を行った．実験の結果，相関ルールマイニングにより予測精度を 70%に維持しつつ，メトリクスを 58 個から 12 個にまで削減出来ることがわかった．

Reduction of the Number of Metrics Used for Prediction of Successful Software Projects

KENICHI OGATA ,[†1] JUNYA DEBARI ,[†1] TOHRU KIKUNO ,[†1]
NAHOMI KIKUCHI [†2,†3] and MASAYUKI HIRAYAMA [†2]

Predicting successful projects at an early stage enables a project manager to control projects.

In this paper we propose a novel method to reduce the number of metrics. The key idea is to select useful metrics from a given set of metrics by association rule mining.

As a case study, we applied the proposed method to actual project data collected by IPA/SEC. We first removed such data that had many missing values, because the data had a large amount of missing values. Then we applied proposed method to the remaining data having 58 metrics. As a result, we successfully reduced the number of metrics from 58 to 12 with keeping relatively high accuracy.

1. まえがき

1.1 背景

近年，ソフトウェアプロジェクトのプロダクトの品質不良によって稼働後の IT システムに障害が発生するなどして，いわゆる失敗するプロジェクトとなる事例が数多く報告されている [9]．そのために，過去に収集されたプロジェクトデータを利用してプロジェクトの成功を予測する試みが増えてきている．プロジェクト現場においてプロジェクトの成功の予測を行う際，①予測精度，②予測にかかるコスト，③データの欠損，と言った問題が存在することが知られている．ここで，予測にかかるコストとは「予測に必要なデータの収集にかかるコスト」であり,「データ収集に利用するメトリクスの総数」に強く依存する．なお，これら 3 つの問題は互いに独立ではなく，例えば，データの欠損が多い場合は予測精度が下がる，予測にかかるコストを減らすと予測精度が下がる，といった依存関係がある．本研究では，プロジェクト予測における 3 つの問題のうちで特に，予測にかかるコストの問題に着目する．

1.2 目的

本研究では，プロジェクトの成功予測にかかるコストを減らすことを目指した試みを行う．予測コスト増加の原因としては，予測モデルに多くのメトリクスを使用しており，適用のための情報収集にコストがかかる事が考えられる．そのため，データ収集コストを減らすためには，予測に用いるメトリクス数を減らすことが有効であると考えられる．

しかし，メトリクスを削減するとコストを減らせるが，予測に必要な情報も減少するため予測精度が下がってしまう心配がある．そこで，メトリクスの削減を予測精度をなるべく下げずに行うことが必要になる．本研究では，相関ルールに基づく方法を導入することによって，プロジェクトの成功予測の予測精度をそれほど下げずにメトリクス数を減らす試みに挑戦する．

1.3 関連研究

ソフトウェア開発プロジェクトにおいてプロジェクトに関するリスク分析をしてプロジェ

†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 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
Information-Technology Promotion Agency Software Engineering Center, Japan
†3 沖電気工業株式会社
Oki Electric Industry Co., Ltd

クトの最終状態に関する予測を行う研究 [3,6] が行われている．

また，ソフトウェア開発プロジェクトの早期段階で実施する問題分析アンケートによって，そのプロジェクトが最終的に混乱状態に陥るかどうかを判定する研究 [4,11] がある．特に文献 [11] では，プロジェクトの混乱を予測するのに，ベイズ識別器を用いた手法を提案している．この手法は，未回答の項目の含まれるアンケートに対しても簡単に予測を行う事ができる．文献 [11] では，記入率が 98%のデータを用いて分析を行っているが，本研究ではより記入率の低いデータを対象にしている．

一方，品質予測に関する研究として文献 [7] がある．この研究 [7] では，品質予測に関わるパラメータを決定し，それらを利用して予測モデルを構築している．なお，予測に用いるメトリクスの決定は専門家の知見と統計手法に基づいて行っている．本研究では，予測に用いるパラメータを定量的なデータ分析によって導くことを目指している．

2. プロジェクトの成功予測の現状

2.1 一般的なプロジェクトの成功予測

図 1 は一般的なプロジェクトの成功予測の流れを表している．

(1) モデル作成：過去のプロジェクトデータから予測モデルを作成する．この際，過去のプロジェクトデータに欠損が多い場合，データの欠損を減らす前処理を実行する．

(2) 予測：新しいプロジェクトデータを予測モデルに与えると，予測結果としてプロジェクトの成功確率が出力される．

(3) 確率の評価：成功確率が基準値以上なら成功，基準値未満なら失敗と判断する．(本研究では基準値は 0.5 と設定している．)

なお，プロジェクトの成功予測という観点では成功プロジェクトの確率だけ求めれば良いが，実際のフィールドを考えると失敗プロジェクトを予測することも重要である．このため，プロジェクトの成功と失敗をそれぞれ予測できるか見ていく必要がある．

2.2 利用するデータ

本研究は，一般的なプロジェクトの現場で適用できる (ある意味で，ロバストな) 予測を目指している．そのため，一般企業において収集されるデータをモデル作成，予測のフェーズで利用する．

これらの理由から，利用するデータとしてデータ白書 2008 [8] のものを採用する．このデータ白書 [8] は IPA/SEC によって 2007 年までに収集された企業 20 社のプロジェクトデータが記載されている．その特徴は，メトリクスの数が多く，一般の企業で使用されているよ

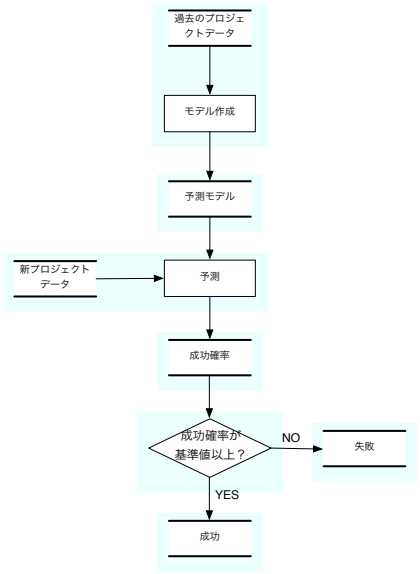

図 1 一般的なプロジェクトの成功予測

うなメトリクスが入っている．ところが，大きな問題としてそのデータには多くの欠損が存在する．

一般的に，ソフトウェアプロジェクトの成功は，品質，コスト，納期の 3 つの側面から評価される．本研究では，品質に関するメトリクスとして，データ白書 [8] の付録 A.4 における発生不具合数 (現象数) を目的変数とする．発生不具合数 (現象数) は，文献 [8] の付録 A.4 の定義の通りに計算したものである．また，説明変数 (メトリクス) の総数が 78 個，発生不具合数 (現象数) が記録されていたプロジェクトの総数が 816 件のデータセットを抽出した．この抽出されたデータセットにも多くの欠損値が含まれていた．

ここでは，データの値が欠損でなく，記入されている割合を記入率と呼び，次のように定義する．まず，含まれるメトリクスを $m_i(1 \leq i \leq 78)$，プロジェクトを $P_j(1 \leq j \leq 816)$ と表す．プロジェクトデータの各要素をデータ項目と呼ぶ．

- 全データに対する記入率 (%)：プロジェクトデータの全データ項目数の中で，値が記入

表 1 発生不具合数 (現象数) の統計値

データ件数	最小値	p25	中央値	p75	最大値
428 件	0	0	0	3	320

されているデータ項目数の割合.

- メトリクス m_i に関する記入率 (%)：メトリクス m_i に関する全データ項目数の中で，値が記入されているデータ項目数の割合.
- プロジェクト P_j に関する記入率 (%)：プロジェクト P_j に関する全データ項目数の中で，値が記入されているデータ項目数の割合.

これらの定義に基づいて計算すると，今回利用するプロジェクトデータ (816 プロジェクト，78 メトリクス) のデータ記入率は 48.3%であり，約半数は欠損している．この欠損による予測への影響を減らすために，研究 [10] の 4.2 節の方法で，記入率の低いデータの削除を行った．具体的には，メトリクスの記入率が 28.4%以下のメトリクスを削除し，プロジェクトの記入率が 53.4%以下のメトリクスを削除した.

その結果，メトリクス数=58，プロジェクト数=428，データ記入率=73.2%となった．これ以降,「本研究で利用するデータ」と述べたときはこのプロジェクトデータを指すものとする．また，この 428 件のプロジェクトの中で，発生不具合数 (現象数) が 0 と記載されていたプロジェクトは 242 件，発生不具合数 (現象数) が 1 件以上であったプロジェクトは 186 件含まれていた．表 1 に 428 件のプロジェクトについての発生不具合数 (現象数) の統計値を示す．本研究では，不具合が 0 件であったプロジェクトを成功プロジェクトとし，不具合が 1 件以上存在したプロジェクトを失敗プロジェクトとして分析を行う.

メトリクスの一覧を表 2 に記載する．データ白書によれば，メトリクスは番号ごとに表 3 のように分類されている．58 メトリクスのうち名義尺度の値をもつメトリクスは 26 個，順序尺度の値をもつメトリクスは 15 個，連続値の値をもつメトリクスは 17 個であった．また，この 58 個のメトリクスの値は文献 [10] に記載される方法で全て二値化してある．表 2 のメトリクスのうち番号の末尾に α，β が記載されているものは導出されたメトリクスである．例えば,「5004α_月あたりの SLOC」は「5004_SLOC 実測値_SLOC」を「5167α_実績月数プロジェクト全体」で割ることで求めている.

2.3 ベイズ識別器による予測

プロジェクトの成功予測はベイズ識別器を利用して行う．ベイズ識別器は単純かつ強力なデータの分類手法であり, それ自体で強力なデータマイニング手法になり得る.

表 2 利用したメトリクス一覧

メトリクス	記入率	型	メトリクス	記入率	型
103_開発プロジェクトの種別	100%	名義	411_コードジェネレータ利用	53.3%	名義
105_開発プロジェクトの形態	100%	名義	412_開発方法論利用	44.2%	名義
106_受託開発作業場所_1	52.1%	名義	422_開発フレームワークの利用	57.2%	名義
108_新規顧客	93.2%	名義	501_要求仕様_明確度合い	55.8%	順序
109_新規業種・業務	93.5%	名義	502_ユーザ担当者_要求仕様関与	49.5%	順序
110_新規協力会社_1	69.2%	名義	512_要求レベル_信頼性	61.0%	順序
111_新技術利用	85.3%	名義	514_要求レベル_性能・効率性	60.7%	順序
112_役割分担_責任所在	63.1%	順序	601_PM スキル	55.6%	順序
113_達成目標_優先度_明確度合い	56.5%	順序	602_要員スキル_業務分野経験	69.2%	順序
114_作業スペース	47.2%	順序	603_要員スキル_分析・設計経験	57.0%	順序
120_計画の評価 (コスト)	92.8%	順序	604_要員スキル_言語・ツール利用経験	61.2%	順序
121_計画の評価 (品質)	92.5%	順序	605_要員スキル_開発プラットフォーム使用経験	54.9%	順序
122_計画の評価 (工期)	93.0%	順序	5001_FP 実績値_調整前	35.3%	連続値
204_利用形態	100%	名義	5004_SLOC 実績値_SLOC	76.2%	連続値
301_システム種別	100%	名義	5223_平均要員数プロジェクト全体	67.5%	連続値
302_業務パッケージ_利用有無	98.6%	名義	5232_ピーク要員数プロジェクト全体	71.5%	連続値
307_処理形態_1	47.0%	名義	5251_テストケース数結合テスト	63.3%	連続値
308_アーキテクチャ_1	100%	名義	5252_テストケース数総合テスト (ベンダ確認)	70.1%	連続値
309_開発対象プラットフォーム_1	100%	名義	5253_検出バグ現象数結合テスト	61.7%	連続値
310_Web 技術の利用	100%	名義	5254_検出バグ現象数総合テスト (ベンダ確認)	68.5%	連続値
312_主開発言語_1	100%	名義	11015_プロジェクト開発工数計画値 (基本設計開始時点)	97.7%	連続値
313_DBMS の利用_1	100%	名義	5253α_テストケースあたりの検出バグ数結合テスト	72.9%	連続値
401_開発ライフサイクルモデル	100%	名義	5254α_テストケースあたりの検出バグ現象数総合テスト	81.8%	連続値
403_類似プロジェクト_有無	51.9%	名義	5223α_ピーク時と平均時の要員数比プロジェクト全体	84.6%	連続値
404_プロジェクト管理ツール利用	64.0%	名義	5167α_実績月数_プロジェクト全体	80.6%	連続値
405_構成管理ツール利用	62.9%	名義	5177α_実績開発工数	96.5%	連続値
406_設計支援ツール利用	57.7%	名義	5001α_月あたりの FP	35.3%	連続値
407_ドキュメント作成ツール利用	57.7%	名義	5004α_月あたりの SLOC	72.9%	連続値
408_デバッグ_テストツール利用	60.0%	名義	5177β_月あたりの工数	94.4%	連続値

表 3 メトリクスの分類

メトリクスの番号[⋆1]	メトリクスの種類
103〜122	開発プロジェクト全般
204	利用局面
301〜313	システム特性
401〜422	開発の進め方
501〜514	ユーザ要求管理
601〜605	要員等スキルと経験
5001〜5004(11015 も含む)	システム規模
5223〜5232(5177 も含む)	工数 (コスト)
5167	工期
5251〜5254	品質

ベイズ識別器を予測に用いる主な理由としては，①ベイズ識別器が確率として予測結果を示すこと，②記入率の低いデータに対しても適用可能であること，③先行研究 [1,11] から判

⋆1 末尾に α，β がつくメトリクス番号は，表 3 では同じ「メトリクスの番号」に纏めた.

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．具体的には，Waikato 大学で開発されているオープンソースのデータマイニングツール Weka [5] を使用する. 手法には, 最も基本的な手法である単純ベイズ識別器を用いた．ベイズ識別器の考え方の基礎となるベイズの定理などについては 3.5 節で述べる．

本研究で利用するプロジェクトデータ (428 プロジェクト，58 メトリクス) を用いてベイズ識別器によるプロジェクト予測を行った．その結果，予測精度 (成功プロジェクトを成功，失敗プロジェクトを失敗，と予測できた割合) は 71.7%であった．

3. 提 案 法

3.1 本研究の狙い

本研究での狙いは予測精度をなるべく下げずに，メトリクスの削減を行うことである．2.3 節の結果である 71.7%から判断して，予測精度の 7 割を維持することを 1 つの目安として，メトリクス総数を出来るだけ減らすことを目指す．そのためには，予測に有益であると見込まれるメトリクスを残す手法を考える必要がある．

3.2 提案法の概要

本研究における提案法の概要図は図 2 のようになっている．図 2 について説明する．

(1) 入力として評価データが与えられる．評価データは，事前に分析に適した加工がされているものとする．
(2) 評価データに対しメトリクスの絞り込みを行う (Phase1)．その結果，メトリクスの絞り込まれたデータが新しく作成される．
(3) 絞り込まれたデータを用いてプロジェクトの予測を行う (Phase2)．その結果，プロジェクト予測の予測精度が求まる．
(4) 求まった予測精度が 70%以上かどうか判定する．70%以上であればパラメータを調節して，メトリクスの絞り込みを繰り返す．
(5) (1)〜(4) を繰り返した結果，絞り込まれたメトリクスが出力される．

本研究では，Phase1 のメトリクスの絞り込みに相関ルールマイニングを用いる．3.3 節，3.4 節において相関ルールマイニングを用いたメトリクスの絞り込みについて詳しく説明する．

3.3 相関ルールマイニング

本研究では，予測に用いるメトリクスの絞り込みに相関ルールマイニングを用いる．相関ルールマイニングでは目的変数に関係のある相関ルールのみが抽出される．そのため，プロジェクトの成功と関係があると思われるメトリクスの選択に適していると思われる．例を用いて説明する．

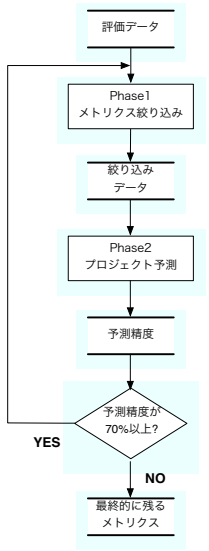

図 2 提案法の概略図

- Rule1：[605_開発プラットフォーム使用経験=ab] ∧ [月あたりの工数=Low]
  ⇒ [発生不具合数 (現象数)=Low]
  但し，信頼度=0.91，支持度=0.27 とする．

この相関ルール (Rule1) は「605_開発プラットフォーム使用経験が ab である (要員の半数以上は使用経験がある)」と「月あたりの工数が Low である」という事象が同時に (つまり AND 条件として) 起こると,「発生不具合現象数が Low である (すなわち，プロジェクトが成功)」という事象も起こりやすいことを意味している．この相関ルールにおいて，[605_開発プラットフォーム使用経験=ab] ∧ [月あたりの工数=Low] の部分を相関ルールの「前提部」，[発生不具合数 (現象数)=Low] の部分を相関ルールの「結論部」とそれぞれ呼ぶ．

相関ルールの重要性を測る指標として，支持度と信頼度を用いる．支持度は全データ中でルールがどの程度出現しているかを示す割合である．一方，信頼度は，前提部が成立する

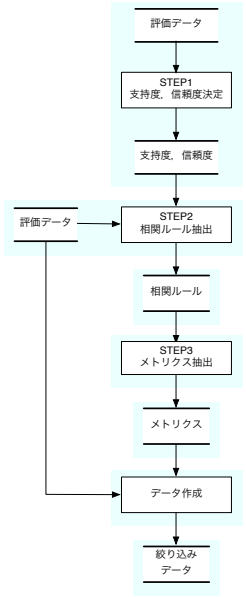

図 3 相関ルールマイニングによるメトリクス絞り込み

という条件下で結論部が発生する確率である．Rule1 はプロジェクトから実際に抽出されたルールである．支持度が 0.27 なので，Rule1 が 428 件の全プロジェクトの中で 116 件で成立している．また信頼度が 0.91 なので Rule1 の前提部が 127 件のプロジェクトで成立しており，そのうちの 116 件のプロジェクトで Rule1 が成立している．

3.4 Phase1(メトリクスの絞り込み)

図 3 は相関ルールマイニングを利用してメトリクスを抽出する様子を表している．メトリクスの絞り込みは 3 つのステップで行う．

3.4.1 STEP1(支持度，信頼度の決定)

相関ルールマイニングでは支持度と信頼度の 2 つのパラメータの値を調整することで，抽出する相関ルールの数が変化する．支持度が高いほど，多くのプロジェクトで成立するルールが抽出されるが，ルールの数が少なくなる．一方，信頼度が高いほど，抽出される相関ルールの前提部が成立したときに結論部も成立している割合が高くなるが，抽出されるルールの数が少なくなる．

ここでは，相関ルール抽出の際に与えるパラメータ (信頼度と支持度) を決定する．

3.4.2 STEP2(相関ルール抽出)，STEP3(メトリクス抽出)

評価データに対して相関ルールマイニングを行うことにより，例えば 3.3 節の Rule1 のような相関ルールが抽出される．Rule1 の前提部には次の 2 つの条件式が含まれる．

(1) [605_開発プラットフォーム使用経験=ab]

(2) [月あたりの工数=Low]

1 つ目の条件式からはメトリクス「605_開発プラットフォーム使用経験」，2 つ目の条件式からはメトリクス「月あたりの工数」がそれぞれ抽出される．このようにしてデータセットから，相関ルールによって得られたメトリクスだけを残してメトリクスを削除する．その結果として得られるデータが「絞り込みデータ」である．

本研究では，支持度を少しずつ増やして，抽出される相関ルールの数を減らしている．支持度を増やすことにより，抽出される相関ルールはより多くのプロジェクトにおいて不具合 (目的変数) に関連あるものに絞り込まれている．このため，相関ルールから抽出されるメトリクスも，より多くのプロジェクト上で不具合に関連あると考えられるものに絞り込まれることになる．

3.5 Phase2(プロジェクトの予測)

絞り込まれたデータを用いてベイズ識別器で予測を行う．ベイズ識別器は次節で説明するベイズの定理に基づいた確率的な識別器である．

3.5.1 ベイズの定理

ベイズの定理とは，事前確率を事後確率に変換するもので，あるデータが得られた時，その結果を反映した下での事後確率を求めるのに使われる [2].

確率変数 A, B において，

- 事前確率:$P(B)$ = 事象 B が発生する確率
- 事後確率:$P(B|A)$ = 事象 A が起きた下に，事象 B が発生する確率

とすると，$P(A) > 0$ の条件の下で

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

が成り立つ．これをベイズの定理という．

3.5.2 ベイズ識別器

データ d の属性集合を $\{q_1, q_2, \cdots, q_n\}$ とする．属性集合が，$q_1 = Q_1, q_2 = Q_2, \cdots, q_n =$

Vol.2010-SE-170 No.1
2010/11/11

Q_n と与えられたとき，名義変数 c が $c = C$ となる確率

$$P(c = C|q_1 = Q_1 \wedge q_2 = Q_2 \wedge \cdots \wedge q_n = Q_n)$$

は，ベイズの定理を用いて次のように表される．

$$\frac{P(c = C)P(q_1 = Q_1 \wedge \cdots \wedge q_n = Q_n|c = C)}{P(q_1 = Q_1 \wedge \cdots \wedge q_n = Q_n)}$$

例えば，

$$P(\text{発生不具合数 (現象数)} = \text{Low}|111\_\text{新技術利用} = \text{なし} \wedge \cdots \wedge \text{月あたりの SLOC} = \text{High})$$

という確率は，次式で求めることができる．

$$\frac{P(\text{発生不具合数 (現象数)} = \text{Low})P(111\_\text{新技術利用} = \text{なし} \wedge \cdots | \text{発生不具合数 (現象数)} = \text{Low})}{P(111\_\text{新技術利用} = \text{なし} \wedge \cdots \wedge \text{月あたりの SLOC} = \text{High})}$$

3.5.3 予測精度の評価方法

予測精度の評価には 10-fold cross validation を用いる．10-fold cross validation とは，「データを 10 個のグループに分け，1 つのグループを予測データとして，残った 9 個のグループのデータは全て学習データとして利用する」という手続きを 10 個のグループ全てについて繰り返す，という手法である．

具体的な予測結果の判定は，基準値に基づいて次のようにして行う．あるプロジェクト P_j について 10-fold cross validation による予測結果として，不具合ありの確率が基準値以上のときは不具合ありと決める．不具合ありの確率が基準値未満のときは不具合なしとする．なお，この基準値は変更可能な数値であるが，本研究では 0.5 としている．

本研究では，予測精度を以下のように定義する．

$$\text{予測精度} = \frac{\text{(成功プロジェクトを成功と予測した数)} + \text{(失敗プロジェクトを失敗と予測した数)}}{\text{全プロジェクト数}}$$

4. 適 用 実 験

4.1 実 験 概 要

4.1.1 実験の目的

本研究では 2 つの実験 A，B を行う．図 4 にその概要を示す．

- 実験 A ･･･ 相関ルールマイニングに基づく提案法によって，どこまでメトリクスを絞り込めるのかを確認する．
- 実験 B ･･･ メトリクスの絞り込みをランダムに実行した場合の予測精度の推移を確認する．

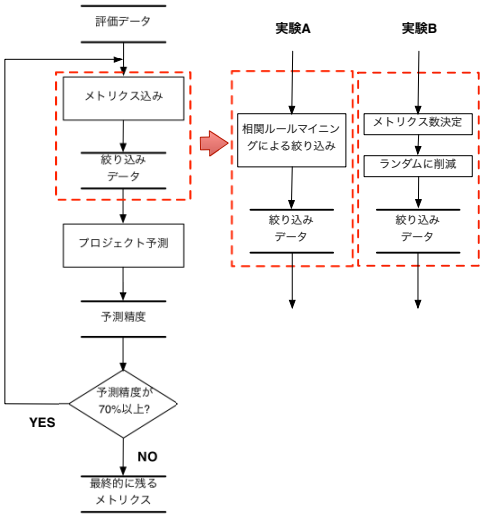

図 4 実験の説明図

まず，実験 A を行い，その後に実験 B を行う．これは，実験 A で求まったメトリクスの総数に関する情報を，実験 B で利用するためである．

4.1.2 実 験 A

実験 A では相関ルールマイニングに基づく提案法を適用して，メトリクスを絞り込んでいく．ここでは，相関ルールマイニングにおける信頼度は 0.9 で固定する．支持度は 0.1 から始めて，0.15，0.20，0.25，･･･ と 0.05 づつ支持度を増やす．

なお，図 4 からも分かるように，メトリクスの絞り込みが行われる度にプロジェクトの成功及び失敗の予測をして，予測精度が 70%以上であれば，支持度を増やして再度，相関ルールマイニングによる絞り込みを行う．但し，今回の研究では，どこまでメトリクスの総数を減らせるかに関心があるため，予測精度が 70%を切る前後の状況については詳細に観察する．

4.1.3 実 験 B

実験 B では，メトリクスをランダムに削減した場合との比較を行う．なお，具体的にランダムに削減するメトリクスの数は，実験 A で決定されたメトリクスの数を参考に決定する．

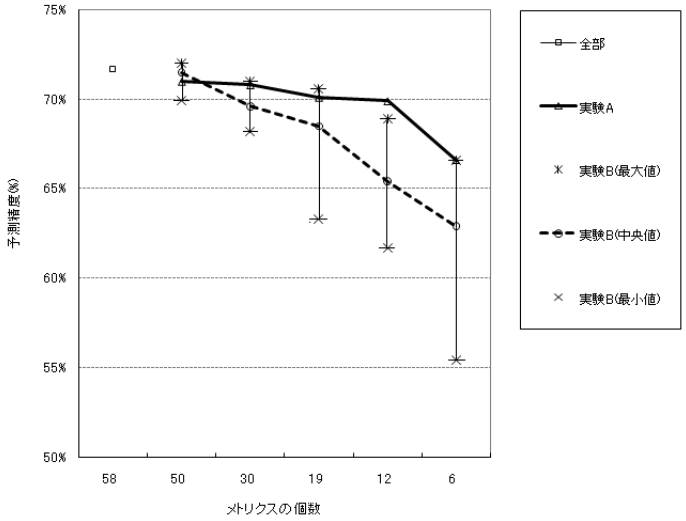

図 5 メトリクスの削減状況

ランダムにメトリクスを削減する方法を次に示す．以下では，メトリクスの総数を n 個から $k(1 \leq k < n)$ 個へとランダムに減らしている．

(1) メトリクスに番号をつけ，m_1〜m_n とする．

(2) 乱数を 1〜n までで発生させ，出てきた値が i ならメトリクス m_i を削除する．

(3) メトリクスの総数 n を 1 減らし，(1) に戻る．

(4) (1)〜(3) の一連の操作を $(n-k)$ 回行う．

メトリクスを減らす試みは 5 回行い，その度にプロジェクト予測を行う．得られた 5 回の予測精度の最大値，中央値，最小値を用いた評価を行う．

4.2 実験結果

図 5 は実験における結果 (メトリクスの総数とプロジェクト予測精度) をプロットしたものである．

実験 A の結果，メトリクスの総数は 50(支持度=0.1)，30(支持度=0.15)，19(支持度=0.2)，6(支持度=0.25) と変化している．ここで，メトリクス総数を 19 個から 6 個に減らしたときに予測精度が 70%を下まわった．そこでメトリクスの総数を 19，18，… と下げながら (実際には，支持度を増加させながら) 予測精度を求めた．その結果，メトリクスの総数が 12(支持度=0.212) まではほぼ 70%の予測精度が維持された．

比較実験として行った実験 B の結果については，予測精度の値の広がりが大きいために中央値で評価するのが妥当である．その中央値の予測精度を見る限り，70%の予測精度を維持すると，メトリクスの総数は 30 までしか絞り込むことが出来なかった．また，メトリクス数が少なくなるにつれて予測精度の最大値と最小値の幅が広がっていくことが分かる．

表 4 相関ルールマイニングで絞り込まれたメトリクス

メトリクス名	19 メトリクス	12 メトリクス	6 メトリクス
105_開発プロジェクト形態	○	○	○
108_新規顧客	○	○	
109_新規業種・業務	○	○	
111_新技術利用	○		
120_計画の評価 (コスト)	○	○	○
121_計画の評価 (品質)	○	○	○
122_計画の評価 (工期)	○	○	○
204_利用形態	○	○	
301_システム種別	○	○	
302_業務パッケージ_利用有無	○		
313_DBMS の利用_1	○		
401_開発ライフサイクルモデル	○	○	○
605_要員スキル_開発プラットフォーム使用経験	○		
5004_SLOC 実測値_SLOC	○	○	○
5251_テストケース数結合テスト	○		
5253_検出バグ現象数結合テスト	○		
5167α_実績月数_プロジェクト全体	○	○	
5177α_実績開発工数	○		
5004α_月あたりの SLOC	○	○	

5. 考察とまとめ

相関ルールマイニングにより絞り込んでいったときのメトリクスの状況を表 4 に示す．表 3 のメトリクスの種類に基づいて表 4 のメトリクスを分析すると表 5 のようになる．

表 5 から分かるように，相関ルールマイニングでメトリクスを絞り込むとメトリクスの総数が 19 個のときはほぼ全ての種類のメトリクスを含んでいる．ところがメトリクス総数を 6 個にまで削減すると 3 種類のメトリクスしか残っていない．このため，メトリクス総

表 5 絞り込まれたメトリクスの分類

メトリクスの種類	19 メトリクス	12 メトリクス	6 メトリクス
開発プロジェクト全般	○	○	○
利用局面	○	○	
システム特性	○	○	
開発の進め方	○	○	○
ユーザ要求管理			
要員等スキルと経験	○		
システム規模	○	○	○
工数 (コスト)	○		
工期	○	○	
品質	○		

数を 6 個にしたときは予測精度が下がったと考えられる．また，メトリクス総数が 12 個のときは，メトリクス総数が 19 個の場合に比べ，残っているメトリクスの種類は少ないが 6 種類のメトリクスを含んでいる．このため，メトリクス総数が 12 個のときにも 70%程度の予測精度を維持できていると考えられる．

今後の課題としては，今回の絞り込み方法をランダム以外の他の絞り込み方法と比較すること，他のデータセットに適用すること，等が挙げられる．

謝辞 この研究の一部は，経済産業省「平成 22 年度産業技術研究開発委託費 (中小企業システム基盤開発環境整備事業)」，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).
2) Dura, R.O., Hart, P.E. and Stork, D.G.: *Pattern Classification*, John Wiley & Sons, Inc. (2001).
3) Jiang, J. and Klein, G.: Software development risks to project effectiveness, *Journal of Systems and Software*, Vol.52, pp.3–10 (2000).
4) Mizuno, O., Kikuno, T., Takagi, Y. and Sakamoto, K.: Characterization of risky projects based on project managers' evaluation, *Proc.of 22nd International Conference on Software Engineering (ICSE2000)*, pp.387–395 (2000).
5) Weka Machine Learning Project : Weka 3 : Data Mining Software in Java, http://www.cs.waikato.ac.nz/.ml/weka/
6) Wohlin, C. and Andrews, A.A.: Prioritizing and Assessing Software Project Success Factors and Project Characteristics using Subjective Data, *Empirical Software Engineering*, Vol.8, pp. 285–303 (2003).
7) Klas, M., Nakao, H., Elberzhager F. and Munch, J.: Predicting Defect Content and Quality Assurance Effectiveness by Combining Expert Judgment and Defect Data - A Case Study, *Proc. of 19th International Symposium on Software Reliability Engineering (ISSRE2008)*, pp.17–26 (2008).
8) (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).
9) 経済産業省，(独) 情報処理推進機構：2008 年版組込みソフトウェア産業実態調査報告書，http://sec.ipa.go.jp/reports/20080715.html
10) 出張純也，尾形憲一，菊野亨，水野修，菊地奈穂美，平山雅之：ソフトウェア開発データに対する相関ルールマイニングを利用した不具合増加要因の調査，情報処理学会研究報告，2010-SE-167, No.3, pp.1–8 (2008).
11) 水野修，安部誠也，菊野亨：プロジェクト混乱予測システムのベイズ識別器を利用した開発，*SEC journal*, Vol.1, No.4, pp.24–35 (2005).