Vol.2010-SE-167 No.4
2010/3/18

ベイズ識別器による不具合予測のための相関ルールマイニングを用いたメトリクス絞り込み

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

近年，ソフトウェア開発現場において，プロジェクト管理の必要性が高まってきている．我々は，企業横断的データに対して，ソフトウェア稼働後に不具合が発生するかどうかを予測するベイズ識別器の作成を目指している．

本研究の目的は，その準備的考察として，モデル作成に利用するメトリクスの絞り込みの可能性を検討することである．絞り込みには相関ルールマイニングを利用する．SEC で収集されたデータを利用した評価適用実験を行い，メトリクスを絞り込んだ場合の各モデルについてその予測精度を調査した．実験の結果，相関ルールマイニングを利用して絞り込みを行うと，かなり少ないメトリクスだけでも高い予測精度を維持できることが分かった．

Selection of Metrics by Association Rule Mining for Identifying Risky Software Projects

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

Recently in the software development projects, the necessity of quantitative project management has been increased. We also try to develop, for project data crossing industries, a Bayesian classifier to identify such risky projects that may produce faulty product.

In this paper we try to choose an appropriate set of metrics for the Bayesian classifier. Actually we apply association rule mining for project data in order to determine the set of metrics. We conducted an experimental evaluation and found several candidate set of metrics for the Bayesian classifier.

1. はじめに

1.1 背景

近年，ソフトウェア開発現場において，ソフトウェアの開発期間が短縮される一方，ソフトウェアに対して求められる品質は高くなってきている．また，プロジェクトの生産物であるシステムの品質不良によって稼働後に障害が発生するなど，失敗となるソフトウェアプロジェクトが数多く報告されている [9]．このため，ソフトウェアプロジェクトの開発現場では，プロジェクトの抱える問題点を早期段階で発見し，回避することが重要である．しかしながら，開発プロジェクトの成功や失敗に繋がる要因は多岐にわたるため，プロジェクトを失敗に導く決定的なリスク要因を特定する事は難しい．こうしたリスク要因の特定や，特定した後のリスク回避方法の検討は経験に基づいて行われる事が多い．また，一方では，多くの開発プロジェクトから観察収集される種々の参考情報が有効に活用されていないという問題もある．そのため，現場で観察収集される規模・工数・工期・不具合数などの量的なデータや，プロジェクト特性を示す質的なデータなどを有効に活用して失敗を事前に回避する手法の確立が求められている．

1.2 目的

プロジェクトの失敗を回避するためには，プロジェクト早期の方が回避策のバリエーションが多く，有効な対策を実行しやすいため，できる限り早い段階でプロジェクトの見通しや健全性を評価することが重要である．しかし，プロジェクトの早期に手に入る情報には限界があるため，重点的に収集するべき情報を選択する必要がある．

そのため，本研究は，プロジェクトの中で観察収集される種々のメトリクスを用いて，ソフトウェアプロジェクトの成否をプロジェクト早期に予測することを目的とする．具体的には，ソフトウェア開発プロジェクトのデータに対してベイズ識別器を用いてプロジェクト結果の予測を行う．しかし，プロジェクトの初期段階では，一般的に多くの情報は収集できず，多くのメトリクスを用いて予測することは難しい．

予測を行う際には，予測に用いるメトリクスを，ソフトウェアプロジェクトの結果と関係

†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 京都工芸繊維大学 大学院工芸科学研究科
Graduate School of Science and Technology, Kyoto Institute of Technology
†3 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

Vol.2010-SE-167 No.4
2010/3/18

があると思われるメトリクスのみに絞り込みを行う．メトリクスの絞り込みの方法として，本研究では，相関ルールマイニングを利用する．具体的には，相関ルールマイニングの前提部に出現するメトリクスを「プロジェクトの成否に関連するメトリクス」と考え，ベイズ識別器による予測のための説明変数として用いることにする．

1.3 関 連 研 究

ソフトウェア開発プロジェクトにおけるリスク分類は Boehm らによって古くから行われてきており [2]，リスクを分析してプロジェクトの最終状態に関する予測を行う研究も行われている [4, 7].

我々の研究グループでは，ソフトウェア開発プロジェクトの早期に行う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手法を提案してきている [5, 11]．特に [11] では，プロジェクトの混乱を予測する手法として，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手法として提案している．

先行研究にベイズを用いてプロジェクトの成否の予測を行った研究 [1, 11] があったが，研究 [1] で用いられたデータは記入率が 95%以上，研究 [11] で用いられたデータは記入率が 100%，とどちらも記入率は非常に高いものであった．このうち，研究 [1] においてはベイズに利用するメトリクスを p 値により絞り込みをしてから予測を行っている．しかし，記入率の低いデータでは p 値による絞り込みを行うことが出来ないため，この手法を用いることはできない．プロジェクトの初期段階で予測を行おうとする場合，情報があまり存在しないため，予測に用いるデータの記入率は低い．そのため，扱うメトリクスの数が少なくても可能な予測手法を考える必要がある．本研究では，出来るだけ少ないメトリクスを用いて効率的に高い精度のプロジェクトの成否の予測を行うことを目的としている．

2. 準　　備

2.1 対象データ

本研究で分析に利用するデータは，IPA/SEC によって 2007 年までに収集され，データ白書 2008 [8] に示されているプロジェクトデータの一部である．一般的に，ソフトウェアプロジェクトの成否は，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] 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を行ったものである．

表 1 データ加工後の発生不具合数 (現象数) の統計値

データ件数	最小値	p25	中央値	p75	最大値
425 件	0	0	1.0	6.0	362

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．

収集されたデータ [8] のうち，発生不具合数 (現象数) が記録されていたプロジェクトデータ 816 件を最初に抽出した．不具合の有無予測のための説明変数は，研究 [10] のように，メトリクスを最初に 75 個選定した．そのデータセットでは，データの記入率 (値が欠損ではなく記入有りのもの) は全体で 49.4%で，データのうち約半分が記入されていた．

記入率が低いことによる分析への偏りなどの影響を減らすために，研究 [10] の 4.2 節に示されている方法で，記入率の低いデータの削除を行う．その結果，説明変数となるメトリクス数は 56 個，プロジェクト数は 425 件となった．この処理後のデータセットでの記入率は 74.3%となっている．また，この 425 件のプロジェクト中，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 205 件，発生不具合数 (現象数) が 1 以上であったプロジェクトは 220 件存在した．表 1 に 425 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．表 2 は，今回実験に利用したメトリクスの一覧である．表 2 の 56 個のメトリクスの値は，文献 [10] に記載する方法で全て二値化してある．

2.2 ベイズを利用する方式

本研究では，プロジェクトの不具合発生を予測するために，ベイズ識別器を利用している．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．

ベイズ識別器を不具合予測に用いるのは，ベイズ識別器が確率として予測結果を示す点，記入率の低いデータに対しても適用可能である点，そして，先行研究 [1, 11] である程度の適用可能性が期待できる点が挙げられる．

本研究では, 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 [6] 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や，単純ベイズ識別器による確率の計算方法を以下で説明する．

2.2.1 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

Vol.2010-SE-167 No.4
2010/3/18

表 2 分析に利用したメトリクス

名義尺度の値をとるメトリクス	
開発プロジェクトの種別	開発プロジェクトの形態
受託開発の場合の作業場所	新規顧客
新規業種・業務	新規協力会社
新技術の利用	利用形態
システムの種別	業務パッケージ利用の有無
処理形態	アーキテクチャ
Web 技術の利用	DBMS の利用
開発ライフサイクルモデル	類似プロジェクト参照の有無
プロジェクト管理ツールの利用	構成管理ツールの利用
設計支援ツールの利用	ドキュメント作成ツールの利用
デバッグ・テストツールの利用	CASE ツールの利用
コードジェネレータの利用	開発方法論の利用
開発フレームワークの利用	
順序尺度の値をとるメトリクス	
開発プロジェクトチーム内での役割分担・責任所在の明確さ	達成目標と優先度の明確さ
作業スペース	計画の評価 (コスト)
計画の評価 (品質)	計画の評価 (工期)
実績の評価 (コスト)	実績の評価 (品質)
実績の評価 (工期)	要求仕様の明確さ
ユーザ担当者の要求仕様関与	要求レベル (信頼性)
要求レベル (性能・効率性)	PM スキル
要員スキル_業務分野経験	要員スキル_分析・設計経験
要員スキル_言語・ツール利用経験	要員スキル_開発プラットフォームの使用経験
連続値をとるメトリクス	
FP 実績値_調整前	SLOC 実績値_SLOC
平均要員数プロジェクト全体	ピーク要員数プロジェクト全体
検出バグ現象数結合テスト	実績開発工数
実績月数_プロジェクト全体	月あたりの FP
月あたりの SLOC	月あたりの工数
納期遅延	工数超過
ピーク時と平均時の要員数の比	

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 [3].

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 が起きた後に事象 B が発生する確率

とすると，$P(A) > 0$ の条件の元で

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

が成り立つ．これをベイズの定理という.

2.2.2 単純ベイズ識別器の確率計算方法

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，$q_1 = Q_1, q_2 = Q_2, \cdots, q_n = Q_n$ 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.

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

ここで，分母は $c = C$ に依存しておらず，分母が一定であるように $q_i = Q_i (1 \leq i \leq n)$ が与えられるため，分子だけを考慮すればよい．分子は次のように表される同時確率モデルと等価である.

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

この式に条件付き確率の定義を適用すると，次のように書き換えられる.

$$\begin{aligned} &P(c = C \wedge q_1 = Q_1 \wedge \cdots \wedge q_n = Q_n) \\ &= P(c = C)P(q_1 = Q_1 \wedge \cdots \wedge q_n = Q_n|c = C) \\ &= P(c = C)P(q_1 = Q_1|c = C)P(q_2 = Q_2 \wedge \cdots \wedge q_n = Q_n|c = C \wedge q_1 = Q_1) \end{aligned}$$

ここで，i，$j (1 \leq i, j \leq n)$ について，各属性 q_i が互いに独立であると仮定すると，次が成り立つ.

$$P(q_i = Q_i|c = C \wedge q_j = Q_j) = P(q_i = Q_i|c = C)$$

従って,

Vol.2010-SE-167 No.4
2010/3/18

$$
\begin{aligned}
&P(c = C \wedge q_1 = Q_1 \wedge \cdots \wedge q_n = Q_n) \\
&= P(c = C)P(q_1 = Q_1|c = C)P(q_2 = Q_2|c = C)\cdots \\
&= P(c = C)\prod_{i=1}^{n} P(q_i = Q_i|c = C)
\end{aligned}
$$

$P(q_i = Q_i|c = C)$ の値は学習データの数え上げによって求められる．

2.3 相関ルールマイニング

本研究では，ベイズ識別器による予測に用いるメトリクスを絞り込むための手段として，相関ルールマイニングを用いる．ここで，相関とは，ある事象が発生すると別の事象が発生しやすいという共起性を意味している．例えば，$A \Rightarrow B$ という相関ルールは，A という事象が起こると B という事象も起こりやすいということを意味している．この相関ルールにおいて，A の部分を相関ルールの「前提部」，B の部分を相関ルールの「結論部」と呼ぶ．

相関ルールの重要性を測る指標として，支持度 (support) と信頼度 (confidence) が存在する．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，前提部が成立するという条件下で結論部が発生する確率である．$A \Rightarrow B$ という相関ルールが存在する場合，支持度は全データ中で A と B が同時に発生しているデータの割合を示す．信頼度は，A が発生している条件下で B が発生している割合を示す．

3. 適 用 実 験

3.1 適 用 準 備

本研究で行うメトリクスの絞込み方について説明する．メトリクスの中に発生不具合数 (現象数) が存在するが，発生不具合数 (現象数) は今回予測対象のクラスとして扱うため，メトリクスの数には含めていない．

研究 [10] では，本研究で用いるものと同様のデータに対して，結論部を「発生不具合数 (現象数)=0」として相関ルールマイニングを行い，不具合数と関連があるかどうかの調査を行っている．その結果，22 個のメトリクスが相関ルールの前提部に現れている．さらに，相関ルールマイニングによって抽出されたメトリクスに対して追加の調査を行い，不具合数への関連が強いと思われるメトリクスから順に C_1，C_2，C_3 の 3 つのクラスに分類している．表 3 はそれぞれのカテゴリに分類されるメトリクスの一覧である．

本実験では，予測モデルの構築に用いるメトリクスの数を変化させながら，5 種類の実験を行う．以下に各実験の内容について説明する．

表 3 不具合と関連があると考えられるメトリクス一覧

C_1 に属するメトリクス	C_2 に属するメトリクス	C_3 に属するメトリクス
実績開発工数	新技術の利用	要員スキル_言語・ツール利用経験
SLOC 実績値_SLOC	実績の評価 (品質)	新規顧客
実績月数_プロジェクト全体	要員スキル_開発プラットフォーム使用経験	新規業務・業種
月あたりの SLOC	要員スキル_業務分野経験	納期遅延
要員スキル_分析・設計経験	業務パッケージ_利用有無	計画の評価 (品質)
	実績の評価 (コスト)	実績の評価 (工期)
		計画の評価 (工期)
		計画の評価 (コスト)
		システム種別
		開発ライフサイクルモデル
		開発プロジェクト形態

3.2 実 験 内 容

実験の概要を図 1 に示す．D_1 は，表 2 のメトリクスからなるプロジェクト・データセットである．D_1 に対して相関ルールマイニングを行い，メトリクスを選択した結果が D_2 であり，表 3 の C_1，C_2，C_3 全てのメトリクスで構成されている．D_2 のメトリクスのうち，プロジェクトの背景となっていると思われる C_3 のメトリクスを除いたものが D_3 であり，C_1，C_2 のメトリクスで構成されている．D_4 は，さらに不具合と関連が強いと思われる C_1 のメトリクスのみのデータである．また，図 1 には記されていないが，D_1 に対してルールマイニングを行った結果選択されなかったメトリクスによるデータ集合として，D_5(34 メトリクスから成る) を準備している．以下の実験 1〜5 は，それぞれ D_1〜D_5 を利用して行う．

実　験　1

実験 1 では，図 1 における D_1 のデータを用いて予測モデルを構築し，予測精度の評価を行う．ここでは，不具合に関係があると考えら得るメトリクスと不具合に関係がないと考えられるメトリクスを両方利用した場合の予測精度を調査する．

実　験　2

実験 2 では，図 1 における D_2 のデータを用いて予測モデルを構築し，予測精度の評価を行う．メトリクス数は実験 1 と比較すると半分以下となっている．ここでは，実験 1 で用いたメトリクスのうち，不具合に関係があると考えられるメトリクスのみに絞った場合，予測精度にどのような違いが出るか調査する．

実　験　3

実験 3 では，図 1 における D_3 のデータを用いて予測モデルを構築し，予測精度の評価を

Vol.2010-SE-167 No.4
2010/3/18

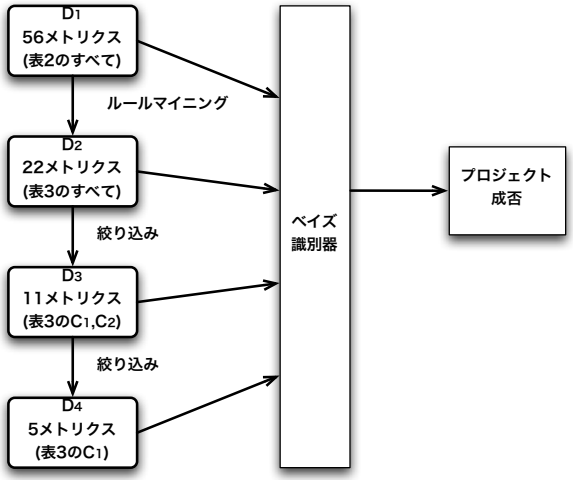

図 1 実験の概要

表 4 実験結果の例

実際の結果	予測の結果	
	不具合あり	不具合なし
不具合あり	a	b
不具合なし	c	d

行う．予測に用いるメトリクスは 11 個である．相関ルールマイニングによって選択されたメトリクスのうち，追加調査によって,「分析対象データセットの背景」と考えられるものを除いている．このデータ絞り込みによって，予測精度がどのように変化するかを調査する．

実　験　4

実験 4 では，図 1 における D_4 のデータを用いて予測モデルを構築し，予測精度の評価を行う．予測に用いるメトリクスは 5 個である．不具合と関連の強いメトリクスのみを用いる事で予測精度がどのように変化するかを調査する．

実　験　5

実験 5 では，D_5 のデータを用いて予測モデルを構築し，予測精度の評価を行う．ここでは，不具合に関係がないと考えられるメトリクスのみを利用して予測を行い，絞り込み方によって予測精度に差がでるかの調査を行う事が目的である．

3.3 予測精度の評価

予測精度評価の方法には 10-fold cross validation を用いている．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．あるプロジェクトについて 10-fold cross validation によって予測を行った結果，不具合ありの確率が 0.5 以上のとき，不具合ありと予測され，不具合ありの確率が 0.5 未満のとき，不具合なしと予測される．全てのプロジェクトについて, この評価方法によって予測を行った結果を表 4 のような形式で纏めている．

a, b, c, d の数値の意味は，それぞれ以下のようになっている．

a　実際に不具合があった (1 個以上) プロジェクトで，予測結果は不具合ありとなったプロジェクト数

b　実際に不具合があった (1 個以上) プロジェクトで，予測結果は不具合なし (0 個) となったプロジェクト数

c　実際に不具合がなかった (0 個) プロジェクトで，予測結果は不具合あり (1 個以上) となったプロジェクト数

d　実際に不具合がなかった (0 個) プロジェクトで，予測結果は不具合なしとなったプロジェクト数

ここで，正しく予測が行われたプロジェクトは a 及び d に含まれるプロジェクトである．つまり，予測精度は $(a+d) \div (a+b+c+d) \times 100$ により求まる．

表 5 実験 1 の結果

実際の結果	予測の結果	
	不具合あり	不具合なし
不具合あり	162	58
不具合なし	59	146

表 6 実験 2 の結果

実際の結果	予測の結果	
	不具合あり	不具合なし
不具合あり	156	64
不具合なし	61	144

表 7 実験 3 の結果

実際の結果	予測の結果	
	不具合あり	不具合なし
不具合あり	151	69
不具合なし	62	143

表 8 実験 4 の結果

実際の結果	予測の結果	
	不具合あり	不具合なし
不具合あり	152	68
不具合なし	66	139

表 9 実験 5 の結果

実際の結果	予測の結果	
	不具合あり	不具合なし
不具合あり	135	85
不具合なし	72	133

表 10 各実験における予測精度

実験 No	使用したメトリクス	予測精度
実験 1	D_1(56 メトリクス)	72.5%
実験 2	D_2(22 メトリクス)	70.6%
実験 3	D_3(11 メトリクス)	69.2%
実験 4	D_4(5 メトリクス)	68.5%
実験 5	D_5(34 メトリクス)	63.1%

4. 結果と考察

4.1 結　　果

実験結果例

予測の実例を，あるプロジェクト PRJ のデータを用いて以下に示す．PRJ を含まないデータセットで実験 4 の予測モデルが既に構築されているとする．PRJ のうち，実験 4 のモデルで不具合発生確率を予測するために必要なデータは { 実績開発工数=中央値未満，SLOC 実績値_SLOC=中央値以上，実績月数_プロジェクト全体=中央値未満，月あたりの SLOC=中央値以上，要員スキル_分析・設計経験=a,b} である．

このデータを用いて予測を行うと，不具合ありの確率が 0.937 となった．不具合ありの確率が 0.5 以上であるため，プロジェクト PRJ は「不具合あり」と予測された．プロジェクト PRJ には実際に不具合があるため，このプロジェクトによる予測結果は a となる．

以下，本研究における実験結果について述べる．

実　験　1

実験 1 の結果は表 5 のようになった. 予測精度は 72.5%と高くなっている．

実　験　2

実験 2 の結果は表 6 のようになった．メトリクス数が実験 1 に比べて半分以下となっているが，予測精度は 70.6%となり，実験 1 に比べて 1.9%低くなった程度である．

実　験　3

実験 3 の結果は表 7 のようになった．予測精度は 69.2%となり，実験 2 と比べて 1.4%低くなっている．

実　験　4

実験 4 の結果は表 8 のようになった．予測精度は 68.5%となり，実験 3 と比べて 0.7%低くなっている．

実　験　5

実験 5 の結果は表 9 のようになった．予測精度は 63.1%となっている．実験 1, 2, 3, 4 の結果と比べると極めて低い結果となっている．

4.2 考　　察

表 10 に各実験における予測精度をまとめた. 実験 1 から実験 4 までの結果を見てみると，メトリクスを絞り込んで予測を行った場合，予測精度はそれほど低くならず，ある程度予測が可能であることが分かった．実験 4 ではメトリクスを 5 個のみ利用しているが，メトリクスを 56 個利用している実験 1 と比較したとき，予測精度は 4%低くなった程度である．この結果から，今回絞り込んだ 5 つのメトリクスさえ分かれば，ある程度の予測は十分可能であるといえる．

また，実験 5 の結果を見てみると，メトリクスを 34 個用いているにもかかわらず，予測精度は 63.1%と低くなっている．これは，実験 4 でメトリクス 5 個を用いた場合の予測精度が 68.5%であることを考えるとかなり低いことが分かる．このことから，今回相関ルールマイニングにより抽出された 22 個のメトリクス及び更に絞り込まれたメトリクスは予測にかなり有意であったといえる．この結果から，相関ルールマイニングで利用するメトリクスを選択することは有効なアプローチであると考える．

従来研究 [1] では，単純ベイズ識別器による予測精度は 82.5%と非常に高いものとなっている．しかし，研究 [1] では記入率 95%以上という記入率の極めて高いデータを用いており，情報の少ないプロジェクトの初期段階における利用が困難な手法であると考えられる．

Vol.2010-SE-167 No.4
2010/3/18

今回用いたデータは記入率 75%程度のものであり，[1] で用いていたものよりも低くなっているが，予測精度は 70%を超えて高くなった．このことから，本研究の手法は従来の手法よりも未記入データに強く，プロジェクトの初期段階での予測において有益と考えられる．

ただし，本研究では，メトリクス選択のために相関ルールマイニングを適用したプロジェクトデータセットと同じデータセットを用いてベイズ識別器による予測モデルを構築している．そのため，予測精度が高くなっている可能性がある．また，今回利用したデータについて，説明変数が独立であるかどうかの検討はしていない．しかし，ベイズ識別器は各属性が独立であるという前提が崩れていても高い精度の予測が可能である [3] と言われているため，今回の予測結果に強い影響をはぼしていないと考えている．

5. ま と め

本研究では，ソフトウェアプロジェクトの成否を早期段階で予測するための手法として，ソフトウェアプロジェクトで収集されるメトリクスに対してベイズ識別器を用いて予測を行う手法に関して，予測モデル構築の際に用いるメトリクス数の絞り込みを行い，予測精度への影響を調査した．

その結果，相関ルールマイニングを用いて不具合と関連があると思われるメトリクスだけを選んで予測モデルを構築する場合と，絞り込みを行わずに多くのメトリクスを利用した場合ではあまり予測精度が変わらない事がわかった．また，相関ルールマイニングによって選ばれなかったデータを用いて予測モデルを構築した場合には，予測精度が下がることがわかった．

この結果から，適切にメトリクスの絞り込みを行う事で，少ない情報からでもプロジェクトの成否をある程度の精度で予測できる事がわかった．

今後の課題は，メトリクス選択に用いるデータと予測精度の評価に用いるデータを分けて，追加実験の実施があげられる．

謝辞 この研究の一部は，経済産業省「平成 21 年度産業技術研究開発委託費（産学連携ソフトウェア工学実践事業)」，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). Shanghai, China.
2) Boehm, B.W.: Industrial software metrics top 10 list, *IEEE Software*, Vol.4, No.5, pp.84–85 (1987).
3) Dura, R.O., Hart, P.E. and Stork, D.G.: *Pattern Classification*, John Wiley & Sons, Inc. (2001).
4) Jiang, J. and Klein, G.: Software development risks to project effectiveness, *Journal of Systems and Software*, Vol.52, pp.3–10 (2000).
5) 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*, pp.387–395 (2000).
6) Weka Machine Learning Project : Weka 3 : Data Mining Software in Java, http://www.cs.waikato.ac.nz/.ml/weka/
7) 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).
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 (2008).
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 (to appear).
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).