

### 定性的なソフトウェアプロジェクトデータに基づくプロダクト品質予測に関する研究

新井, 雄一郎 / ARAI, Yuichiro

---

(出版者 / Publisher)

法政大学大学院理工学・工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学・工学研究科編 / 法政大学大学院紀要. 理工学・工学研究科編

(巻 / Volume)

57

(開始ページ / Start Page)

1

(終了ページ / End Page)

6

(発行年 / Year)

2016-03-24

(URL)

<https://doi.org/10.15002/00013370>

# 定性的なソフトウェアプロジェクトデータに基づく プロダクト品質予測に関する研究

A STUDY ON PRODUCT QUALITY PREDICTION BASED ON  
THE QUALITATIVE SOFTWARE DEVELOPMENT PROJECT DATA

新井 雄一郎

Yuichiro Arai

指導教員 木村 光宏

法政大学大学院理工学研究科システム工学専攻修士課程

This article discusses a method for finding the software development project in which the developed software may cause at least one software failure in its operational phase. For this, we analyze the data sets which were obtained before the test phase as the results of questionnaire from the real domestic software development companies. In particular, the data sets consist of not quantitative variables but qualitative ones. As a result of the actual data analysis, we found that the Random Forests showed better performance against the logistic regression analysis with dummy variables.

*Key Words* : Software reliability, qualitative variable, machine learning, Random Forests

## 1. はじめに

### (1) 研究背景

近年、我々が生活する現代社会において、IT(information technology)の担う役割は非常に重要になり、我々の生活とは切っても切れない関係となった。例えば、銀行の口座はITによって管理され、電車の運行もITによって管理されている。ITに頼らない生活は現代では困難であろう。このような我々の生活と結びついているITとは全て、ソフトウェアによって動いている。したがって、ソフトウェア開発の現場には滞りの無い確実に安全な開発が求められている。例えば、ソフトウェアの開発段階からリリース後に発生する不具合の数が予測することができれば、このような要求の実現の大きな助けとなるだろう。このようなアイディアは特に新しいものではなく、以前から議論されている[1]。例えば、例えば開発人員数やSLOC(Source Lines Of Code)数などの定量的に表現されたデータを元に、バグの有無もしくは数の推定を行うモデルの提案は今までに多くされている。しかしC言語やJAVA言語などの開発に使用されたプログラミング言語であったり、ソフトウェアがどのように使用されることが想定されているか、ソフトウェア開発の管理者の熟練度などの様な選択式のアンケートの質問に対する回答のみで構成されたデータをもとにバグの有無の推定を行う報告は前述のモデルに比べ多くない。しかし、このような予測もまた、開発者にとっては有用であると考えられる。なぜならば量的なデータに比べ、用意されたものの中から適するものを選ぶ選択式のアンケートのような質的なデータの方が、集めやすいからである。それによって開発しているソフトウェアがリリース後にバグが発生するかどうかを予測できれば開発者の大きな助

けとなる。

図1は最も基本的で一般的なソフトウェア開発のプロセスである。これはソフトウェアの開発過程をいくつかの工程に分解して、各工程の終了時には文書を作成して次の工程に移るといったものである。滝の水が流れ落ちる様子にたとえ、ウォーターフォールモデルと呼ばれる。これは要件定義、外部設計、内部設計、プログラミング、テストの5つの工程からなる。ウォーターフォールモデルは滝の水が逆流することがないのと同じように、次の工程に進んだら原則的には前のステップに逆戻りをしないように開発を進める。これにより、全体を見通すことができ、スケジュールの立案や資源配分、進捗状況の理解が容易にできるなどの点から、高い信頼性が要求される大規模なシステムを開発する際に、有効な代表的手法として使われ続けている。ソフトウェア開発の初期において要求定義を明確にできる状況において適切な開発モデルである。

ソフトウェア開発におけるテスト工程には、多くのコストと人員が投入され、あらゆる状況でも要求通りにソフトウェアが動作するかが検証される。テスト工程よりも前の段階で確実にあれば望ましいが、ある程度の確率でリリース後に不具合が発生するかどうかを予測出来れば大きな助けとなるだろう。なぜならばもし、不具合が発生すると予測されれば、通常以上にテスト工程に人員を投入することにより、安全なソフトウェア開発が可能となる。逆に不具合が発生しないと予測されれば、通常通り、若しくは通常以下の人員にすることで他の工程に人員を配置することができ、コストの削減や、適材適所の人員配置が可能となる。このように、テスト工程よりも前の段階で得られる質的なデータを元にリリース後に開発しているソフトウェアが不具合が発生するかどうかを予

測するモデルの議論は非常に有用であると考えられる。

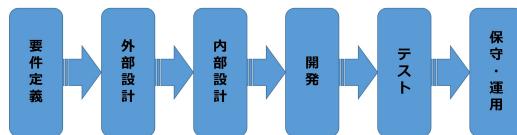


図1 ウォーターフォールモデル。

## (2) 研究目的

本研究の目的はソフトウェア開発プロジェクトにおいてテスト工程よりも前段階で得られる質的データを元に、ソフトウェアがリリースヶ月以内に不具合が発生するか否かの予測である。一般に多変量解析や機械学習を用いてデータセットを分析もしくは予測を行う場合、説明変数は距離の概念が考慮されている量的変数である必要がある。従って本研究で用いる説明変数によって構成されたデータセットには本来適応することは出来ない。そこで本研究ではダミー変数変換という処理をデータセットに対して行う。この処理によって質的変数は擬似的な量的変数に変換され量的変数と同等に扱う事が可能になる。そして質的変数によって構成されたデータセットにダミー変数変換を施し、分析が可能な状態にし、Random Forests と呼ばれる機械学習の手法を適用することで、予測精度の評価を行うことを目的にした。また Random Forests は学習過程で各変数が分類結果にどの程度影響しているかを定量的に評価することが可能である。そこでソフトウェア開発においてバグの発生の有無にはどの項目が影響しやすいかを考察することも目的とした。このようなモデルの提案はソフトウェアの品質や信頼性を定量的に評価を行うソフトウェア信頼性工学において報告されていない。

## 2. データセット

本研究では独立行政法人情報処理推進機構技術本部ソフトウェア・エンジニアリング・センター (IPA/SEC) によって提供されているデータ、「ソフトウェア開発データ白書 2012-2013」[2] の元データを用いた。このデータは 2005 年から 2012 年の間に国内 24 社の企業が実際に開発を行ったソフトウェア開発プロジェクトについて開発形態やプロジェクトの概要等をまとめたものである。このデータの規模は 3,089 件のプロジェクトに対して 611 項目の回答である。

### (1) データセットの抽出

本研究の目的は、1.2 節で述べた様にソフトウェア開発におけるテスト工程以前に得られる質的データのみを用いてソフトウェアがリリースされたヶ月以内に不具合が発生するか否かを予測することである。従って本研究の条件を満たすデータを抽出する必要がある。ただし、欠損データは許容しないものとする。しかし提供されているデータは全てのプロジェクトの全ての項目が回答されているわけではなく大量の欠損値を含んでいる。よって本研究の前提条件を完全に満たすデータを抽出しようとする非常に少数になってしまい、予測の精度の評価が困難になる。そこで本研究では回答数の少ない質的データに関する項目は除外し、「プロジェクト数は多いが、項目数は少ないデータセット」と「プロジェクト数は少ないが、項目数は多いデータセット」の二種類のデータ

セットを用意した。本研究では以降、前者をデータセット A、後者をデータセット B と呼ぶことにする。

### (2) 目的変数の設定と加工

本研究ではデータセット A、データセット B の項目の「5267 発生不具合現象数 (合計) 1ヶ月」を加工処理した新たな変数を目的変数に設定した。本研究の目的は「リリースヶ月以内の不具合の発生の有無」を予測することであるので、各プロジェクトが「5267 発生不具合現象数 (合計) 1ヶ月」で 1 以上の値が記載されていれば「Bug」、0 が記載されていれば「NonBug」と書き換え、その項目名を「リリースヶ月以内の不具合の発生の有無」と設定した。

### (3) ダミー変数変換

前述のとおり本研究の目的は質的変数によって構成されたデータセットの目的変数の推定である。しかしながら質的変数には変数間に距離の概念が考慮されていないため量的変数と同等に扱う事が出来ない。従ってデータセットに対し、分析が可能になるように何らかの加工を施す必要がある。具体的な加工方法としてダミー変数変換という処理が挙げられる。ダミー変数変換とは選択肢として  $1, 2, \dots, n$  を取る質的変数  $x_i$  を  $n-1$  個のダミー変数  $x_{i1}, x_{i2}, \dots, x_{i(n-1)}$  に変換を行う事で擬似的な量的変数を作ることである。ここで注意すべき点は  $x_i$  は 1 から  $n$  の値を取りうるが、これは量的な値ではなく、質的な値であるということである。 $x_{i1}, x_{i2}, \dots, x_{i(n-1)}$  はそれぞれ 2 値の値を取るが、0, 1 の 2 値 (バイナリ変数) を採用するのが一般的である。この作業をすべての質的変数に施すことで質的変数によって構成されたデータセットを量的変数で構成されたデータセットと同等に扱うことを可能にする。なお  $n$  種類の値を取る質的変数に対してダミー変数は  $x_{i1}, x_{i2}, \dots, x_{in}$  の  $n$  個を用意することが可能であるが、このまま扱うと多重共線性が発生するため 1 つの説明変数に対して任意の 1 つのダミー変数を削除する必要がある。変換されたダミー変数が  $n-1$  個であるのはこのためである。ダミー変数変換の例を表 1 と表 2 に示す。

表 1 はダミー変数変換前のデータであり 10 個の説明変数、1 個の目的変数を持つ 10 個のデータによって構成されている。各クラス間の距離は考慮されていないのでこのデータをそのまま分析することは困難である。そこでダミー変数変換を用いると表 1 のデータは表 2 の様になる。変換前の説明変数はそれぞれ  $x_1$  は 4 種類、 $x_2$  は 4 種類、 $x_3$  は 3 種類、 $x_4$  は 5 種類、 $x_5$  は 3 種類、 $x_6$  は 2 種類の選択肢を持っていた。従って、変換後の説明変数の個数は変換前の説明変数の選択肢の種類である 21 種から多重共線性を避けるために変換前の説明変数分だけ引いた 15 個となる。例えば、表 1 の  $x_4$  の  $ID = 2$  のデータを  $x_{2,4}$  と表記することになると  $x_{2,4} = e$  である。このデータをダミー変数変換を施すと、同様の表記を用いて  $e$  は  $\{x_{2,41} x_{2,42} x_{2,43} x_{2,44}\} = \{0 0 0 1\}$  と表現される。

また、本研究で用いるデータセットにダミー変数変換を施したところ変数の数はデータセット A の場合、40 変数 (変換前) から 114 変数 (変換後) となり、データセット B の場合、58 変数 (変換前) から 169 変数 (変換後) となった。

表2 データの例 (変換後).

ID	$x_{11}$	$x_{12}$	$x_{13}$	$x_{21}$	$x_{22}$	$x_{23}$	$x_{31}$	$x_{32}$	$x_{41}$	$x_{42}$	$x_{43}$	$x_{44}$	$x_{51}$	$x_{52}$	$x_{61}$	$y$
1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	31
2	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	15
3	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	92
4	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	65
5	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	35
6	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	89
7	1	0	0	0	0	1	0	0	0	0	1	0	1	0	1	79
8	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	32
9	0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	38
10	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	46

表1 データの例 (変換前).

ID	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$y$
1	$a$	$b$	$a$	$d$	$c$	$b$	31
2	$a$	$a$	$c$	$e$	$b$	$b$	15
3	$b$	$c$	$b$	$d$	$a$	$a$	92
4	$c$	$c$	$c$	$b$	$b$	$a$	65
5	$a$	$b$	$b$	$c$	$c$	$b$	35
6	$c$	$a$	$a$	$c$	$c$	$a$	89
7	$b$	$d$	$a$	$d$	$b$	$b$	79
8	$d$	$b$	$a$	$a$	$a$	$b$	32
9	$c$	$a$	$b$	$e$	$b$	$a$	38
10	$a$	$b$	$c$	$a$	$a$	$a$	46

### 3. Random Forests

本章では本研究で用いた Random Forests について説明する<sup>1</sup>. Random Forests とは決定木を多数集めて森を構成する機械学習アルゴリズムであり, 2001 年に Leo Breiman によって提案された [3, 4]. 決定木自体は優れた識別性能を持っている訳ではないが決定木を集め, 個々の決定木の識別結果の平均値の多数決を Random Forest の識別結果とすることで高い識別性能を得る. Random Forest は高速でシンプルな学習を行い, 大規模データにも適用が可能である. 更に学習過程にランダム性を導入していることにより, 高い汎化性能を持つ優れた機械学習アルゴリズムとして知られている.

#### (1) 決定木 (decision tree)

##### a) 決定木の概要

決定木とは機械学習の1つである. また, 機械学習とはある規則に従ってデータを解析し, データの特性などを見出すことにより回帰問題, 分類問題, 文字認識, 画像認証やクラスタリングなど様々な分野に応用を試みるアルゴリズムである. 現在までに様々な機械学習の手法が存在し, 最適化の視点, 確率的な視点, 統計的な視点など様々な視点からのアプローチが行われている [5, 6]. その中でも本項で説明を行う決定木は, 統計的な側面が強い機械学習であり, 回帰問題と分類問題の解析に主に利用される. 決定木はアルゴリズム

<sup>1</sup>英語表記では Random Forests であるが, 日本語では頻繁にランダムフォレストと呼ばれ, 表記されている.

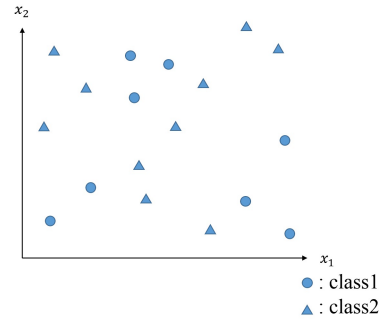


図2 データの例.

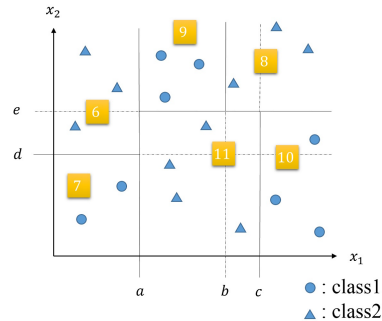


図3 分割されたデータ (実線で分割される).

がシンプルであるので, 他の手法に比べ識別能力は他の手法に劣る事が多いが高速な学習速度とある程度の識別能力が保証されている点の特徴である.

決定木は分岐関数 (devision function) と呼ばれる関数によってクラスラベルが偏る様にデータの分割を行う. この処理によってデータは2つの集合に分けられる. 集合  $S_1$  を分割することによって得られる領域をそれぞれ  $S_1^R, S_1^L$  とし, 同様の処理を  $S_1^R, S_1^L$  に対して行う. このような処理を繰り返す事によって, 最終的に領域には特定の1つのクラスラベル, もしくは1つのクラスラベルのみしか存在しないと許容できるまで分割を繰り返す.

決定木の学習の例を図2と図3に示す. 図2は二次元データの散布図である. 各データは  $x_1$  成分と  $x_2$  成分によって構成されていて class1 もしくは class2 のクラスラベルを持つ.

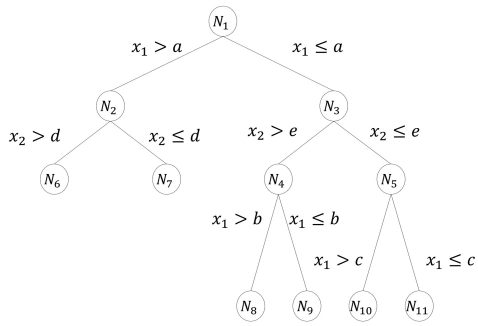


図4 図3の書き換え。

このデータ集合に対して決定木によって学習を行った結果が図3である。データ集合を  $x_1$  軸と  $x_2$  軸の交互から1種類のクラスになる様に分割を行い、その時の分岐関数をそれぞれ  $a, b, c, d, e$  と名付ける。データは分岐関数  $a, b, c, d, e$  によって分割され、領域6から10のいずれかに属する。図3は図4に示す様に樹形図の形に書き換える事ができる。上端にあるノード  $N_1$  を根ノードと呼び、データはまず根ノードに入力される。次にデータの  $x_1$  成分に注目し、 $x_1 > a$  を満たすデータは左に、そうでないデータは右に分岐する。分岐したデータの集合はそれぞれ  $S_2 (= S_1^L)$  と  $S_3 (= S_1^R)$  となりこれに対して同様の処理を繰り返すと図4の末端のノードは  $N_6$  から  $N_{10}$  となる。この6つのノードを末端ノードと呼び、図3の領域6から領域10とこれは同義であるとわかる。また、根ノードから注目しているノードまでのエッジの個数を深さといい、根ノードから末端ノードまでの深さで最も大きいものを最大深さ  $D$  と呼ぶ。例えば図4のノード  $N_2$  の深さは2であり、最大深さ  $D$  は3である

**b) 分岐関数の決定方法**

次に決定木が分岐関数をどのように決定をするか説明する。決定木は分岐関数を決定する際に情報利得という情報量を参照する。情報利得について説明を行う前に事前知識としてまずエントロピーという情報量について説明する。エントロピーとはデータの純度を表現する量であり、

$$H(S) = - \sum_{c_i} p(c_i) \times \log_2 p(c_i), \quad (1)$$

によって定義される。なお、 $S$  はデータ集合、 $c_i$  はノード  $S$  内でのクラスのカテゴリの種類を意味する。従って、 $p(c_i)$  は  $S$  内でのクラス  $c_i$  の割合を意味する。エントロピーはデータのカテゴリが偏っている程低く、均一に存在されている程高くなる特性を持つ。そして情報利得とは分割前と分割後のエントロピーの変化量を意味する量であり、

$$I = H(S) - \sum_{K=L,R} \frac{|S^K|}{|S|} H(S^K), \quad (2)$$

で定義される。ここで  $S^L$  は分岐関数よりも大きい値を取るデータの集合である。つまり分岐関数よりも左側の領域を意味し、 $S^R$  は右側の領域を意味する。式(2)の第二項は分割後の2つの領域  $S^R, S^L$  のエントロピーの重み付きの平均であるので、式(2)はエントロピーの変化量であると理解できる。エントロピーは、クラスのカテゴリが偏っている程低い値を取ることから、情報利得  $I$  は高ければ高いほど効率の良い分割ができていいる事が評価出来る量であると言える。この

ことを踏まえて、ノード  $N_j$  内のデータ集合  $S_j$  の最適な分岐関数の決定方法は以下のように定式化できる。

$$\begin{cases} I(\theta_i) = H(S_j) - \sum_{K=L,R} \frac{|S_j^K(\theta_i)|}{|S_j|} H(S_j^K(\theta_i)), \\ \theta_i^* = \arg \max_{\theta_i \in \tau} I(\theta_i). \end{cases} \quad (3)$$

なお、第1式の第2項の  $S_j^K(\theta_i)$  はノード  $j$  の集合  $S_j$  に対してパラメータ  $\theta_i$  の分岐関数によって分割された集合を意味する。また  $\tau$  は考え得る分岐関数の引き方の集合を表す。

**c) 決定木の予測方法**

次に決定木がデータを学習することによって構成をした識別モデルを用いて、クラスラベルが未知であるデータを、どの様に予測を行うかを説明する。決定木の予測方法は、データのクラスラベルの多数決を取ることによって決定される。例えば今、図3の領域7にクラスラベルが未知のデータが説明変数の入力により判明したとする。図3の領域7はclass1のデータのみが属する領域であるので、この場合、決定木はclass1の満場一致で未知のデータのクラスラベルはclass1であると予測をする。

例で挙げた図3は、それぞれの領域には1つのクラスしか存在しない場合であるが、2種類のクラスが混在する場合も同様に一般化して考えれば良い。領域内でクラスの多数決を取るということは、考えている領域内でのクラスについての事後確率を最大化するクラスであることと同義である。全ての末端ノードは先程述べた各末端ノードにおけるクラスごとの事後確率が対応付けることが出来る。従って決定木は未知のデータがどの領域に属するかが分かれば、対応付けられた事後確率からクラスの予測が可能となる。このことを定式化すると以下ようになる。

$$\begin{cases} P(C_j | \mathbf{v}') = P(C_j | S_t) = \frac{P(C_j, S_t)}{P(S_t)} = \frac{N_j(S_t)}{N(S_t)}, \\ \text{識別クラス} = \arg \max_{c_j} P(C_j | \mathbf{v}'). \end{cases} \quad (4)$$

ただし  $\mathbf{v}'$  はクラスが未知のデータ、 $S_t$  はノード  $t$  番目のノードのデータの集合、 $N_j(S)$  は集合  $S$  内のクラス  $C_j$  であるデータの個数、 $N(S)$  は集合  $S$  のデータの個数を意味する。

次に決定木がどのようなタイミングで学習を終了するかについて説明する。一般に決定木は次の3点の終了条件を設けることが望ましいとされる。

決定木の学習の終了条件

- ・あらかじめ設定した決定木の最大深さ  $D$  に到達したら学習終了。
- ・ノードに割り当てられた学習データの個数が一定個数以下になったら終了。
- ・分割による情報利得が一定値以下になったら終了。

これはデータを必要以上に分割してしまうことが起因となる、過学習という現象を避ける事が目的である。過学習とは学習を行ったデータに対してはクラスの予測を正しく出来るが、未知のデータに対しては正しく予測が出来ない状況を意味し、汎化能力不足が原因である。上記の終了条件の各パラメータ等の設定は経験的な設定に頼る必要があるが、この終

了条件によって敢えて早期に学習を打ち止めることにより、汎化性能を確保し未知のデータに対しても正しい予測が出来るようにする。

## (2) Random Forests

### a) Random Forests の概要

Random Forests とは 3.1 節で説明した決定木を大量に組み合わせることで、識別モデルを構成する機械学習の手法である。Random Forests は決定木に並列学習をさせることで識別モデルを大量に作成し、各決定木に未知データを与える事で分類結果を集計する。そして集計したものの中で最も投票の多かったクラスが Random Forests の識別結果となる。この様に、決定木同様に Random Forests も非常にシンプルな構造をしており、大規模データに対しても高速な学習が可能である点の特徴である。また、Random Forests は学習の過程で 2 点のランダム性を導入している。これによって決定木同士の相関を下げ、高い精度を実現している。

### b) Random Forests の学習

図 5 に Random Forests の学習過程のイメージを示す。まず、Random Forests は自身を構成する  $T$  個の決定木  $T_1, T_2, \dots, T_T$  に学習データを与える。3.1 節では決定木は学習データをそのまま学習していたが、Random Forest の場合は異なる。Random Forests の場合、構成する決定木  $T_1, T_2, \dots, T_1, \dots, T_T$  にはデータ集合  $S$  からブートストラップサンプリングされた部分集合  $S_1, S_2, \dots, S_1, \dots, S_T$  が学習データとして与えられる。図 6 に Random Forests のサンプリングのイメージを示した。サンプリングの際、各部分集合の大きさは全て同じでなければならない。また、部分集合間であれば、データが重複してサンプリングされるのは許容され、どの部分集合にもサンプリングされていないデータがあっても許容される。各々の決定木  $T_i$  は与えられたデータ集合  $S_i$  に対して 3.1 節で述べた様に学習を行うが Random Forests を構成する決定木の場合、1 点異なる点がある。

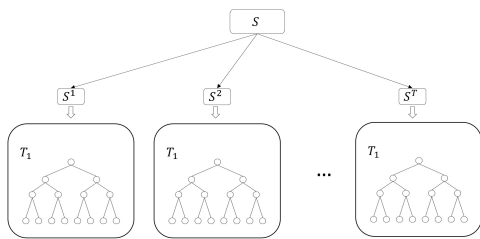


図 5 Random Forests の学習のイメージ。

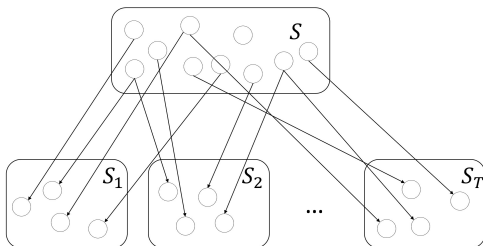


図 6 Random Forests のサンプリングのイメージ。

式 (4) の第 2 式で  $\theta_i$  の候補の集合は  $\tau$  であり、考えうる全ての場合の集合であるとしていた。しかし、Random Forests の場合は各ノード  $N_j$  ごとに部分集合  $\tau_j \in \tau$  をランダムに決定し、 $\tau_j$  の中から最適なパラメータ  $\theta_j^*$  を選ぶ。なお、このランダム性は

$$\rho = \frac{|\tau_j|}{|\tau|}, \quad (5)$$

によって制御する事が可能である。そして各決定木の末端ノードには各クラスごとの末端ノード  $N_j$  における各クラスの事後確率  $P_i(c_j|S_j)$  が対応付けられ、Random Forest の学習は終了する。3.2.a 項で Random Forests は二点のランダム性が導入されていると述べたが、この二点とは (1) データ集合からブートストラップサンプリングをしている点、(2) 各分岐で分岐関数のパラメータの候補をランダムに選んでいる点のことである。このランダム性が相補的に働く事で、決定木同士の相関を下げ、分散を低減している。

### c) Random Forests の予測

次に Random Forests の予測方法について説明する。Random Forests の予測の過程を図 7 に示す。クラスラベルが未知のデータ  $v'$  とすると、テストデータ  $v'$  はまず  $T_1, T_2, \dots, T_T$  の  $T$  本の決定木の根ノードに入力される。テストデータ  $v'$  はそれぞれの木のノードの分岐条件に従って分岐を繰り返していき、それぞれの木のいずれかの末端ノードに振り分けられる。  $t$  番目の決定木  $T_t$  の末端ノードにおけるテストデータ  $v'$  のクラスの事後確率は、3.1 節と同様に  $P_t(c_j|v')$  となる。Random Forests の予測では、全ての決定木のクラスごとの未知データの事後確率を計算し、事後確率の相加平均を考える。そして、平均化されたクラスごとの事後確率  $P(c_j|v')$  を比較し、 $P(c_j|v')$  が最も大きかったクラスが Random Forests の予測したクラスとなる。このことを定式化すると以下の様になる。

$$\begin{cases} P(c_j|v') = \frac{1}{T} \sum_{t=1}^T P_t(c_j|v'), \\ \text{識別クラス} = \arg \max_{c_j} P(c_j|v'). \end{cases} \quad (6)$$

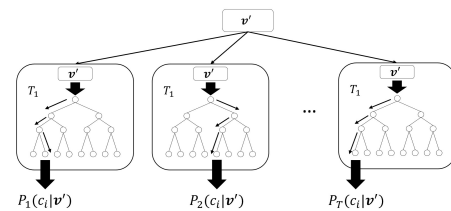


図 7 Random Forests の予測のイメージ。

### (3) Random Forests の重要度

本節では Random Forests の変数重要度について説明をする。Random Forests は学習の過程で、各変数が目的変数の予測にどれくらい重要であるかを数量的に評価した量である変数重要度を算出することが可能である [3, 7]。

変数重要度を得ることにより、データの規模が大きい場合、膨大な説明変数の中から有用である説明変数と有用でない説明変数の分別ができる。また、重要度の低い変数を削除をすることで、計算時間の削減にも繋がる。Random Forests の重要

度の算出方法にはジニ係数やエントロピーなどの不純度を用いて算出する方法 (Mean Decrease Impurity; MDI) と、予測精度の変化量によって算出する方法 (Mean Decrease Accuracy; MDA) がある。以降、それぞれを MDI, MDA と省略して呼ぶことにする。

#### a) MDI (Mean Decrease Impurity)

MDI は Random Forests を構成する決定木が一回あたりの分岐でデータの不純度をどれくらい下げることが出来るかを計算することにより算出される重要度である。3.1.b 項ではデータ集合の不純度を評価するのにエントロピーを用いたが、エントロピーも用いて重要度を算出する方法は特許が取得されており、そこで代替としてエントロピーと似た振る舞いを持つジニ係数 (Gini Index) を不純度を測る量として用いる事が一般的である。ノード  $S$  で  $K$  種類のクラス  $c_k$  を持つデータのジニ係数は

$$H(S) = 1 - \sum_{k=1}^K P^2(c_k), \quad (7)$$

と与えられる。

説明変数  $x_j$  の変数重要度を MDI によって求める場合、Random Forests は説明変数  $x_j$  のデータ  $v_{ij}$  とクラスラベル  $t_i = \{c_k | 1 \leq k \leq K\}$  によって構成されたデータ  $\mathbf{v}_i = (v_{ij} \ t_i)$  ( $i = 1, 2, \dots, N$ ) を学習する。この時  $t = 1, 2, \dots, T$  に対して  $t$  番目の決定木についての  $x_j$  の重要度を  $MDIImp_t(x_j)$  とすると、一回の分岐あたりの不純度の変化量は

$$MDIImp_t(x_j) = \sum_{i \in M_t \setminus \tilde{M}_t} \frac{I_i(\theta^*)}{|M_t \setminus \tilde{M}_t|}, \quad (8)$$

である。ただし、 $M_t, \tilde{M}_t$  はそれぞれ  $t$  番目の決定木の根ノードから末端ノードまでの通し番号の集合と末端ノードの番号の集合を意味している。また、 $I_i(\theta^*)$  は  $i$  番目での分割での情報利得 (式 (2)) を意味している。よって  $x_j$  の変数重要度を  $MDIImp(x_j)$  とすると、Random Forest 全体を考えればよいので

$$MDIImp(x_j) = \frac{1}{T} \sum_{t=1}^T Imp_t(x_j), \quad (9)$$

となる。この処理をすべての説明変数に対して行うことで MDI によって各説明変数の重要度が算出できる。

#### b) MDA (Mean Decrease Accuracy)

前述のとおり、Random Forests は学習の時に各決定木にデータ集合  $S$  からブートストラップサンプリングによって得たサブセットを与え、学習をさせる。この時、データ集合  $S$  の全体の約 3 分の 1 に相当するデータはサンプリングされない。このサンプリングされなかったデータを OOB (Out-Of-Bug) データと呼ぶ。予測精度による変数重要度はこの OOB データを用いることによって得られる。図 8 に予測精度による変数重要度の算出方法のイメージを示す。  $i = 1, 2, \dots, \tilde{N}$  に対して各変数名が  $x_1, x_2, \dots, x_N$  と名付けられた  $N$  列の説明変数とクラスラベル  $t_i = (c_k | 1 \leq k \leq K)$  を合わせたベクトル  $\tilde{\mathbf{v}}_i = (v_{i1} \ v_{i2} \ \dots \ v_{iN})$  を OOB データとする。まず  $\tilde{N}$  個の OOB データをクラスラベルを伏せた状態で 1 つずつ Random Forests の構築した識別モデルに予測をさせ、予測したクラスと実際のクラスを照らし合わせることで正答率を算出する。この正答率を  $\alpha$  とする。次に  $j$  番目の変数名  $x_j$  の変数重要度を求める場合には  $j$  番目のデータ  $v_{1j}, v_{2j}, \dots, v_{\tilde{N}j}$  をランダムに入れ替える。入れ替えることによって得た新たな

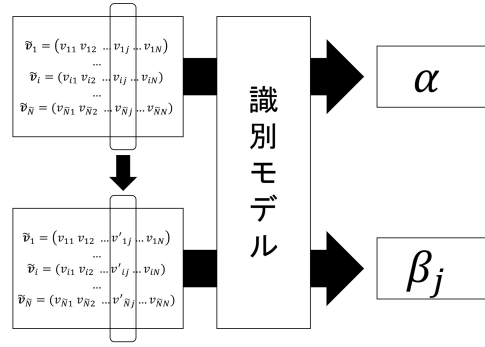


図 8 MDA の算出方法。

ベクトルを  $\mathbf{v}'_i$  とし、入れ替えられた要素を  $v_{ij}$  として識別モデルに同様に予測をさせ、正答率を  $\beta_j$  とする。以上より変数  $x_j$  の予測精度による変数重要度を  $MDAImp(x_j)$  とすると、2 つの正答率を用いて

$$MDAImp(x_j) = \beta_j - \alpha, \quad (10)$$

で得られる。これは正規の変数  $x_j$  と入れ替えることによって意味の無い変数  $x'_j$  の正答率の変化量と解釈できる。よって式 (10) は高い値ほど目的変数に影響が大きい値をとり、影響が無いものは 0、予測低下に繋がる不要な変数であれば負の値を取る量である。なお、本研究ではダミー変数変換を行っているため、変数  $x_j$  のダミー変数ごとに変数重要度を算出し、平均値を取ることで、 $x_j$  の変数重要度としている。

## 4. 分析

本章では 2 章で説明をしたデータセット A, データセット B に対して分析を行った結果をまとめる。分析に用いる手法は Random Forests と比較手法としてロジスティック回帰を用いた。分析では両手法の正答率、第一種の過誤の割合、第二種の過誤の割合、そして変数重要度の算出を行い評価を行った。なお、ダミー変数変換を行った後のデータセット A とデータセットと B の大きさはそれぞれ  $405 \times 115$ ,  $258 \times 170$  である。また、ロジスティック回帰の分析を行った際、予測ができないデータが数件存在したので除外をした。従って、Random Forests とロジスティック回帰の学習を行ったデータのプロジェクト数は異なることに注意されたい。

### (1) 各パラメータの設定

本章で扱う Random Forests の各パラメータの設定は表 3 のように設定を行った。

引数	説明	値
ntree	決定木の個数	500
mtry	各分岐で評価に用いる変数の数	$\sqrt{m}$ $m$ : 変数の数

### (2) 変数選択

変数選択とは予測に用意されている全ての変数を使わずに目的変数の予測に有用であると考えられる説明変数変数のみを用いてモデルを構成する方法である。説明変数が多い場合、変数選択によってデータの大きさを小さくすることにより、モデルの解釈が容易になるメリットがある。そこで本研究は

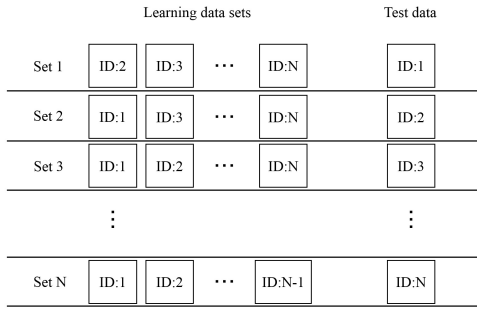


図9 leave-one-out 法の概要.

表4 予測結果と真値の対応表.

		真値	
		Bug	NonBug
予測	Bug	$N_1$	$N_2$
	NonBug	$N_3$	$N_4$

両手法とも leave-one-out 法の試行一回ごと変数選択を行った後、識別モデルを構築し、予測を行った。Random Forests の場合、MDI もしくは MDA を用いて変数重要度が上位 3 分の 2, 上位 2 分の 1, 上位 3 分の 1 の 3 パターンの説明変数を用いて予測を行った。ロジスティック回帰の場合、一般的な変数選択手法として知られている変数増減法 (stepwise forward selection method) を用いて予測を行った。

### (3) 正答率, 第一種, 第二種の過誤の割合

まず、両手法の精度を評価するために正答率, 第一種, 第二種の過誤の割合の算出を行う。本研究では leave-one-out 法を用いて算出を行った。leave-one-out 法とは、図9に示すようにデータセットの中の1つのデータを予測を行うためのテストデータに設定し、残りのデータを学習することによりテストデータを予測をする。この処理を全てのデータがテストデータになるようにデータの個数分繰り返す、予測結果を得る方法である。データセットのデータ数を  $N$  として予測結果と真値の対応を表4のように設定すると正答率, 第一種, 第二種の過誤の割合は式 (11), 式 (12), 式 (13) のように定義される。

$$\text{正答率} = \frac{N_1 + N_4}{N}. \quad (11)$$

$$\text{第一種の過誤の割合} \alpha = \frac{N_2}{N}. \quad (12)$$

$$\text{第二種の過誤の割合} \beta = \frac{N_3}{N}. \quad (13)$$

以下に分析結果をまとめる。

表5 データセット A における両手法の予測結果 (小数点第二位を四捨五入).

Random Forests			ロジスティック回帰		
正答率	$\alpha$	$\beta$	正答率	$\alpha$	$\beta$
75.3%	13.6%	11.1%	65.4%	16.2%	18.4%

表6 データセット B における両手法の予測結果 (小数点第二位を四捨五入).

Random Forests			ロジスティック回帰		
正答率	$\alpha$	$\beta$	正答率	$\alpha$	$\beta$
72.5%	18.2%	9.3%	66.4%	17.4%	16.2%

表7 データセット A における MDA によって変数選択した場合の Random Forests の予測結果 (小数点第二位を四捨五入).

選択数	正答率	$\alpha$	$\beta$
変数選択なし	75.3%	13.6%	11.1%
上位 3 分の 2	76.0%	12.6%	11.4%
上位 2 分の 1	74.8%	14.6%	10.6%
上位 3 分の 1	73.5%	16.0%	10.4%

表8 データセット A における MDI によって変数選択した場合の Random Forests の予測結果 (小数点第二位を四捨五入).

選択数	正答率	$\alpha$	$\beta$
変数選択なし	74.8%	13.8%	11.4%
上位 3 分の 2	75.8%	12.6%	11.6%
上位 2 分の 1	76.0%	14.3%	9.6%
上位 3 分の 1	73.4%	15.3%	10.9%

表9 データセット B における MDA によって変数選択した場合の Random Forests の予測結果 (小数点第二位を四捨五入).

選択数	正答率	$\alpha$	$\beta$
変数選択なし	70.5%	19.0%	10.5%
上位 3 分の 2	71.3%	18.6%	10.1%
上位 2 分の 1	72.5%	19.0%	8.5%
上位 3 分の 1	69.8%	22.5%	7.8%

表10 データセット B における MDI によって変数選択した場合の Random Forests の予測結果 (小数点第二位を四捨五入).

選択数	正答率	$\alpha$	$\beta$
変数選択なし	70.5%	19.8%	9.7%
上位 3 分の 2	73.6%	18.6%	7.8%
上位 2 分の 1	70.5%	20.5%	8.9%
上位 3 分の 1	70.2%	21.3%	8.5%

表5, 表6 はデータセット A とデータセット B に対して Random Forests とロジスティック回帰で分析した結果の正答率, 第一種の過誤の割合  $\alpha$ , 第二種の過誤の割合  $\beta$  の値である。両データに対しても僅かではあるが、正答率の上昇を確認することが出来た。また、一般に好ましくない事象と考えられている第二種の過誤の割合に関しても Random Forests は第一種の過誤の割合よりも低い値に収まったが、ロジス



ティック回帰では第二種の過誤の割合が第一種の過誤の割合を上回った。表 7, 表 8, 表 9, 表 10 は Random Forests の変数重要度から上位の変数のみを採用し, 学習を行った結果である<sup>2</sup>。選択数は経験的ではあるが, 変数選択を行った方が, 行わない場合よりも正答率は高く, 第二種の過誤の割合  $\beta$  は低くなることが確認出来た。

## 5. 考察

Random Forests の予測結果がデータセット A, データセット B 共にロジスティック回帰の予測結果を上回る事が確認できた。これは Random Forests の特徴である 2 つのランダム性がうまくデータに作用し, Random Forests を構成する決定木同士の相関が下がる事で高い汎化性能を得ることが起因であると考えられる。またソフトウェア開発において「リリース一ヶ月以内に不具合の発生は無いと予測したが, 実際は不具合が発生してしまった」という事象である第二種の過誤の割合  $\beta$  が, 「リリース一ヶ月以内に不具合が発生すると予測したが, 実際は不具合が発生しなかった」という事象である第一種の過誤の割合  $\alpha$  を Random Forests においては下回る事が確認出来た。 $\alpha$  と  $\beta$  はトレード・オフの関係から正答率を上げない限り, 両者を減少させることは出来ない。従ってソフトウェア開発においては  $\alpha > \beta$  の関係が成立している事が好ましいので, ソフトウェア開発における品質予測問題において新たなモデルが提案出来たと考えられる。更に, 用いるデータセット, 変数重要度を算出する基準 (MDI もしくは MDA) によって変数の選択数は異なるが, Random Forests は変数選択を行わない場合よりも行った場合の方が, 精度の上昇が見込める事が確認できた。これは目的変数に効果の大きい説明変数のみで学習を行う事で, Random Forests を構成する決定木が目的変数に効果の薄い説明変数を分割の際に選出する可能性が排除され, 結果として精度の上昇につながったと考えられる。よって, 変数重要度を考慮し, 変数選択を行った Random Forests は質的変数によって構成された 2 値分類問題の解析手法として有用であると考えられる。

## 6. おわりに

本研究では質的変数によって構成されたデータセットの目的変数の予測を行った。本来は質的変数は量的変数と同等に扱うことは出来ないが, ダミー変数変換という処理をデータセットに対して行うことにより量的変数と同等に扱う事が可能にした。ダミー変数変換を行ったデータセットに対し, 本研究では機械学習の 1 つである Random Forests と一般的な 2 値分類手法として知られてるロジスティック回帰で分析を行った。結果として, Random Forests は精度や好ましくない事象と考えられる第二種の過誤の割合  $\beta$  に関してロジスティック回帰よりも優れた値を得た。このことにより, ソフトウェア開発におけるテスト工程よりも前の段階で得られる質的データをもとに, ソフトウェアがリリース後一か月以内に不具合が発生するかどうかを予測することも, また有用であることを示した。

<sup>2</sup>なお, 表 7 と表 8 の変数選択無しの場合と表 5 の Random Forests, 表 9 と表 10 の変数選択無しの場合と表 6 の Random Forests は同じ環境で分析を行ったが, Random Forests はランダム性を含んでいるため数値に違いがあることに注意されたい。

## 謝辞

データを提供して頂いた独立行政法人情報処理推進機構ソフトウェア高信頼化センターに対して謝意を表します。

## 参考文献

- [1] M. Kimura and R. Shimada: "A note on static software reliability models by GMDH: comparison with a multiple regression model", Proc. 18th IEEE Pacific Rim Intern. Symp. on Dependable Computing (PRDC 2012), Niigata, Japan, CD-ROM (2012).
- [2] 独立行政法人情報処理推進機構 技術本部ソフトウェア・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 2012-2013, 独立行政法人情報処理推進機構 (2012).
- [3] L. Breiman: "Random forests", Machine Learning, Vol. 45, Issue 1, pp. 5-32, DOI: 10.1023/A:10109334043 24 (2001).
- [4] 下川敏雄, 杉本知之, 後藤昌司, R で学ぶデータサイエンス 9 樹木構造接近法, 共立出版 (2013).
- [5] C. M. ビショップ, パターン認識と機械学習 上, 丸善出版 (2012).
- [6] C. M. ビショップ, パターン認識と機械学習 下, 丸善出版 (2012).
- [7] G. Louppe, I. Wehenkel, A. Sutera, P. Geurts: "Understanding variable importances in forests of randomized trees", Advances in Neural Information Processing Systems 26 (NIPS 2013).