

卒業研究報告書

題目 組み合わせ不具合特定のための
ロジスティック回帰を用いた分析

指導教員 水野 修 准教授

京都工芸繊維大学 工芸科学部 情報工学課程

学生番号 12122038

氏名 西浦 生成

平成28年2月15日提出

組み合わせ不具合特定のためのロジスティック回帰を用いた分析

平成 28 年 2 月 15 日

12122038 西浦 生成

概 要

ソフトウェア開発においてテストは重要な役割を果たす。テストの一種である組み合わせテストはパラメータ間の相互作用による不具合を検出するのに有効だが、組み合わせが大きくなると網羅的にテストを行うことが困難になるため未検出の不具合が残る可能性があるという欠点がある。

そこで本研究では、実際に行われたテストを統計的に分析調査することで、ある次数の不具合が含まれる可能性の高い組み合わせをそれより低い次数の組み合わせテスト結果から予測する手段を発見し、実際にテストする組み合わせの数を減らすことで、組み合わせ爆発を回避してより効率的に広範囲の組み合わせテストを行うことを目的とする。分析にはロジスティック回帰を利用し、1 から 4 の t に対して t -wise 組み合わせのテスト結果への影響度を計算して、ある t -wise 組み合わせとそれをサブセットに含むより大きな次数の組み合わせの影響度の間にかしらの相関や傾向が見られるかを調査する。

本研究ではこうした分析の結果、ある組み合わせをサブセットに含む上位次数の組み合わせにいくつかの傾向を発見し、それが不具合を含む組み合わせの予測に有用であることを示すことで、より効率的で効果的なテスト手法の可能性を示した。

目 次

1. 緒言	1
2. 研究の目的	2
3. 準備	3
3.1 組み合わせテスト	3
3.2 Fault localization	3
3.3 ロジスティック回帰	3
4. 分析	5
4.1 分析の対象	5
4.2 分析の準備	5
4.2.1 従属変数の付与	5
4.2.2 不必要なバージョンの除外	6
4.2.3 4-wise までの組み合わせ存在表の作成	6
4.2.4 組み合わせの削減	6
4.2.5 ロジスティック回帰分析の実行	7
4.3 分析の基本的な理念	7
4.4 組み合わせの属性の決定	7
4.5 分析 1	8
4.6 分析 2	8
4.7 分析 3	8
5. 結果	9
5.1 実験結果 1	9
5.2 実験結果 2	13
5.3 実験結果 3	16
6. 考察	18
6.1 分析の有効性	18
6.2 分析の妥当性	19

6.2.1 外的妥当性	19
6.2.2 内の妥当性	19
7. 結言	20
謝辞	20

1. 緒言

リリースしてしまったソフトウェアに不具合が見つかった場合，その修正には多大な労力が必要となり，また市場に普及したソフトウェアを漏れなく修正できる保証はない．ユーザーは不具合による不利益や面倒を被り，そのソフトウェアは低品質だという烙印を押され，開発者の信用は失墜する．ソフトウェアの品質を保つため，リリース前になるべく多くの不具合を検出するソフトウェアテストは必要不可欠である．

ソフトウェアテストの一つの方法として，単体テストのみでは発見できない欠陥，つまりパラメータの相互作用による欠陥を発見する組み合わせテストがある．完璧な組み合わせテストを求めるならば全ての組み合わせをテストすればよいが，規模の大きなソフトウェアの場合には存在する全ての組み合わせを網羅してテストすることは現実的でない．従来の組み合わせテスト手法では主に2から4程度に組み合わせ上限を定め，All-pair法や直交表による方法を開発し，不具合発見率とテスト労力の適度なトレードオフを実現してきた．だが，それ以上の組み合わせの間にも致命的な不具合が潜む可能性を完全には排除できない．

この問題解決へのインプレッションは，例えば二因子間までのテスト結果の情報から，三因子間で不具合のありそうな組み合わせを予想できることが出来れば，三因子間で実際にテストする組み合わせの数を絞り込むことができるのではないかという考えに基づく．この上位次数の組み合わせの絞り込みを一般化できれば，理論的にはどれだけ大きな次数の組み合わせに対しても不具合特定を行うことが可能に，あるいは容易になる．

そこで本研究では，上位組み合わせ次数での組み合わせ因子の絞りこみを利用した組み合わせテスト手法開発の足掛かりとなる，上位組み合わせ次数の絞り込みに役立つような情報の分析を，ロジスティック回帰を利用した方法で行う．

本報告書の構成を以下に示す．2章では本研究の目的を，3章では分析に必要な事前知識の説明を，4章では分析対象および分析準備を含む分析の方法を述べる．5章に分析の結果を示し，6章で分析結果の考察を行い，7章にて結言とする．

2. 研究の目的

本研究の目的はロジスティック回帰を用いて実際に行われたテストを統計的に分析調査し、不具合を含む組み合わせの予測に有用な分析結果を得ることである。

この目的のため、以下の研究設問を設定し、分析手法と分析結果について考察する。本研究における研究設問は以下の通りである。

RQ1: パラメータ組み合わせのテスト結果への影響度を表すロジスティック回帰モデルは構築できるか。

RQ2: 明確な傾向として分析結果は得られるか。

RQ3: 得られた分析結果は不具合を含む上位組み合わせの予測に有用か。

3. 準備

本章では分析を行う前に，分析の前提となる事項を紹介する．

3.1 組み合わせテスト

複数の環境や入力をパラメータとして組み合わせでソフトウェアの動作を確認するテスト方法で，単体テストでは見つけられない欠陥，つまりパラメータの相互作用による欠陥を発見できる．複数パラメータの相互作用による不具合検出に使用する効率的なテストケースを生成する方法や，生成されたテストケース自体を組み合わせテストと呼ぶ場合もあるが，本報告書では不具合検出手法自体のみを指して組み合わせテストと呼称する．組み合わせ数を t としたときの組み合わせテストを t -way テストや t -wise テストと呼び，本報告書では t -wise テストに呼称を統一する．

3.2 Fault localization

組み合わせテストの結果から，どの因子の組み合わせがテスト結果の不具合に影響するかを割り出し，問題のある組み合わせ (Faulty Interaction) を特定すること．

3.3 ロジスティック回帰

ロジスティック回帰は一般化線形モデルを用いた多変量解析を行う重回帰の一種である．回帰式は次の式で表される．

$$p = \frac{1}{1 + \exp(-b_0 - b_1x_1 - \dots - b_px_p)} \quad (3.1)$$

ここで， p は従属変数， $x_1 \sim x_p$ は独立変数であり， $b_1 \sim b_p$ は各独立変数の回帰係数， b_0 は回帰定数である．

はじめに回帰分析とは，独立変数と従属変数の間の関係を統計的手法によって式の形に推計する方法である．本研究ではテストに使用するパラメータが独立変数であり，テスト結果が従属変数となる．独立変数が単一の場合は単回帰となり，複数

ある場合は重回帰となる．本研究では複数のパラメータを入力としているので重回帰となる．

線形回帰分析では回帰直線を求めるのが一般的だが，従属変数が例えば「ある」「ない」といった2値で表される場合，直線への当てはまりが極めて悪くなり，正しい回帰係数を求めることが困難になる．そこでこのような場合は0と1に漸近するロジスティック曲線に回帰させることでうまく回帰係数を求められるようになる．本研究では従属変数であるテスト結果が「成功」あるいは「失敗」という2値であるので，ロジスティック回帰を利用できる．

ロジスティック回帰によって得られる独立変数ごとの回帰係数は，その独立変数が従属変数に与える影響の大きさを表すことになる．独立変数 x_i の回帰係数 b_i が別の独立変数 x_j の回帰係数 b_j より大きければ，独立変数 x_i の変位による従属変数への影響度は独立変数 x_j のものより大きいということが言える（影響度を定量的に比較するには回帰係数のオッズ比を取る必要がある．）また統計処理ツールであるRでロジスティック回帰分析を行った場合，係数値にNAという値が出ることがあるが，これは係数値0と同値で，その独立変数は従属変数に一切関係ないと推計されたということである．

4. 分析

本章では、分析の対象・準備・方法について説明する。

本研究では、 t -wise($t = 1, 2, 3, 4$) のパラメータの組み合わせに対してロジスティック回帰分析を行い、その回帰係数をもって t -wise の各ペアの属性とし、主に n -wise ペアの回帰係数と $(n+1)$ -wise ペアの回帰係数の変化を観測して集計する。

4.1 分析の対象

本研究の分析は、産業技術総合研究所から提供された次のファイルを対象とした。

- 243 のソフトウェア・バージョンに適用するテストスイート
- 各テストスイートのテスト結果

テストスイートは1つのバージョンに対するテストケースの集合である。テストスイートに含まれるテストケース数はバージョンによって異なるが、500 程度である。テストケースは一回のテストに使用する入力パラメータ群の集合である。テストスイートは全てのパラメータの組み合わせが存在するように作られている。

テスト結果の情報は、各バージョン内のテストケースに対して pass あるいは fail として記されたものである。

4.2 分析の準備

分析を行うにあたり、テストスイートとそのテスト結果から 4-wise までの組み合わせを抽出し、ロジスティック回帰分析を行って組み合わせごとに回帰係数を導出しておく。また、導出された回帰係数から、その組み合わせがテストの失敗に寄与する影響度の大きさを組み合わせを分類する。

4.2.1 従属変数の付与

テストスイートファイルのテストケースの末尾に、そのテストケースが失敗したか成功したのかを表す変数 (result 変数) を追加したファイルを作成する。不具合原因の組み合わせの回帰係数を正とするために、result 変数は fail なら 1、pass なら 0 とする。

4.2.2 不必要なバージョンの除外

全てのテストケースが pass あるいは全て fail であるバージョンは実験の趣旨に沿わないため除外する。全てのテストに合格したソフトウェアはつまり完成しているので Fault Localization を行う必要がなく、全てのテストに失敗するソフトウェアはそもそもテスト段階に到達していない。除外を行った結果、実験に使用できるバージョンは 64 個になった。

4.2.3 4-wise までの組み合わせ存在表の作成

各バージョンに対し、各テストケースを見出し、テストスイート全体において出現した n -wise の組み合わせを列見出しとした表を想定し、その列の組み合わせがその行のテストケースに含まれていれば 1、そうでなければ 0 で表したファイルを csv で作成する。これは組み合わせの存在をダミー変数で表したものであり、ロジスティック回帰においてはこのダミー変数群が独立変数の役割を果たす。また末尾の列には従属変数である result 変数を付け加える。これを 1-wise から 4-wise にかけて行う。

4.2.4 組み合わせの削減

まず各バージョンに対し 1-wise 存在表でのロジスティック回帰を行い、回帰係数が NA だった 1-wise パラメータを列挙する。次にそのバージョンの 1-wise から 4-wise の組み合わせ存在表の中から、回帰係数が NA だった 1-wise パラメータをサブセットに含む組み合わせの列を削除する。こうすることで組み合わせを大幅に削減できる。

本来であればこの工程は不必要であり、正確な分析の妨げになる行為である。しかし、ロジスティック回帰分析は独立変数の数が増えると処理の回数=処理にかかる時間が指数的に大きくなるため、組み合わせ数を減らさずに 3-wise 以上の存在表をロジスティック回帰分析に投入すると回帰分析完了まで途方もない時間がかかり、実験どころではない。

組み合わせ削減の方法としてこの処理を採用したのは、(1) 回帰係数が NA だったパラメータを独立変数から抜いて再度ロジスティック回帰分析をしても、他の独立変数の回帰係数に影響はないこと (2) 回帰係数が NA であるパラメータを含む組み

合わせはほぼ全て NA になることが、組み合わせ削減を行わない 1-wise:2-wise 間で認められたこと、を根拠にしている。

しかし回帰係数 NA を含む組み合わせの回帰係数が完全に全て NA にならない以上、回帰係数 NA を含む組み合わせを全て削除することによる分析結果への多少の影響はあるだろう。なのでこの実験は厳密には「1-wise の時点でロジスティック回帰の回帰係数が NA だったものを予め省いたケースセットを実験対象とする実験」と定義される。

4.2.5 ロジスティック回帰分析の実行

各バージョンの t-wise 組み合わせ存在表に対して result 変数を従属変数としたロジスティック回帰分析を行い、各 t-wise ペアの回帰係数を決定する。また組み合わせとそれに対応する回帰係数の表を作成し csv ファイル等で作成しておく。これを 1-wise から 4-wise にかけて行う。本研究では統計処理ソフトウェアである R を用いて回帰分析を行った。

ここまでの準備で、各バージョンに対し t-wise(t=1,2,3,4) の組み合わせがテスト結果に寄与する影響度を、回帰係数という形で求めることができた。以降の分析はこの回帰係数を用いて行う。

4.3 分析の基本的な理念

4.4 組み合わせの属性の決定

次に行う分析を円滑に進めるため、得られた回帰係数からその組み合わせがテスト結果に寄与する影響度を BIG, SMALL, NA の 3 つの属性に分類する。

まず係数値 NA のものを属性 NA に分類する。

次に BIG と SMALL に分類するため、回帰係数の大小を区切る border を決める。border は各バージョンの各 t-wise に対してひとつ求められる。border の導出方法を次に示す。

1. 回帰係数の絶対値をソートしたものを $A = \{a_1, a_2, \dots, a_n\}$ とする。 $b_k = \frac{a_{k-1}}{a_k}$ と

して $B = \{b_1, b_2, \dots, b_{n-1}\}$ とする . $\max(B)$ と $\max(B) = b_k$ を満たす k を求め ,
border 候補値として $\frac{a_k + a_{k+1}}{2}$ を求める .

2. $\max(B)$ が 10 に満たない場合 , border=1 とする .
3. border 候補値が 1 に満たない場合 , border=1 とする .
4. 2,3 に当てはまらない場合 , border 候補値を border とする .

端的に言えば , 基本的に border は 1 だが , 係数値が大きく上に外れた組み合わせのグループが存在する場合はそれらとそれ以外を分断するような border になる .

回帰係数の絶対値が border 以上のものを BIG 属性 , border 未満のものを SMALL 属性に分類する . 各バージョン , 各 t-wise に対して組み合わせ , 回帰係数 , 属性の情報が記されたファイルが作成されれば , それらが以降に示す分析の素材となる .

4.5 分析 1

ある属性を持つ 1-wise パラメータがまたある属性を持つ 2-wise ペアのサブセットになっている場合 , その 2-wise ペアは 1-wise パラメータにまた別の 1-wise パラメータが付加され , 組み合わせの次元が上がったと見ることができる . その際 , 1-wise ペアの持つ属性が 2-wise ペアの持つ属性に遷移した , という言い方をすることにする .

ある属性の n-wise ペアがある属性の (n+1)wise ペアに遷移した数を集計する . また , 属性の遷移率を求める .

4.6 分析 2

ある n-wise の組み合わせが (n+1)wise の組み合わせへと遷移するとき , n-wise ペアに 1-wise パラメータが付加されていると言える . その遷移元の属性と付加される 1-wise パラメータの属性の組み合わせについて , 遷移先の (n+1)wise の属性はどのようになるか調べる .

4.7 分析 3

ある n-wise ペアを 1-wise パラメータ n 個の集合としたとき , 要素である 1-wise パラメータの属性の組み合わせのパターンと n-wise ペアの属性の関係を調べる .

5. 結果

本章では，分析によって得られた結果を示す．

5.1 実験結果 1

[4.5 分析 1] によって得られた集計結果を表 5.1～表 5.7 に示す．

表 5.1 について説明する．交絡に示す数字は，左に示す 1-wise パラメータの属性が上に示す 2-wise ペアの属性に遷移したケースの個数を表す．例えば項 (BIG, BIG) の 1973 という数字は，BIG という属性を持っていた 1-wise パラメータが BIG という属性を持つ 2-wise ペアに遷移した個数は 1973 個あったという意味である．

同様に，表 5.2 は 2-wise から 3-wise への遷移，表 5.3 は 3-wise から 4-wise への遷移についての集計である．

表 5.4 について説明する．表 5.1 における遷移元である，ある属性を持つ 1-wise パラメータのなかで，またある属性を持つ 2-wise ペアに遷移した数を内訳として表したものが表 5.4 である．例えば BIG の属性を持っていた 1-wise パラメータの内 42.16% が BIG の属性を持つ 2-wise ペアに，31.74% が SMALL の属性を持つ 2-wise ペアに，26.11% が NA の属性を持つ 2-wise ペアに遷移したことが示されている．この遷移の内訳の割合を遷移率と呼ぶことにする．

また同様に，表 5.5 は 2-wise から 3-wise への遷移率，表 5.6 は 3-wise から 4-wise への遷移率の表である．

表 5.7 は表 5.4，表 5.5，表 5.6 の遷移率の平均をとったものである．ただし，表 5.4 の遷移元属性 NA はサンプル数 0 ゆえの 0% なので計算には含めていない．この表からは，組み合わせ次数が一つ上がる際の一般的な属性遷移の特性が読み取れる．

この結果から読み取れる事柄を以下に示す．

- 表 5.7 から，NA の属性を持つ t -wise ペアは NA の属性を持つ $(t+1)$ -wise に遷移する割合が非常に高い．
- 表 5.7 から，BIG の属性を持つ t -wise ペアは BIG へ，SMALL の属性を持つ t -wise ペアは SMALL への遷移割合が比較的高い．
- 表 5.7 から，BIG と SMALL の t -wise ペアからの NA への遷移率はあまり変わら

ない。

- 表 5.4・表 5.5・表 5.6 から，どの wise 間の遷移も大きな違いはない。しいて言えば，BIG 属性から BIG 属性への遷移率が wise の上昇につれ若干大きくなり，SMALL 属性から SMALL 属性への遷移率が wise の上昇につれて若干小さくなっている。
- SMALL 属性から BIG 属性への遷移率はそれほど小さくはない。

表 5.1 1-wise から 2-wise への属性遷移

		2-wise		
		BIG	SMALL	NA
1-wise	BIG	1973	1485	1222
	SMALL	1469	4271	2245
	NA	0	0	0

表 5.2 2-wise から 3-wise への属性遷移

		3-wise		
		BIG	SMALL	NA
2-wise	BIG	8887	4416	7143
	SMALL	6321	16420	13024
	NA	528	750	19609

表 5.3 3-wise から 4-wise への属性遷移

		4-wise		
		BIG	SMALL	NA
3-wise	BIG	25940	6216	15085
	SMALL	13000	25988	27619
	NA	3003	4477	119596

表 5.4 1-wise から 2-wise への属性遷移率

		2-wise		
		BIG	SMALL	NA
1-wise	BIG	42.16%	31.74%	26.11%
	SMALL	18.40%	53.49%	28.12%
	NA	0.00%	0.00%	0.00%

表 5.5 2-wise から 3-wise への属性遷移率

		3-wise		
		BIG	SMALL	NA
2-wise	BIG	43.47%	21.60%	34.94%
	SMALL	17.67%	45.91%	36.42%
	NA	2.53%	3.59%	93.88%

表 5.6 3-wise から 4-wise への属性遷移率

		4-wise		
		BIG	SMALL	NA
3-wise	BIG	54.91%	13.16%	31.93%
	SMALL	19.52%	39.02%	41.47%
	NA	2.36%	3.52%	94.11%

表 5.7 平均遷移率

		(n+1)wise		
		BIG	SMALL	NA
n-wise	BIG	46.84%	22.16%	30.99%
	SMALL	18.53%	46.14%	35.33%
	NA	2.45%	3.56%	94.00%

5.2 実験結果 2

[4.6 分析 2] によって得られた集計結果を表 5.8～表 5.10 に示す。

表 5.8 について説明する。Base は遷移元である 1-wise パラメータの属性，add は付加される 1-wise パラメータの属性を表し，交絡の数字は左に示す Base と add の組み合わせである遷移先の 2-wise ペアの，上に示す属性の内訳を表している。

例えば，遷移元の 1-wise パラメータの属性が BIG，付加される 1-wise パラメータの属性も BIG である 2-wise ペアの属性の内訳は BIG が 60.83%，SMALL が 14.83%，NA が 24.34%である。

同様に，表 5.9 は 2-wise を Base に，表 5.10 は 3-wise を Base にした属性遷移の内訳を示している。

この結果から読み取れる事柄を以下に示す。

- BIG 属性の Base に BIG 属性の add を付加した場合の BIG 属性への遷移率がかなり高い。
- SMALL 属性の Base にどちらの属性の add を付加した場合でも，SMALL 属性への遷移率は BIG 属性への遷移率より高い。
- 1-wise の BIG 属性を持つ Base に SMALL 属性を持つ add を付加した場合の 2-wise は SMALL がかなり多いが，2-wise 以上の BIG 属性を持つ Base に SMALL の add を付加した場合は BIG 属性への遷移率のほうが高くなる。
- SMALL 属性の Base に SMALL 属性の add を付加した場合でも，BIG 属性への遷移がそれほど低くない割合が存在する。
- BIG 属性と SMALL 属性の Base にどちらの属性の add が付加した場合でも NA 属性への遷移率はあまり変わらない。
- NA 属性の Base にどちらの属性の add を付加した場合でも，NA 属性への遷移率がかなり高い。
- NA 属性の Base が BIG または SMALL に遷移するとき，add の属性のほうに遷移することが比較的多い。

表 5.8 1-wise から 2-wise への付加・被付加パターンと属性遷移

		2-wise				
		Base	add	BIG	SMALL	NA
1-wise	BIG	BIG		60.83%	14.83%	24.34%
		SMALL		15.81%	60.75%	23.44%
		NA		0.00%	0.00%	0.00%
	SMALL	BIG		15.81%	60.75%	23.44%
		SMALL		20.24%	51.98%	27.79%
		NA		0.00%	0.00%	0.00%
	NA	BIG		0.00%	0.00%	0.00%
		SMALL		0.00%	0.00%	0.00%
		NA		0.00%	0.00%	0.00%

表 5.9 2-wise から 3-wise への付加・被付加パターンと属性遷移

		3-wise				
		Base	add	BIG	SMALL	NA
2-wise	BIG	BIG		51.40%	14.70%	33.91%
		SMALL		35.12%	25.35%	39.53%
		NA		0.00%	0.00%	0.00%
	SMALL	BIG		24.80%	45.62%	29.58%
		SMALL		14.25%	45.86%	39.90%
		NA		0.00%	0.00%	0.00%
	NA	BIG		4.23%	2.98%	92.79%
		SMALL		2.31%	3.65%	94.04%
		NA		0.00%	0.00%	0.00%

表 5.10 3-wise から 4-wise への付加・被付加パターン毎の属性遷移率

		4-wise			
	Base	add	BIG	SMALL	NA
3-wise	BIG	BIG	60.29%	9.23%	30.47%
		SMALL	47.17%	16.83%	36.00%
		NA	0.00%	0.00%	0.00%
	SMALL	BIG	22.98%	43.36%	33.66%
		SMALL	17.40%	35.87%	46.73%
		NA	0.00%	0.00%	0.00%
	NA	BIG	5.10%	4.35%	90.55%
		SMALL	1.54%	4.09%	94.37%
		NA	0.00%	0.00%	0.00%

5.3 実験結果 3

[4.6 分析 3] によって得られた集計結果を表 5.11～表 5.13 に示す。

これらの表について説明する。ある n -wise ペアを構成する 1-wise パラメータの属性が BIG か SMALL であるとき、そこに含まれる BIG の属性を持つ 1-wise パラメータの数によって n -wise の属性はどう変化するかを集計したものである。

表 5.11 では、交絡の数字は左に示す 1-wise の持つ属性の組み合わせのうち、上に示す 2-wise ペアの属性の内訳を表している。例えば、BIG の属性を持つ 1-wise パラメータが 2 つで構成される 2-wise ペアの属性の内訳は BIG が 60.83%、SMALL が 14.83%、NA が 24.34%である。また、BIG の属性を持つ 1-wise パラメータ 1 つと SMALL の属性を持つ 1-wise パラメータ 1 つで構成される 2-wise ペアの属性の内訳は BIG が 15.81%、SMALL が 60.75%、NA が 23.44%である。

同様に、表 5.12 は 3-wise の構成要素による属性の内訳、表 5.13 は 4-wise の構成要素による属性の内訳を示している。

この結果から読み取れる事柄を以下に示す。

- 一般的に、 n -wise ペアが全て BIG 属性を持つ 1-wise パラメータで構成される場合のみ、 n -wise ペアの属性が BIG である割合は SMALL である割合より大きくなる。また 1 つでも SMALL 属性の 1-wise が含まれていれば、 n -wise ペアが SMALL 属性である割合が BIG である属性より大きくなる。
- SMALL 属性の 1-wise が 2 つ以上増えていっても、 n -wise が SMALL 属性を持つ割合はあまり変化がなく、代わりに NA 属性を持つ割合が高くなっていく傾向にある。
- SMALL の属性を持つ 1-wise が多くなると、NA である割合は大きくなる。
- 構成する 1-wise の属性に関係なく、wise が上がると NA である割合は大きくなる。

表 5.11 2-wise の要素分解

1-wise		2-wise		
BIG	SMALL	BIG	SMALL	NA
2	0	60.83%	14.83%	24.34%
1	1	15.81%	60.75%	23.44%
0	2	20.24%	51.98%	27.79%

表 5.12 3-wise の要素分解

1-wise		3-wise		
BIG	SMALL	BIG	SMALL	NA
3	0	44.06%	13.74%	42.20%
2	1	20.68%	38.96%	40.36%
1	2	18.95%	29.18%	51.87%
0	3	13.13%	27.14%	59.73%

表 5.13 4-wise の要素分解

1-wise		4-wise		
BIG	SMALL	BIG	SMALL	NA
4	0	33.47%	11.73%	54.80%
3	1	23.28%	27.10%	49.62%
2	2	25.63%	12.28%	62.09%
1	3	16.38%	15.56%	68.06%
0	4	8.24%	11.96%	79.80%

6. 考察

6.1 分析の有効性

第5章で示した分析結果を踏まえて、第2章で定めた研究設問に答えることで分析手法と分析結果の有効性について考察する。

RQ1: パラメータ組み合わせのテスト結果への影響度を表すロジスティック回帰モデルは構築できるか。

本研究では、第4章に述べた分析準備の手法でテストスイートとテスト結果から 4-wise までの組み合わせを抽出し、t-wise 組み合わせ存在表を作成してロジスティック回帰の独立変数とし、またテスト結果を存在表に対応する result 変数としてロジスティック回帰の従属変数とすることで、各パラメータ組み合わせの持つテスト結果への影響度をロジスティック回帰係数として表すことに成功し、ロジスティック回帰モデルを構築できた。

RQ2: 明確な傾向として分析結果は得られるか。

本研究で分析を行った結果、第5章に示したように大小様々な傾向が得られた。特に「NA の属性を持つ t-wise ペアをサブセットに含む組み合わせの9割が NA の属性を持つ」という結果は複数の分析から読み取れ、顕著なものであると言えるだろう。

RQ3: 得られた分析結果は不具合を含む上位組み合わせの予測に有用か。

不具合を含む上位組み合わせを予測して絞り込むには、不具合を含む組み合わせに共通する傾向を利用して抽出する方法か、不具合を含まない組み合わせに共通する傾向を利用してそれらを排除する方法のどちらかが取られると考えられる。

例えば上で述べた「NA の属性を持つ t-wise ペアをサブセットに含む組み合わせの9割が NA の属性を持つ」という傾向を予測絞り込みに用いることで、不具合を含まない組み合わせの9割を実際のテスト実行から省くことができる。これは大いに有用である。また得られた他の傾向に関しても精度は劣るものの、不具合を含む組み合わせの予測に利用することでテストの効率化は見込

まれる。これらの傾向を組み合わせると予測に利用することができれば精度はより高まるだろう。

6.2 分析の妥当性

6.2.1 外的妥当性

今回、分析に使用したテストデータはバージョン 64 個分であり、一般的な性質を導くのに十分な数であると断言はできない。また [4.2.4 組み合わせの削減] で述べたように、この分析には NA の属性を持つ 1-wise パラメータを含めていない。結果的に NA 属性はほとんどの割合で NA 属性にしか遷移しないことが示されたとはいえ、NA の属性を持つ 1-wise パラメータを除外せずに使用することで分析結果に影響を及ぼす可能性は否定できない。

6.2.2 内的妥当性

回帰係数による属性分けの分類基準は厳密さとはほど遠い。この分類方法により BIG と分類されるべき組み合わせが SMALL に分類されていたり、その逆が存在しないという可能性は 0 に近いだろう。分類が厳密でなければ分析結果に影響が出るのは自明である。その影響次第では本研究の成果は全て覆される可能性もある。

7. 結言

本研究では、不審な組み合わせの絞り込みを用いた組み合わせテストの新手法開発に有益な情報を得るため、ロジスティック回帰モデルを用いた t-wise 組み合わせの不具合影響度の分析を行った。またロジスティック回帰によって得られた回帰係数を通して、ある組み合わせの影響度がより大きな組み合わせの影響度に作用するかといった分析を行った。分析の結果、特定の組み合わせとそれをサブセットに含むより大きな組み合わせの影響度に関する様々な傾向が確認できた。またこれらの傾向が前述の手法に有効である可能性が示された。

今後の課題としては、このモデルからさらに有用な傾向発見が得られないか模索すること、今回使用した分析対象以外にも他のデータを対象として再実験を行い結果を比較検証すること、そして今回の研究で得られた結果を利用して高次の不審な組み合わせを高い精度で予測する手法を確立し、実用的なテスト手法を提案することなどが挙げられる。

謝辞

本研究を行うにあたり、研究課題の設定や研究に対する姿勢、本報告書の作成に至るまで、全ての面で丁寧なご指導を頂きました、本学情報工学・人間科学系 水野修 准教授に厚く御礼申し上げます。貴重なデータをご提供頂きました、産業技術総合研究所情報技術研究部門ソフトウェアアナリティクス研究グループの崔 銀恵氏に深く感謝致します。本報告書執筆にあたり貴重な助言を多数頂きました、本学情報工学専攻河端 駿也先輩、山田 晃久先輩、采野 友紀先輩、藤原 剛史先輩、森 啓太先輩、情報工学課程 黒田 翔太君、田中 健太郎君、原田 禎之君をはじめとする、ソフトウェア工学研究室の皆さん、学生生活を通じて著者の支えとなった家族や友人に深く感謝致します。