日本オペレーションズ・リサーチ学会
2021年 秋季研究発表会

大量のフォールトデータに対する判別分析の適用可能性に関する一考察

05001452　東京都市大学大学院　*渡邉勇人 †　WATANABE Hayato
01308725　山口大学大学院　田村慶信 ††　TAMURA Yoshinobu
01702425　鳥取大学大学院　山田茂 †††　YAMADA Shigeru

E-mail: †g2181496@tcu.ac.jp,††tamuray@yamaguchi-u.ac.jp,and†††yamada@see.tottori-u.ac.jp

1　研究の背景と目的

近年，オープンソフトウェア (Open Source Software, 以下 OSS と略す) が急激に普及している．大きな理由としては，OSS は無償で公開されているソフトウェアのため，誰でも自由に改良や再配布することができることや，低コストで高品質な開発を行える点があげられる．しかし，急激に普及されている一方で運用中に大量のフォールトが発見される．こうした大量のフォールトを処理するのはほとんどの場合，修正担当者が行う．修正担当者はバグの原因調査から修正方法の決定を行っているため，原因調査にかかる時間が増加するとコード修正の長期化へとつながり問題になる．こうした修正時間を減らすためにはバグの原因調査を早期に行う必要がある．

本研究では，こうした大規模バグトラッキングデータに対して，判別分析の中の線形判別分析を適用することで，フォールト出現の原因となった様々な要因に関する情報が登録されているデータから発見されたバグの要素をいくつかのグループに分類することを目的としている．また，信頼性の観点から有用となる情報を判別分析により明らかにする．これにより，フォールトを発見する際にどの項目に注目したら良いのか明らかにできることで原因調査の時間削減が可能になり，修正時間の削減やフォールト数の減少によって OSS の信頼性向上にもつながると考える．

2　先行研究と本研究の位置付け

OSS に関する研究は，これまでソフトウェアの信頼性評価など多数行われている [1]．しかし，統計的に分析した研究は数少ない．また，バグトラッキングデータは膨大な量があり，要素間の相関関係を探すなどの統計的な傾向を把握するためには，何らかの統計学的な手法をとらなくてはいけない．これまでの研究では，脆弱性や Critical などの危機的なフォールトの出現率，コメントアウトとフォールト潜在との関係を明らかにする研究が行われてきた [2,3]．これらの研究では，ある特定の項目に着眼した推定手法が提案されており，全体的な傾向を把握するという俯瞰的な観点からの研究は行われていない．本研究では，全体的な傾向を見てバグの種類を判別することで，今までの研究とは異なるアプローチにより統計的な分析を行う手法を提案する．

3　研究方法

3.1　判別分析

本研究では判別分析を用いる．判別分析とはいくつかのグループに分かれているデータをもとに，それらのデータがどのグループに属するのかを調べる方法とされている．次に，判別分析の方法として 2 種類ある．1 つ目は線形判別関数で仕切る方法である．2 つ目はマハラノビス距離で仕切る方法である．本研究では，線形判別関数を用いて分析を行う．特に，フィッシャーの線形判別分析によりフォールトデータの解析を行う．線形判別分析とは，データを 2 種類の群に分けるための境界線を直線で引く方法である．この方法は，その直線を線形判別関数として数式化することで，未分類であったデータが数式によって分類できる手法である．線形判別分析を行うために必要な以下の条件は満たしていると想定される．

1. データは多変量正規分布に基づいている
2. 各クラスは同じ共分散行列を持っている

しかし，1 つ目の条件であるデータは多変量正規分布に基づいていることは現実問題として多くのデータがきれいに正規分布しているものは存在しない．そこで，本研究では図 1 に，実際にフォールトビッグデータの全カテゴリーの正規分布に基づく近似曲線の一例を示す．上記の図 1 における x 軸は，変数の出現回数を示している．また，y 軸は各要因の発生率を表している．これらの図は全体として，ほとんどのデータ要因は，図 1 の正規分布にほぼ基づいていることが分かる．また，判別問題を扱う際の変数の値は，故障解析データの特性を考慮して，各要素の出現回数とする．これは，信頼性の観点から扱うデータとして，従来から故障回数データがソフトウェア信頼性評価モデルを利用した信頼性評価法として利用されているためである．特に，非同次ポアソン過程（NHPP：Non-homogeneous Poisson process）に基づくソフトウェア信頼度成長モデル，ハザードレートモデル，確率微分方程式モデルのような確率モデルに基づく多くのソフトウェア信頼性モデルにおいて，フォールト発見数データもしくは故障発生時間間隔データが用いられてきた [5]．よって，本研究の判別分析における変数に対する数量データとしては，上述した既存のソフトウェア信頼性モデルにおいて歴史的に多用されている故障回数データを各要因に対して適用する．

3.2　分析の手順

上記の条件を満たしたものの分析の流れとしては，はじめに変数の 1 つを目的変数とし，それ以外の変数を説明変数にする．それらの関係を調べ説明変数を x，係数を a，判別得点を Z とした以下のような関係式を作成する．

$$Z = a_1 \times x_1 + a_2 \times x_2 + \cdots + a_n \times x_n + a_0. \qquad (1)$$

この判別得点 Z を求めた上で散布図に表し分類を行う．

本研究では，目的変数の中にある種類が多様なため，正準判別分析を行う．このとき，判別得点は正準スコアとなる．さらに，上のような式を正準判別関数と呼び，群の個数より 1 つ少ない数か説明変数の数を比較して小さい方の数の式が出力される．出力された式に分類したいデータを代入し，正準スコアが最大となる群に属すると判定する．

4　数値例

目的変数を Severity および Version とした線形判別分析に基づく数値例の一部を図 2 に散布図として示す．ここでの横軸は第 1 正準スコアである．また，縦軸は第 2 正準スコアである．さらに，データ全体の情報の中で各要素の持つ情報が占める割合のことである寄与率を算出してい

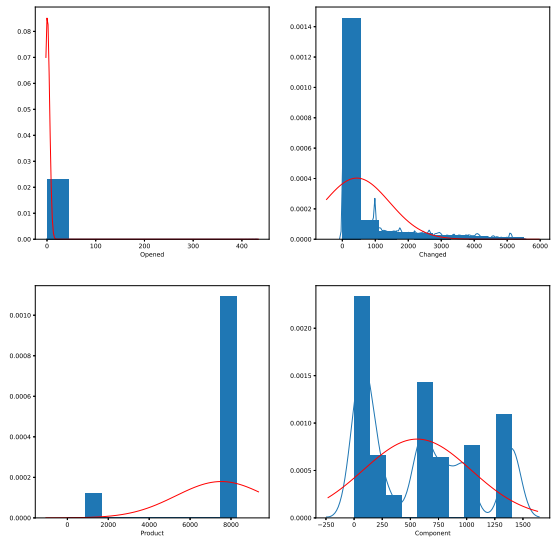

図 1　正規分布の近似曲線.

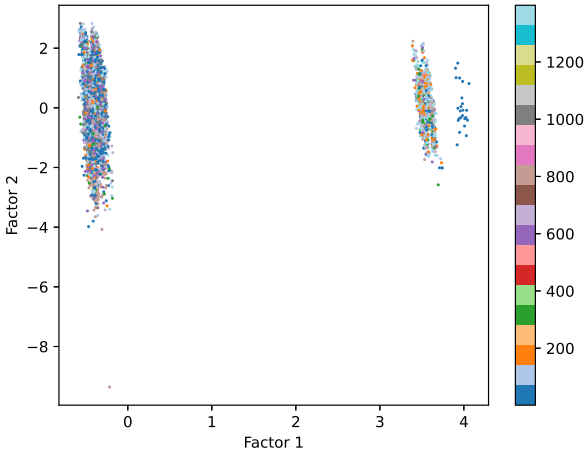

図 2　Component の正準スコアの散布図.

る．多くの変数で，第 2 正準関数まで約 80〜90 %の情報が把握できるため，第 2 正準関数までを用いる．

5　要約

Component:　第 1 正準関数の正負という 2 つのグループに分類することができる．負のグループでは，黄緑の点である Documentation，黒の点である Core などのコアコンポーネントやメインとなるコンポーネントが分類されている．一方，正のグループでは緑の点である Other やサブコンポーネントなどといった小さいサイズのコンポーネントが分類されている．

6　考察

各変数に対して要約を行った結果から，いくつかの結果が同じような分布を表していることが分かった．例えば Hardware，Serverty および Reporter はグループの形成の仕方が酷似している．これは 3 つの変数が同じような傾向にあるとことを意味している．これらは同じような傾向であるため，関係性を散布図と正準関数を用いて考察を行う．Severity の正準係数は Reporter と Version，Summary が高く，Hardware が低くなっていた．このことから，重要度に同様な傾向で影響度が高いのは報告者の報告数であることが分かった．また，Version，Summary も影響度が高い．よって，バージョンが新しく，多数の報告者から報告されており，コメントが多いバグは重要度が高くなる傾向にあることが推察できる．これは複雑なバグであることや多数の報告を受けていることから早急に対処する必要があるものと考える．また，Component や Version も似たようなグループを形成していることから，同じような傾向を持っていることが分かる．Component と Version の関係性を先ほどと同様，散布図と正準得点を用いて考察を行う．すると，片方の変数で正準得点が大きい値ほど，もう一方の変数では低い点数になる．このため，Version が古いものであるとサブコンポーネントなどの小さいコンポーネントでバグが発生しており，新しい Version であるとコアコンポーネントなどの大きいコンポーネントでバグが発生していることが推察できるため，バグの状態を把握する際には Version の新旧を注視していく必要がある．

7　おわりに

本研究では，フォールトビッグデータの全体的な傾向を把握する方法として，判別分析法を適用した．また，実際に Bugzila というバグトラッキングシステムに登録されている Apache HTTP Sever のフォールトデータを用いて数値例を示した．この，分析方法を適用することによって，過去に提案された何かしらに特化し分析を行った統計手法とは異なり，バグの全体的な傾向を把握することが可能になる．このことから，修正者はフォールトを発見する際には製品のバージョン，報告者，バグ内容が記載されている要約の文字数を注視することで原因調査の時間短縮につながると考える．

しかし，さらなる信頼性向上を図るためには，修正時間の短縮のためにバグの特定を予測する必要がある．今度の課題として，データの傾向を把握するために行った統計的手法は判別分析の他にも多数あることから，他の分析方法と比較し検証することで，より精度の高い分析手法を確立する必要がある．

8　謝辞

本研究の一部は，JSPS 科研費基礎研究 (C) (課題番号 20K11799) の援助を受けたことを付記する．

参考文献

[1] 山田茂，井上真二，田村慶信「ソフトウェア信頼性研究　モデリングアプローチ」　電子情報通信学会，vol. 1，pp. 38-50, 2018.

[2] 阿萬裕久「オープンソースソフトウェアにおけるコメント文記述とフォールト潜在率との関係に関する実証的考察」　情報処理学会論文誌，vol. 53，pp. 97-100, 2010.

[3] 田村慶信「組込み OSS 移植工程に対する MCMC およびハザードレートモデルに基づく信頼性評価法」　数理解析研究所講究録，vol. 1802，pp. 120-126, 2012.

[4] 田村慶信，“組込み OSS 移植工程に対する MCMC およびハザードレートモデルに基づく信頼性評価法”，　数理解析研究所講究録，vol. 1802，pp. 120-126, 2012.