

2017年2月24日

数理科学アプローチを用いた 客観的欠陥弁別法

— 外因欠陥の弁別方法とその効果・意義 —

第7分科会 欠陥エンジニアリング Team TuKuLu

研究員：伊藤 弘毅（三菱電機株式会社）

大島 修（エプソンアヴァシス株式会社）

角 修二（株式会社インテリジェンスビジネスソリューションズ）

牟田 香奈（日本ATM株式会社）

主査：細川 宣啓（日本アイ・ビー・エム株式会社）

副主査：永田 敦（ソニー株式会社）

2021年2月24日 SQiP研究会 特別講演

人工知能による欠陥分類の 次の挑戦

— バグの自動修復技術の実用化に向け —

(2016年度 SQiP研究会発表論文)

数理科学アプローチを用いた客観的欠陥弁別法

～ 外因欠陥の弁別方法とその効果・意義 ～

■本日の講演プログラム

はじめに（2017年当時の振り返り）（15分）

研究の背景（効かない再発防止策）

ソフトウェア欠陥を…

“過ち以外”の軸で分類できるか

科学的な方法で分類できるか

実験と結果

本研究の意義

バグの自動修復技術（1時間30分）

:

■ 研究の背景（効かない再発防止策）

「土下座してみんなに謝れ」

私がバカで不注意だから？土下座して反省すればもう起きない？

様々な原因分析手法や再発防止策が提案され改善を繰り返しているにも関わらず
欠陥は発生し続けている。

それって、エンジニアや企業の反省や努力が足りないから？

欠陥の原因を分析（コーディングミスをした、I/Fが変更されたことに気付かなかった…等）



過ちを繰り返さない対策



対策が効かない



分析と対策が足りない

定性的で主観的な原因分析や

そんな分析結果から再発防止策を捻り出す慣習

もう“やめたい”と思いませんか？

■ ソフトウェア欠陥を“過ち以外”の軸で分類できるか

過ちだけとは言い難い欠陥 = 外因欠陥

例) 環境変化やマネジメント不良による開発ルール変更など

過ちによる一般的な欠陥 = 内因欠陥

分類は科学のはじまり

■ ソフトウェア欠陥を科学的な方法で分類できるか

問題管理票（バグ票）などから欠陥の内容を瞬時に読取る「直感」の正体とは？

斜め読みの技術 + 鳥瞰的視点



テキストマイニング
(自然言語処理)



クラスタリング
(類似度測定)

人間は過去の経験や知識に照らして判断している。

数理科学を用いた**弁別機**を作り
外因欠陥と**内因欠陥**を
自動で**分ける**ことはできるか？

実験してみよう！

■ 外因欠陥の弁別実験 《実施条件》

問題管理票：

第7分科会研究員が作成した欠陥研究用の個票を使用

- ・ 固有名詞や案件固有の制約事項等を削除し、抽象表現に置き換えて汎化したもの

同一起票者によるものを選択

- ・ 書き癖の影響を極力排除
- ・ 実験結果が恣意的に導出されるリスクを低減

テキストマイニングツール：

樋口耕一氏の開発したKH-Coderを利用

- ・ 統計解析機能に「R」
- ・ 形態素解析ツールとしてChasenを利用
(奈良先端科学技術大学院松本研究室開発)
- ・ フリーソフトウェアとして誰でも入手可能
- ・ 本ソフトウェアを利用した研究事例が豊富

■ 外因欠陥の弁別実験 《概要・目的》

入力 : 外因欠陥 1 編 (# 6) を含む欠陥情報 8 編

6 組織が縦割りであるが故に曖昧なルールが正しいものとして信じられていたケース

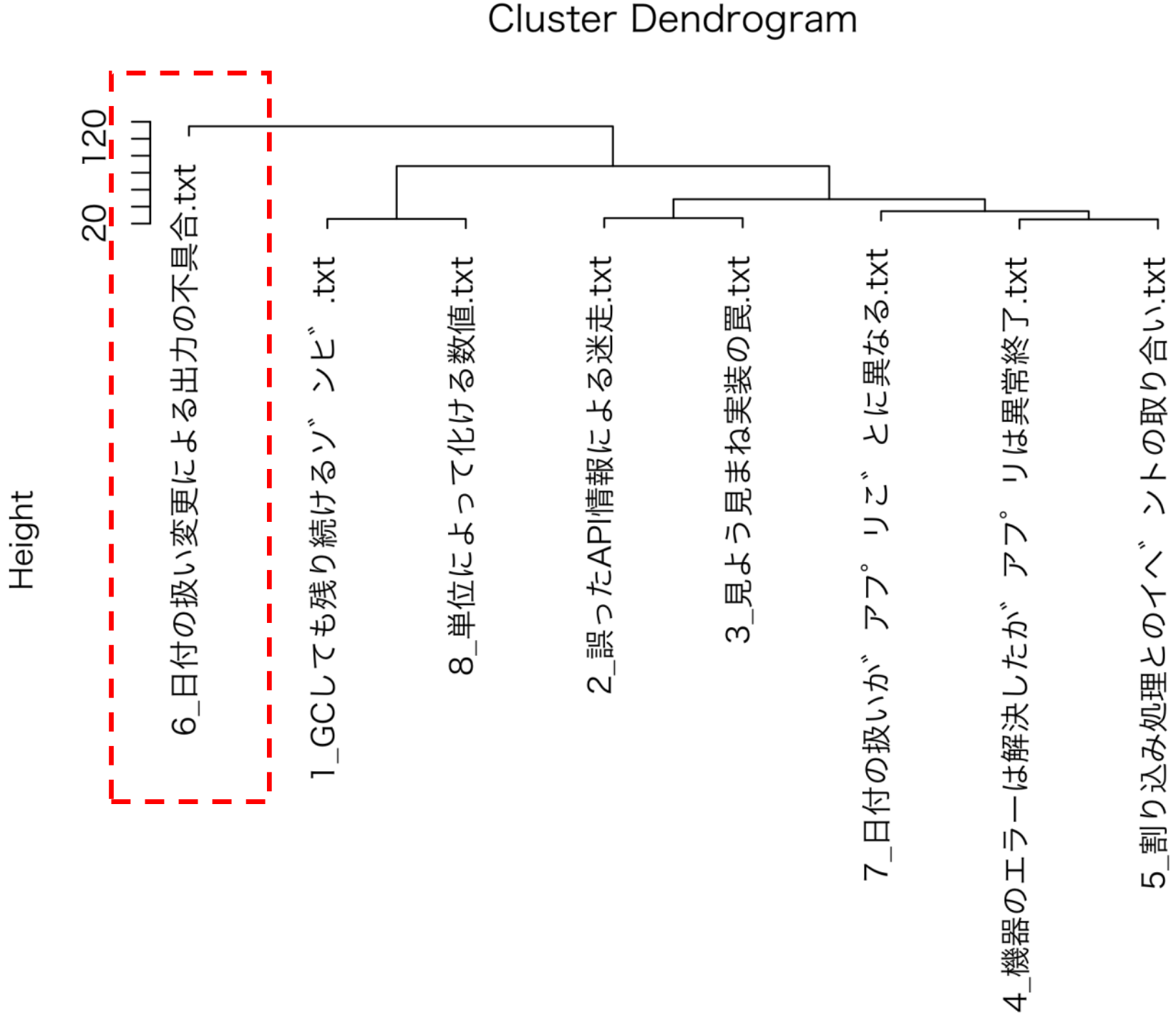
弁別機 : 形態素解析とターム行列生成により特徴量を抽出。
(タームの出現頻度の調整はTF-IDF法を利用)
K-meansクラスタリング※1を行う。

出力 : 生成されたデンドログラム (ツリー構造) にて
外因欠陥情報 # 6 が弁別されていれば※2
弁別方法として有意であることが証明される

※1 ユークリッド距離にて類似度を測る一般的な手法を使用

※2 外因欠陥は内因欠陥群との類似度が低い (距離が遠い)
ことが予測され、この性質を利用する

■ 結果



■ 検証

#1

Word cloud for #1: The largest word is 'する' (do). Other prominent words include 'Thread', 'プラットフォーム', '画面', '異常', 'ない', 'GC', and '実行'.

#2

Word cloud for #2: The largest word is 'する' (do). Other prominent words include 'API', '提供', '改行', 'インデント', '文字', 'データ', 'パーサー', '行', '渡す', '含む', 'API', and 'ない'.

#3

Word cloud for #3: The largest word is 'する' (do). Other prominent words include 'スタック', '修正', '行', 'データ', 'ない', '処理', '担当', '積む', '誤る', '決済', and '実行'.

#4

Word cloud for #4: The largest word is 'する' (do). Other prominent words include 'API', '異常', 'プログラム', '提供', 'アプリケーション', '充電', 'サンプル', '印刷', 'API', '遠移', '実行', 'インストール', '実行', '実行', '実行', '実行'.

#5

Word cloud for #5: The largest word is 'する' (do). Other prominent words include '発生', '通知', '異常', '割り込み', 'センター', 'メイン', '情報', '処理', 'データ', '実行', '実行', '実行', '実行'.

#6

Word cloud for #6: The largest word is '出力' (output). Other prominent words include '未定義', '返す', '編集', 'ロジック', '日付', '利用', 'データ', '機能', '処理', 'ライブラリ', 'なる', '入力', '変換', '考える', '場合', '和'.

#7

Word cloud for #7: The largest word is 'アプリケーション' (application). Other prominent words include '表示', 'データ', '時刻', '実行', 'データ', '営業', 'なる', 'ない', '時刻', '実行', 'データ'.

#8

Word cloud for #8: The largest word is '単位表示' (unit display). Other prominent words include '顧客', '変換', 'データ', '表示', '単位', '表示', '顧客', '変換', 'データ', '表示', '単位', '表示'.

#6の特徴：

- ・他動詞『する』が無い
- ・自動詞『なる』の出現頻度が高い

■ 本研究の意義

弁別機を使うことで“いつでも・簡単に・大量に”欠陥を弁別できる。

科学的アプローチから欠陥情報から対策の打ち出しに大別できることを証明した。

二群分類とはいえ、特徴量による欠陥弁別は前例がなく、その意味では欠陥分類の始まりと言える。

その後の欠陥研究そのものを大きく進歩させる可能性のある発明。

2021年2月24日 SQiP研究会 特別講演

人工知能による欠陥分類の
次の挑戦

— バグの自動修復技術の実用化に向け —

次は、いよいよ本題です！
バグの自動修復技術（1時間30分）

2017年2月24日 SQiP研究会 成果発表会

数理科学アプローチを用いた 客観的欠陥弁別法

— 外因欠陥の弁別方法とその効果・意義 —

本研究の2017年現在の課題：

ソフトウェア欠陥情報を弁別対象とするなら
欠陥情報の品質に弁別結果が左右されないか？

■ 解決案：人工知能を使った文意読み取り

例) 機械が言葉の意味を覚える ⇔ コンピュータも日本語を学習する

- 文節／単語に区切る (形態素解析)

例) 機械が／言葉の／意味を／覚える

- ・ ルールベースで発展、大量文例による機械学習

[関連技術] 係り受け解析、品詞判定 (固有名詞も抽出)

<辞書を作る技術>

- 同義語・語句の類似度を解析

- ・ 語句登録型 (ツールや人手で作成、機械学習による自動生成など)
- ・ 数値判定型 (学習結果などから数値で類似度を表現する)

例) 同義・類似語：学ぶ、学習する、教える、理解する、分かる

- より柔軟に語句を解析する

- ・ 単語の出現頻度などを行列やベクトルで表現
- ・ 単語共起行列：出現回数で解析

- 文脈を反映して語句・文章解析

- ・ 再帰型ニューラルネットワーク (RNN)

- 語句一致度による文意の近さ (距離) 判定

- ・ 一般に数値で優先度を付ける

■最後に

理不尽な土下座は科学が解決します。

今日の発表はゴールではありません。

ご清聴、ありがとうございました。

■ Appendix



■ 実験 《手順》

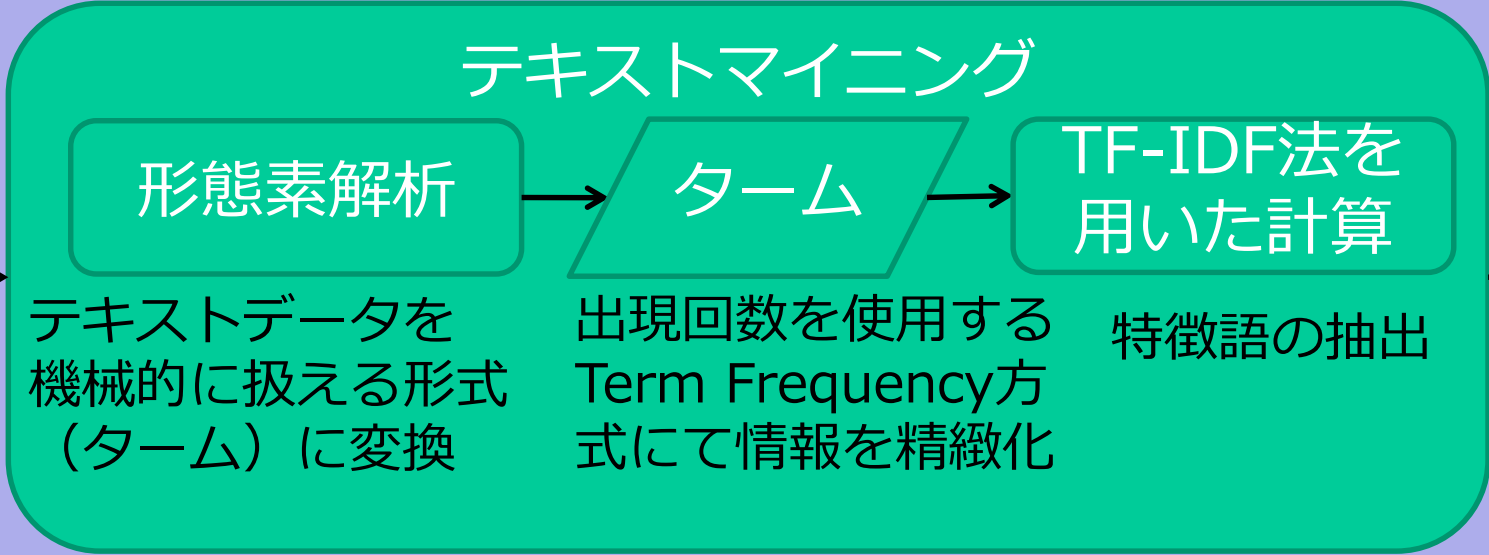
- 1) 欠陥情報8編の形態素解析を行う。解析ルールは以下の通り。
 - ルール① 対象を「名詞」「動詞」「形容詞」「助詞」の4品詞とする。
 - ルール② 出現頻度が8編合計で2以上になるタームのみを対象とする。
 - ルール③ 記号（句読点・句点・括弧他）はタームとして除外する。
- 2) 1)の結果からターム行列（縦軸をターム、横軸を欠陥情報とし、どのタームがどの文書に出現するかをカウントするマトリクス）を生成
- 3) 2)の出力結果から、各出現頻度を局所的・大域的重みにより影響を調整する。本実験では TF-IDF法を利用。
- 4) 出力結果を文章量によって正規化（一般的なコサイン正規化）し、最終的に重みを調整（ノーマライズ）。
- 5) 上記ターム行列の各列間のユークリッド距離を計算する。
- 6) K-Meansクラスタリングにてユークリッド距離の近い（似ている）ものからクラスターが形成され、**デンドログラムが生成**される。
※クラスタリングツリーとも呼ばれる

■ ターム行列 (実験手順 2)

	A	B	C	D	E	F	G	H	I	J
1		1_GCしても残り	2_誤ったAPI情報	3_見よう見まね	4_機器のエラー	5_割り込み処理	6_日付の扱い変	7_日付の扱いか	8_単位によって化ける	数値.txt
2	GC	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	MVVM	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	No	0.08	0.00	0.00	0.11	0.10	0.00	0.00	0.00	0.00
5	Thread	0.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	いる	0.14	0.13	0.12	0.09	0.05	0.10	0.08	0.05	0.05
7	が	0.09	0.11	0.22	0.08	0.15	0.16	0.12	0.16	0.16
8	から	0.08	0.10	0.00	0.15	0.00	0.00	0.00	0.00	0.00
9	こと	0.06	0.09	0.05	0.00	0.00	0.03	0.00	0.06	0.06
10	する	0.28	0.17	0.16	0.25	0.20	0.18	0.19	0.16	0.16
11	て	0.19	0.14	0.16	0.14	0.05	0.15	0.11	0.11	0.11
12	で	0.07	0.04	0.00	0.00	0.09	0.17	0.14	0.18	0.18
13	と	0.09	0.04	0.06	0.05	0.11	0.07	0.12	0.04	0.04
14	ない	0.09	0.07	0.00	0.00	0.00	0.05	0.05	0.09	0.09
15	に	0.07	0.13	0.15	0.08	0.13	0.09	0.12	0.09	0.09
16	の	0.16	0.15	0.28	0.20	0.21	0.20	0.25	0.20	0.20
17	は	0.12	0.15	0.09	0.08	0.04	0.11	0.10	0.11	0.11
18	プラットフォーム	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	まで	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	メモリ	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	も	0.16	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.00
22	モデル	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	リーク	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	れる	0.09	0.10	0.09	0.05	0.05	0.03	0.05	0.05	0.05
25	安定	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	異常	0.13	0.00	0.00	0.22	0.22	0.00	0.00	0.00	0.00
27	印字	0.10	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00
28	画面	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	解放	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

テキストマイニングとは

自由書
式文書



結果の
取得

テキストマイニングとは

形態素解析

テキストデータを機械的に扱える形式に精緻化する

1. 自然言語で意味を持つ最小単位である形態素に分類し、その品詞を特定する。
2. 文書に含まれる単語を列に、文書を行とすると単語の出現回数を要素とした行列形式に変換する。

TF-IDF

以下の考え方にてその文書の特徴を表す単語を抽出する

1. 同じ意味合いの語の出現回数が多い。
2. 他の文書では出現回数が低いが、当該文書では出現回数が多い

■ TF-IDF法

各文書の長さ（＝文章量・単語数）は互いに異なり，その正規化と重み付けの調整を行う必要がある。

（例：10万語の文章 a にターム A が出現した回数が 3 回． 100 語の文章 b にターム A が 3 回出現した場合，結果は同じターム A の 3 回出現であっても，a, b それぞれに対する重要度は異なる）

それぞれ一般的な正規化 (Normalization)，局所的重みを索引語頻度 (TF: Term Frequency) 大域的重みを文書頻度逆数 (IDF: Inverse Document Frequency) として影響調整を行う。TF (tf) と IDF (idf) 二つの指標に基づいて計算される。

$$\text{tfidf}_{ij} = \text{tf}_{ij} \cdot \text{idf}_i \quad \text{tf}_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \quad \text{idf}_i = \log \frac{|D|}{|\{d: d \ni t_i\}|}$$

$n_{i,j}$ は単語 t_i の文書 d_j における出現回数， $\sum_k n_{k,j}$ は文書 d_j におけるすべての単語の出現回数の和， $|D|$ は総文書数， $|\{d: d \ni t_i\}|$ は単語 t_i を含む文書数である。そのため，idf は一種の一般語フィルタとして働き，多くの文書に出現する語（一般的な語）は重要度が下がり，特定の文書にしか出現しない単語の重要度を上げる役割を果たす。

TF-IDF 法 (TF=Term Frequency=単語の出現頻度) と IDF (Inverse Document Frequency=逆文書頻度) ，，“ウィキペディア日本語版，” 13 9 2016.

<https://ja.wikipedia.org/wiki/Tf-idf.>)