N-001

# IRM と決定木を用いたプログラミング初学者の能力判定のための特徴量の抽出
# Extraction of Feature Value for Estimating Novice Programmer Skill using IRM and Decision Tree

飯棲 俊介† 大枝 真一‡
Shunsuke Iizumi Shinichi Oeda

## 1. はじめに

初学者に対するプログラミング教育では，授業に追従できていない学生の把握が難しく，学生間で能力に差が生じてしまうという課題がある．

千枝ら[1]の先行研究では，授業に追従できていない学生の早期発見のため，学生の書いたソースコードをもとに決定木を作成し，プログラミング能力の判別に重要となる特徴量の抽出が試みられている．

他方で，Križanić et al.[2]による先行研究では，e-Leaning システムの講義のアクセスデータを決定木と k-means 法を同時に用いた手法により，成績の良い学生グループを見つけることで効果的な学習方法の抽出に成功している．

本研究ではこれら先行研究に基づき，プログラミング学習において能力判定に重要となる特徴量をより詳細に求めるべく，k-means 法と決定木を組み合わせた手法を用いた実験・検証を行う．

また，k-means 法よりも良い精度を得るべく，共クラスタリング手法である IRM を用いて学生と特徴量からクラスタ群を生成する手法を提案する．生成されたクラスタ群より類似した特徴量のクラスタ，学生のクラスタを抽出し決定木を構成することで，学生の能力による特徴量の抽出が可能になると考える．

## 2. 手法

### 2.1 決定木

決定木とは，特徴軸に設定した閾値により領域を分割するクラス予測モデルである．決定木は教師あり学習手法により，分割した領域に所属するデータの混じりけの汚さを表す指標である不純度を最も小さくするように閾値が決定される．領域分割の結果は木構造により可視化することができ，結果の解釈に優れた手法である．

### 2.2 ランダムフォレスト

ランダムフォレストとは，決定木に生じる過学習の問題を解消するために開発された手法で，決定木を組み合わせその多数決により出力するクラスを決定するアンサンブル学習手法である．ランダムフォレストは学習データに対する分散を減らすため，各決定木は特徴軸を分割するための候補となる特徴量を限定して学習を行い，各決定木同士の相関を減らしている．ランダムフォレストでは決定木集合全体の結果として，特徴量がどれだけ分割に影響したかを表す特徴重要度が算出される．本実験ではこのランダムフォレストを予測モデルに使用する．

† 木更津工業高等専門学校 制御・情報システム工学専攻
Advanced Course of Control and Information Engineering, National Institute of Technology, Kisarazu College
‡ 木更津工業高等専門学校 情報工学科
Information and Computer Engineering, National Institute of Technology, Kisarazu College

## 3. ソースコードの特徴量抽出

Caliskan-Islam et al.[3]の先行研究では，ソースコードより構文的特徴，文法的特徴，また書き方の癖や嗜好を表す特徴量を抽出し，ソースコードの著者推定を行なっている．本研究ではこの先行研究で利用された特徴量を基に計 20 種類の特徴量の抽出を行った．表 1 に使用した特徴量を示す．

表 1　使用した特徴量

| | |
|---|---|
| 関数の引数の数 | 空行の出現頻度 |
| 関数名の平均長 | 1 行コメントの出現頻度 |
| 関数の平均行数 | 複数行コメントの出現頻度 |
| 関数呼出の頻度 | “if”の出現頻度 |
| 変数の数 | “else if”の出現頻度 |
| 変数名の平均長 | “else”の出現頻度 |
| インデント文字 | “switch”の出現頻度 |
| インデント形式 | “while”の出現頻度 |
| 1 行あたりの平均文字数 | “for”の出現頻度 |
| 1 行あたりの文字数の標準偏差 | “do”の出現頻度 |

## 4. 実験方法

本実験では，2022 年度木更津工業高等専門学校情報工学科 2 年の学生 42 名を対象に行われる「プログラミング基礎 I」の授業にて収集された C 言語のソースコードを利用する．対象とするプログラムは校内の中間試験において試験時間 90 分内に作成された 4 種の問題のプログラムであり，100 点を満点とする教員の採点結果が付加されている．各問題の内容は次のとおりである．

1. 0 から 99 までの数字のうち 3 の倍数の和を求めよ．
2. 2 つの商品の購入個数を引数とし，合計金額を戻り値とする関数を設計せよ．
3. 任意の 1 次元配列から最大値とそのインデックスを求めるプログラムを作成せよ．
4. 入力した整数値を逆順に出力せよ．

本実験では，最終的に提出されたソースコードから表 1 に示した 20 種類の特徴量の抽出を行う．特徴量は設問ごとに異なるものとみなし，計 80 種類の特徴量として扱う．各特徴量は正規化を行う．以下の実験ではこの 80 次元特徴量ベクトルより，教員の採点結果を予測するモデルを作成する．

### 4.1 ランダムフォレスト単体による実験

得られたソースコードを特徴量に変換し，全てのデータを用いて教員の評定を予測するランダムフォレストモデルを作成する．作成されたランダムフォレストより特徴量重要度を算出し，分割に大きく影響した特徴量を算出する．

### 4.2 k-means 法を用いた実験

得られたソースコードを特徴量に変換した後，k-means 法により学生のソースコードをクラスタリングする．そし

て，得られた各クラスタに属するデータ集合から教員の評定を予測するランダムフォレストモデルを作成し，特徴量の解析を行う．Križanić et al. [2]の先行研究では，k-means 法によりあらかじめクラスタリングを行うことで成績の良いグループとそうでないグループに分割することができ，グループに応じた詳細な特徴重要度が得られている．本実験でも各クラスタに応じた詳細な特徴量の重要度を得ることでグループの能力に合わせた特徴量の抽出を図る．

本実験では，k-means 法のハイパーパラメータである分割クラスタ数を 3 として実験を行った．

### 4.3 IRM を用いた実験

得られたソースコードを特徴量に変換した後，IRM によりソースコードの特徴量，学生を同時クラスタリングする．作成された各クラスタに所属する学生ソースコードを，所属する特徴量を用いて教員の評定を予測するランダムフォレストモデルを作成する．

k-means 法ではクラスタ数をあらかじめ設定しクラスタリングを行なうのに対し，IRM はクラスタリングの中でクラスタ数を決定する手法であり適切なクラスタ数を自動的に求めることができる．また，共クラスタリングにより特徴量間でもクラスタを作成し類似した特徴量を使って決定木を作成することで，k-means 法よりも詳細なクラスタリング結果を得ることができると考える．

## 5. 実験結果

データ収集の結果，41 名分のソースコードを収集した．学生によっては，制限時間の関係で問題 4 が未回答の学生も存在した．また，全ての学生で特徴量の値が 0 であったものを除外した結果，64 種類となりこれらの特徴量を使用した．試験の得点を見ると平均点が 88 点，標準偏差が 7.5 という分布であったため，得点が 90 点以上の学生，80~89 点の学生，79 点以下の学生の 3 種類をラベルとしてランダムフォレストによる分類を行った．

### 5.1 ランダムフォレスト単体による実験

実験の結果，重要度の高かった特徴量上位 5 種を表 2 に示す．表 2 を見ると問題 3,4 の文字数や変数に関わる特徴量が上位を占めているという結果になった．これは，問題 3,4 の解法を思い付かずプログラムの行数が少なかった学生を分ける要因となったためだと考えられる．

表 2　特徴重要度の上位 5 種類

| |
|---|
| 問題 3，文字数の標準偏差 |
| 問題 4，文字数の標準偏差 |
| 問題 4，平均文字数 |
| 問題 3，変数名の平均長 |
| 問題 2，変数名の平均長 |

### 5.2 k-means 法を用いた実験

k-means 法によるクラスタリングの結果，比較的点数の高いグループ A，中程度のグループ B，低いグループ C の 3 種類にクラスタ分けをすることができた．そして，各クラスタでランダムフォレストより算出された特徴重要度のうち上位 5 種類を表 3 に示す．各グループで算出された重要度はいずれも大きく異なるという結果となった．グループ B の結果は実験結果 5.1 と似通っているが，グループ A の結果は全く違うものとなった．これは，問題 3,4 を解くためには for 文や while 文を駆使する必要があり，得点の高いグループ内で解けた学生と解けなかった学生を分ける大きな要因となったためだと考えられる．グループ C は問題 1,2 の特徴重要度が高かったことから，問題 3,4 を上手く解けなかったグループであり，問題 1,2 の内容が大きく影響したためと考えられる．

表 3　各クラスタの特徴重要度上位 5 種類

| グループ A | グループ B | グループ C |
|---|---|---|
| 問題 3<br>“for”の出現頻度 | 問題 4<br>文字数の標準偏差 | 問題 1<br>変数名の平均長 |
| 問題 4<br>“while”の出現頻度 | 問題 3<br>文字数の標準偏差 | 問題 2<br>空行の数 |
| 問題 4<br>空行の出現頻度 | 問題 3<br>変数名の平均長 | 問題 3<br>文字数の標準偏差 |
| 問題 4<br>変数の数 | 問題 4<br>変数の数 | 問題 2<br>変数名の平均長 |
| 問題 1<br>平均文字数 | 問題 4<br>変数名の平均長 | 問題 2<br>文字数の標準偏差 |

### 5.3 IRM を用いた実験

IRM によるクラスタリングの結果，特徴量のクラスタ数は 9，学生のクラスタ数は 6 という結果になった．クラスタリングの結果，例えば「関数の平均行数」と「空行の数」，「“for”の出現頻度」と「“if”の出現頻度」は同じクラスタに所属し類似した特徴量であると見なされた．各学生クラスタの教員の評定を見ると，比較的点数の高いグループが 2 種類，中程度のグループが 3 種類，低いグループが 1 種類という結果となった．ランダムフォレストにより各クラスタの特徴重要度を見ると，同じ特徴クラスタを対象としたものでも学生クラスタによって例えば「関数の平均行数」の方が重要度の高いものと「空行の数」の方が要度の高いものに分かれるという結果になった．この結果より，IRM は類似したクラスタを得た上で，詳細な特徴量の重要度を算出し役立てることができると考える．

## 6. まとめ

本研究では，初等プログラミングにおいて能力判定に重要となる特徴量を k-means 法，IRM を用いて効果的に抽出するための手法についての研究を行った．実験の結果，どちらの手法も評定が同程度のクラスタを得た上で詳細な特徴重要度を得ることができ，詳細な解析に役立てられると考えられる．

**謝辞**

本研究は JSPS 科研費 19H01728 の助成を 受けたものです.

**参考文献**

[1] 千枝 睦実 ,大枝 真一, “プログラミング授業での決定木を用いたドロップアウト原因の可視化”, 2019 年情報科学技術フォーラム(2019).

[2] Snježana Križanić, “Educational data mining using cluster analysis and decision tree technique: A case study”, International Journal of Engineering Business Management, Vol.12, No.3(2020).

[3] Aylin Caliskan-Islam, “De-anonymizing Programmers via Code Stylometry”, 24th USENIX Security Symposium(2015).