2ZJ-08

クラスタリング手法とランダムフォレストを用いたプログラミング能力を決する特徴量の抽出

飯棲 俊介 †　大枝 真一 ‡

木更津工業高等専門学校 制御・情報システム工学専攻 †　木更津工業高等専門学校 情報工学科 ‡

1. まえがき

初学者に対するプログラミング教育は，学生によって理解状況に差が生じやすく，授業に追従できていない学生の早期発見が求められる．千枝らの先行研究 [1] は，学生の書いたソースコードをランダムフォレストにより分類し，成績の低い学生の早期発見と分類に重要となった特徴量の抽出が試みられている．他方で，Križanić et al. による先行研究 [2] では，学習ログデータを決定木と k-means 法を用いた手法により，類似した行動をする学生間で成績の良い学生の持つ特徴量を見つけ効果的な学習方法の抽出に成功している．本研究は，プログラミング能力の判定に重要となる特徴量をより詳細に求めるため，k-means 法とランダムフォレストを組み合わせた手法を用いた特徴量抽出の実験を行う．また，k-means 法よりも良い精度を得るため，クラスタリング手法を共クラスタリングである後述の無限関係モデル (IRM:Infinite Relational Model) として同様の実験を行う．IRM はソースコードのクラスタのみではなく特徴量のクラスタも同時に得ることができ，解析に効果的に機能すると考える．

2. 手法

2.1. ランダムフォレスト

ランダムフォレストは，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の決定木を，複数組み合わせた手法である．ランダムフォレストの各決定木は，特徴軸分割の候補となる特徴量を限定して学習を行い，相関が低くなるように作成をする．ランダムフォレストの結果より，クラスの分かれ具合を表す不純度の減少量から特徴量がどれだけ分割に影響したかを表す特徴重要度を算出することで，分割に重要となった特徴量を得ることができる．

2.2.IRM

IRM は，異なるオブジェクト同士の関係データに対する確率モデルであり，関係の類似性から異種オブジェクトを共クラスタリングする手法である [4]．IRM は，クラス数をあらかじめ決定することなく，クラスタ群に仮定した事前確率を最小とするように学習の中で最適なクラスタ数が決定される．

3. 提案手法

本研究は，千枝ら [1] の先行研究と同様にソースコードの特徴量化を行い，ランダムフォレストでプログラミング能力の判定に重要となる特徴量抽出を行うが，先行研究 [2] を参考にしてソースコードに対して k-means 法，IRM によるクラスタリングを事前に行い，得られたクラスタ群に対しランダムフォレストによる特徴量抽出を試みる．特徴量の抽出は，Caliskan-Islam et al.[3] の先行研究において著者推定のためにソースコードより抽出された構文的特徴，文法的特徴，また書き方の癖や嗜好を表す特徴量を参考に，計 31 種類の特徴量の抽出を行う．表 1 に使用した特徴量の一覧を示す．

4. 計算機実験

4.1. 概要

本提案手法の有効性を検証するべく，実際に初学者の書いたソースコードに対して先行研究と同様の手法，提案手法による特徴量抽出を行い比較検証をする．ソースコードは木更津工業高等専門学校情報工学科の学生より収集し，実験を行った．

4.2. 特徴量を抽出するソースコード

実験に用いるソースコードは情報工学科 4 年のプログラミング授業にて学生が課題として提出する 3 つの C 言語プログラムである．これは既存のゲームである「Hangman」,「Robots」,「Minesweeper」をそれぞれプログラムで再現するという内容である．学生はゲームの仕様とサンプルを見て，関数仕様や変数構造を各自で定義し，授業時間の他各自で作成を行う．

4.3. 実験方法

4.3.1. 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，全データを学習データとしてソースコードの評価を分類するランダムフォレストを作成する．作成されたモデル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抽出する．

4.3.2.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価を予測するランダムフォレストを作成し，重要度の高い特徴量の抽出を行う．先行研究 [2] 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，悪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．k-means 法の分割クラスタ数は先行研究と同じ 3 として実験を行った．

4.3.3.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

Extraction of Feature Value determining Programming Skills using Clustering Methods and Random Forest

†Shunsuke Iizumi, Advanced Course of Control and Information Engineering, National Institute of Technology, Kisarazu College

‡Shinichi Oeda, Department of Information and Computer Engineering, National Institute of Technology, Kisarazu College

表 1: ソースコードから抽出する特徴量

関数の個数	関数の引数の個数	関数名の平均長	関数の占める割合
変数の個数	変数名の平均長	関数の変数個数の平均	関数の変数個数の分散
関数行数の平均	関数行数の分散	インデント文字 (タブ/スペース)	空行の出現頻度
インデント方式	1 行あたりの文字数の平均	1 行あたりの文字数の分散	ネストの深さの最大値
リテラルの出現頻度	コメントの出現頻度	ヘッダーの個数	マジックナンバーの個数
タブの出現頻度	スペースの出現頻度	“if”の出現頻度	“else if”の出現頻度
“else” の出現頻度	“switch”の出現頻度	“while”の出現頻度	“for”の出現頻度
“do”の出現頻度	1 行コメントの出現頻度	複数行コメントの出現頻度	

ングする．作成された学生クラスタに属するデータ集合から教員の評価を予測するランダムフォレストを作成し，重要度の高い特徴量の抽出を行う．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うのに対し，IRM はクラスタリングの中でクラスタ数を自動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，特徴量間でも作成されるクラスタの結果より，類似した特徴量の解析を行う．

5. 実験結果

収集した 23 名のソースコードを実験に使用した．ソースコードの評価は教員 1 名と著者によりプログラムの動作，アルゴリズム，ソースコードの可読性を総合して E(最低)〜A(最高) の 5 段階評価を行い，E と D の学生，C の学生，B の学生，A の学生の 4 種類のラベル付けを行った．

5.1. ランダムフォレスト単体による実験

得られた特徴量重要度を表 2 に示す．Hangman, Minesweeper の「関数名の平均長」が重要とされたが，これは自作の関数を定義する上で分かりやすい関数名を付けれたかどうかが影響したためだと考えられる．

表 2: 重要な特徴量の上位 5 種

Hangman	関数名の平均長
Hangman	リテラルの出現頻度
Hangman	文字数の分散
Minesweeper	関数名の平均長
Hangman	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，評価が混在するクラスタ A，評価 A，評価 B の学生のみが所属するクラスタ B，評価 D の学生のみが所属するクラスタ C に分けられた．各クラスタから得られた特徴量重要度を表 3 に示す．クラスタ C は全員が同じラベルであったため，特徴量重要度が得られなかった．クラスタ B の結果より，コメントの出現頻度，1 行あたりの文字数平均が評価の高い学生を更に分ける要因となったといえる．

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 7，ソースコードのクラスタ数は 6 という結果になった．特徴量のクラスタは「タブの出現頻度」,「インデント文字」といったソースコードの体裁に関わるものや,「関数の変数個数の平均・分散」,「関数の行数の平均・分散」といった関数における特徴量が同じクラスタに所属した．ソースコードのクラスタリング結果を見ると，評価 A の学生のみが所属するクラスタが 2 つ，平均評価が B のクラスタが 2 つ，平均評価が C,D のクラスタが 2 つ得られた．ソースコードのクラスタの重要度は一例として，Robots の特徴量が大きいもの，Minesweeper の特徴量が大きいものに分かれた．

表 3: k-means クラスタの重要な特徴量の上位 5 種

クラスタ A	クラスタ B
Robots スペースの出現頻度	Robots コメントの出現頻度
Hangman 関数の変数個数の平均	Hangman 文字数の平均
Robots 関数名の平均長	Hangman “else if”の出現頻度
Robots 関数の変数個数の平均	Minesweeper 文字数の分散
Robots 関数の変数個数の分散	Robots 関数の行数の分散

6. まとめ

本研究は，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方法を提案した．実験の結果，k-means 法はランダムフォレスト単体の実験では重要度が高いとされなかった特徴量が得られ，グループごとに詳細な特徴量を得ることができた．IRM は k-means 法に比べ適切なクラスタ数を得た上で，評価が同様なクラスタからも異なる重要度の高い特徴量を算出できた．

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] S. Križanić et al., “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] A. Caliskan-Islam et al., “De-anonymizing Programmers via Code Stylometry”, 24th USENIX Security Symposium, 2015.

[4] C. Kemp et al., “Learning systems of concepts with an infinite relational model”, 21st national conference on Artificial intelligence, pp.381-388, 2006.