2ZH-04

プログラミング授業でのランダムフォレストを用いたスキルのモデル化

千枝 睦実 †　　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

プログラミング教育の初期で学ぶ事項は，各プログラミング言語によらない共通の概念やアルゴリズムに関わるものであり，学生の理解度や，授業に追従できなくなる (ドロップアウト) 原因の把握が特に重要となる．

しかしながら，ほとんどの学校では，教師が一度に 30 人以上の学生を一度に教えるため，教師が各学生の理解度を把握したり，各学生に適した指導を提供することは困難である．学生によるスキルの習得度の確認には，定期的な試験が用いられるが，これには時間がかかり，採点作業は教師の負担となるため，頻繁には実施できないため，学生の理解度のチェックが遅れることがある．

潜在的なドロップアウトを予測するとき，教師あり学習で評価器を作成するのが一般的である [1] が，学生の理解度を予測するためにコマンドログやキーログを用いた研究では，修飾キーの打鍵数と成績との間に正の相関が見られたのみであり，それ以上の知見は得られなかった [2].

また, 既存サービスである paiza[3] や testdome[4] は，就職・転職希望者がプログラミング課題を Web 上で解答すると，その結果および過程が検証され，受検者の能力をサービスを利用する企業に保証するものである．しかし，その評価基準は「複数のテストケースに対する正答率，コードの記述時間，実行速度，メモリ消費量により評価を行う」としており，コード自体の評価はされていなかった．

これらの背景を踏まえ，本研究では，学生のソースコードやコマンドログ，キーログに現れる特徴をランダムフォレストの学習過程で生成される決定木群を用いて可視化し，ドロップアウトする可能性が高い学生と低い学生を分ける原因を明らかにする．

Modeling Skill in Programming Class Using Random Forest

†Mutsumi CHIEDA, National Institute of Technology, Kisarazu College

††Shinichi OEDA, National Institute of Technology, Kisarazu College

2. システム概要

本研究で開発するシステムを図 1 に示す．授業中に取得したソースコードやログを解析して CSV 形式に変換して入力する．ランダムフォレストは入力されたデータを分類する過程で決定木群が出力する．これらの決定木群から，各学生へのアドバイスを抽出する．

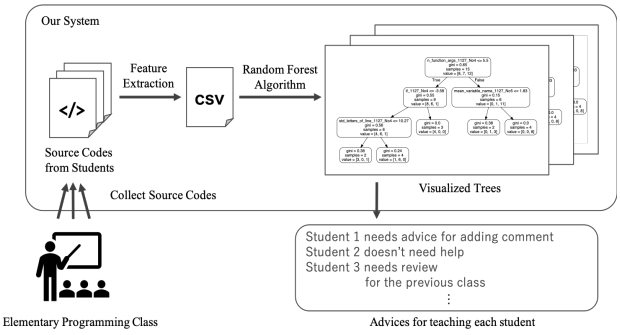

図 1　システム概要図

3. 特徴抽出

本研究では，学生の課題への答案となるソースコードやコマンドログ，キーログの 3 つを用いてランダムフォレストにより決定木群を生成する．収集したデータをランダムフォレストに適用するためには，すべての特徴が数値もしくはカテゴリカル変数である必要がある．そのため，収集したソースコードやログを互いに関連させて，いくつかの特徴に変換する．抽出する特徴は，ソースコードの著者を特定する実験で有効だったものの中で，指導において意味を持つものを採用した [5]．ソースコードから抽出した特徴を表 1 に示す．

また，コマンドログからは，エディタの呼び出しやコンパイル，実行による動作確認などの行動の頻度を抽出し特徴とし，キーログからは，修飾キーや移動キーなどの行動に関わるキーの生起頻度やキーが打たれた時期を抽出し特徴とした．

表 1　抽出した特徴 (ソースコード由来)

自作関数の出現頻度	自作関数名の平均長さ
関数の引数の数	変数名の平均長さ
インデント方式	インデント文字
1 行あたりの文字数	1 行の文字数のばらつき
空行の出現頻度	自作関数の平均長さ
各予約語の出現頻度	自作関数の平均長さ
1 行コメントの出現頻度	複数行コメントの出現頻度

4. 決定木

本研究では，教育の現場で使用されることを想定するため，機械学習に詳しくない人でもわかりやすく可視化することが不可欠となる．決定木 [6] は，木構造によって表される予測モデルである．予測結果に対する原因が，人が把握しやすい形で出力される長所を持つため，本研究に適している．

決定木を生成する際は，以下の手順に従う．

1. 全データをルートに入れる．
2. ジニ不純度 (式 1) が最小になるようにデータを分割する．

$$I_G = 1 - \sum_{i=1}^{N} p(i|t)^2 \tag{1}$$

3. ノード内のジニ不純度がある値を下回るか，木がある深さに達するまで，分割された各ノードで手順 1，2 を繰り返す．

決定木は，データを混じりけなく分割するために学習した規則を木構造の形で可視化する．決定木の各ノードは，データ分割の条件，ジニ不純度，属するサンプル数，各クラスのサンプル数，およびそのノードで最も多いサンプルのクラスを示す．

5. ランダムフォレスト

ランダムフォレスト [7] は，アンサンブル学習の手法の一つである．アンサンブル学習とは，決定木などの基本モデルを集約してより優れた予測モデルを作成する方法である．ランダムフォレストは，同じトレーニングセットの異なる部分で訓練された複数の決定木を平均化することで，モデルの汎化性能を向上させる．

本研究では，ランダムフォレストを適用する過程で生成される決定木群から，ドロップアウトの危険がある学生とそうでない学生とを分けた原因を可視化する．

6. まとめ

本研究では，ランダムフォレストでの分類を用いて学生のスキルの予測と学生の評定を分けた要因の可視化を行うことを提案した．また，可視化と同時に予測することにより，決定木群の実際のスキル状態への当てはまり具合の検証も行った．

今後は，データ収集や特徴抽出の段階を重点的に改善する．そこで，具体的な今後の課題として，より大規模なデータに対する実験や，取得したソースコードの編集履歴をさらに活用した，時系列でのソースコードの変化の考慮が挙げられる．これにより，より信頼性の高い結果や，新しい知見の獲得を目指す．

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] Gerben Dekker, Mykola Pechenizkiy, Jan Vleeshouwers, “Predicting students drop out: a case study”, Proceedings of the 2nd International Conference on Educational Data Mining (EDM2009), pp.41-50, 2009.

[2] 橋本 玄基, 清野 真理子, 大枝 真一, “プログラマのスキル評価のためのログデータ解析”, 情報処理学会第 78 回全国大会 (2016).

[3] paiza, `https://paiza.jp`

[4] testdome, `https://testdome.com`

[5] E. Dauber, A. Caliskan, R. Harang, R. Greenstadt, “Git Blame Who?: Stylistic Authorship Attribution of Small, In- complete Source Code Fragments”, EASE’19 Proceedings of the Evaluation and Assessment on Software Engineering, pp.340-345, April 2019.

[6] J. R. Quinlan, “Induction of Decision Trees”, Machine Learning, Vol. 1, Issue 1, pp.81-106 (1986).

[7] Tin Kam Ho, “Random decision forests”, ICDAR ’95 Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 1 (1995).