未完成コードを入力とするプログラミングロジック推定のための構文木分析
Syntax Trees Analysis for Estimating Programming Logics from Incomplete Codes

佐藤 光浩[*1], 大沼 亮[*1], 大波 奨[*1], 中山 祐貴[*1], 神長 裕明[*1], 宮寺 庸造[*2], 中村 勝一[*1]
Mitsuhiro SATO[*1], Ryo ONUMA[*1], Sho ONAMI[*1], Hiroki NAKAYAMA[*1],
Hiroaki KAMINAGA[*1], Youzou MIYADERA[*2], Shoichi NAKAMURA[*1]
[*1] 福島大学 共生システム理工学研究科／共生システム理工学類
[*1]Department of Computer Science and Mathematics, Fukushima University
[*2] 東京学芸大学 教育学部
[*2]Faculty of Education, Tokyo Gakugei University
Email: {mitsuhiro, onami}@cs.sss.fukushima-u.ac.jp,
{onuma, hnakayama, kami, nakamura}@sss.fukushima-u.ac.jp, miyadera@u-gakugei.ac.jp

あらまし：プログラミング演習では，一人ひとりの学習者の状況に応じた指導が重要だが，少数の教授者で多数の学習者に対応する制約下では限界がある．特に，同じ演習課題でも，プログラムの組み立て方（プログラミングロジック）は全ての学習者に一様ではなく，その把握は困難である．本研究では，最新の演習課題に対する作成途中のコードから，それぞれの学習者のプログラミングロジックを自動的に推定する手法の開発を目指す．本稿では，構文木の類似度の分析に基づくロジック推定，機械学習を用いたロジック推定の概要を示す．その上で，提案手法に基づいた実験について報告し，その結果に基づいて提案手法の特徴について考察する．
キーワード：プログラミングロジック，構文木の類似度，機械学習，プログラミング演習支援

1. はじめに

大学等におけるプログラミングの演習授業では，初学者が躓きながらコーディングに取り組む．そのため，一人ひとりの学習者の状況に応じた指導が重要であるが，一人の教授者と数名の Teaching Assistant(TA)で学習者の指導にあたる人的・時間的制約下では限界がある．特に，同じ演習課題でも，プログラムの組み立て方(プログラミングロジック)は全ての学習者に一様ではなく，その把握は困難である．

これに対して，学習者が提出したソースコードを類似度に基づいてグループ化することで共通の誤りを検出する手法(1)，学習者の提出物に対して最も類似する解答例を提示するシステム(2) などが報告されている．しかし，これら既存研究の殆どは，プログラミングロジックの違いは考慮できていない．

そこで本研究では，最新の演習課題に対する作成途中のコードから，それぞれの学習者のプログラミングロジックを自動的に推定する手法の開発を目指す．これにより，教授者が多様なロジックを大きな負担なく把握することを可能とする．

2. 問題点と支援方針

2.1 問題点

本研究では，プログラミング演習における問題点のうち，以下の 2 つに焦点を当てる．

（問題点１）一人ひとりの演習中における学習者のプログラミングロジックに応じて指導だが，少数の教授者によるロジックの把握は困難である．

（問題点２）教授者が各ロジックの学習者群を把握することは困難である．

2.2 方針

本研究では，ある演習課題において生じ得るロジックは既知であり，過去の学習者が作成したコードがロジックごとに分類されているものとする．その上で，最新の学習者が作成途中のソースコードを入力として，構文木の類似度の分析，機械学習モデル，それぞれでロジックを推定する手法を開発する（問題点 1 への対応）．また，ロジックごとの学習者群とソースコードを理解しやすい形で教授者に提示するためのシステムを開発する（問題点 2 への対応）．これにより，教授者による，学習者のロジック把握を支援する（図 1）．

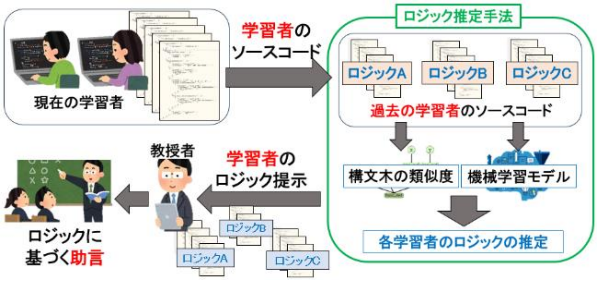

図 1 ロジック把握支援の流れ

3. 構文木の類似度分析によるロジック推定

3.1 構文木を用いたロジック推定の狙い

構文木は階層構造を持つため，プログラムの構造を正確に把握することが可能であり，各ノードの要素の関係（ノード間の親子関係や兄弟関係など）を明確にすることができる．アルゴリズムや処理の手順などを表現する性質に注目し，本手法では構文木を採用する．

3.2 構文木の類似度分析に基づくロジック推定手順

以下の手順でロジックを推定する．

(1) 現在の学習者と過去の学習者の C 言語ソースコードに対して構文木を生成（図 2）
(2) 現在の学習者の構文木と各ロジックの過去の学習者の構文木を比較して類似度を算出
(3) 算出した類似度に基づくロジック推定

構文木の生成には C 言語のコンパイラである clang を用いる．またソースコード間の類似度の算出には，TED (Tree Edit Distance)，TO (Tree Overlapping)，Tree Kernel を用いる．これらを用いて，ロジックを推定したい対象ソースコードとロジックごとの代表ソースコードをそれぞれ比較し，最も類似度が高いソースコードを対象ソースコードのロジックとする．

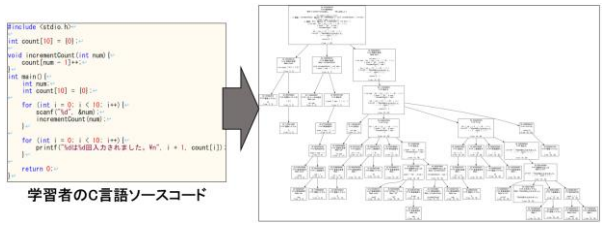

図 2 構文木の生成

4. 機械学習によるロジック推定

4.1 機械学習を用いたロジック推定の狙い

構文木はプログラムの構造を正確に把握することが可能である．しかし，同じ意味を持つが構文的に異なるコードに対して柔軟性が欠けてしまうという課題がある．そこで本手法では，ソースコードのバリエーションに対応するために，機械学習に着目する．機械学習は，大量のデータから複雑なパターンや特徴を自動的に学習し，分類することが可能である．結果として，様々なソースコードに対して柔軟な分類を行うことを可能とし，構文木の欠点を補完することができると考えられる．

4.2 機械学習モデルを用いたロジック推定手順

まず機械学習ライブラリの設計をする．続いて，教師データの準備を行い，機械学習モデルに読み込ませるための特徴量設計を行う．最後に機械学習モデルの評価として，内部検証・外部評価を行う．

機械学習用のライブラリには scikit-learn を使用する．多値分類を目的としているため，分類モデルには非線形分類器であるランダムフォレスト（RFC），非線形サポートベクトルマシン（SVM），K-近傍法（KNN），決定木（DTC）を用いることで分類モデルの有用性を検証している．

次に教師データを準備する．過去の学習者の C 言語ソースコードに人の手でラベル付けを行う．そのソースコードをそのまま用いるのではなく，ツリー構造を持つ JSON 形式のコードに変換したものを教師データとする．

続いて特徴量設計を行う．特徴量には主に，Graph2vec によりツリー形式のデータをベクトル化したものとエッジの属性情報をベクトル化したものを用いる．エッジの属性情報の取得には，データフローグラフを使用する．

上記で述べた特徴量を学習した機械学習モデルの評価には，内部検証と外部評価を行う．内部検証では，K 分割交差検証を行う．外部評価のテストデータとして学習者の演習課題における途中段階までのソースコードを用いる．

5. 実験と考察

5.1 実験概要

本研究では，ロジック推定手法の有効性検証を目的として実験を行った．具体的には，構文木の類似度の分析に基づく推定，機械学習モデルによる推定をそれぞれ実施し，その精度を確認した．機械学習については，演習課題に対する C 言語のソースコード 1000 個を教師データとした．

5.2 結果と考察

外部評価としてロジックが 2 分類の時において実験を行った．構文木の類似度を用いた推定結果，機械学習モデルの外部評価結果である F 値をそれぞれ表 1，表 2 に示す．2 分類の場合，構文木の類似度の方法では TED を用いた時，機械学習では分類モデルで RFC，SVM を用いた時に良好な結果が得られている．一方で，TO や Tree Kernel，DTC を用いた場合に振るわない結果となっており，構文木の比較の仕方，また特徴量の検討が必要である．

表 1 構文木の類似度に基づく推定の F 値

類似度算出法	TED	TO	Tree Kernel
2 分類	0.74	0.65	0.57

表 2 機械学習における外部評価の F 値

分類モデル	RFC	SVM	KNN	DTC
2 分類	0.76	0.83	0.70	0.62

6. おわりに

本稿では，構文木の類似度に基づいたロジック推定手法とツリー形式に変換したソースコードを学習データとする機械学習を用いたロジック推定手法について述べた．今後は，プロトタイプシステムを用いた提案手法の妥当性の検証と改善を進めたい．

参考文献

(1) 浦上理，長島和平，並木美太郎，兼宗進，長慎也: “プログラミング学習者の躓きの自動検出”，情報処理学会研究報告, Vol.2020-CE-154, No.4, pp.1-8 (2020)
(2) 北庄司亮，松下誠，肥後芳樹: “機械学習を用いて模範解答コードを提示する初学者向けプログラミング学習システムの構築”，情報処理学会研究報告, Vol.2023-SE-213, No.7, pp.1-8 (2023)