K-004

コード片の類似性にもとづく学習者の理解不足点の推定
Estimating Learning Failures of Students With Similarity Between Code Fragments

西本 航 † 梶原 祐輔 ‡ 島川 博光 ‡
Wataru Nishimoto Yusuke Kajiwara Hiromitsu Shimakawa

1. はじめに

多くの大学でプログラミングの演習授業が実施されている．プログラミング演習では一人一人の学生が課題のプログラムを実装していく形式が取られている．学生には新しく習った知識を使いプログラムを実装することが求められている．しかし, 新しい知識の利用は既知の知識の利用に比べ困難であるため, 既知の知識のみを使いプログラムを実装する学生がいる．このような実装はソースコードやプログラムの振舞としては間違っていないが, 課題の解答としては適切ではない．そのため, 指導者はこのようなソースコードを書く学生に対して新しい知識を使うように指導しなければならない．しかし, 学生がなぜ新しい知識の利用を諦め, 既知の知識のみを利用したのかを特定することは困難である．

本論文では学生が何が理解できず, 既知の知識のみを使ったプログラムを書いているかを推定する手法を提案する．本手法により指導者は学生に指導すべき内容を容易に推測できる．

2. プログラミング演習授業における問題点

2.1 学習者の理解状況特定の必要性

プログラミング演習において, 新しい知識の使用を諦め既知の知識のみを使い課題に取り組む学生がいる．彼らの書くソースコードやプログラムの振舞は課題の解答として間違ってはいない．しかし, 課題を解くことによって獲得すると期待されている知識の獲得はできていない．学生は何かが理解できていないため, 新しい知識の利用をあきらめ, 既知の知識を使ったと考えられる．本論文ではこの理解できていない箇所を理解不足点と呼ぶ．学生の理解不足点を突き止め, 指導する必要がある．

既知の知識のみを使い書かれたプログラムには作法に則らない無作法なコードが使われる．一塊の無作法なコードを無作法コード片と呼ぶ．学生はプログラミング演習の解答として無作法コード片を使うべきではない．したがって, 無作法コード片を使う学生に対して指導を与えなければならない．しかし, なぜ無作法コード片を使うのかを特定できなければ指導できない．そのためには, 無作法コード片を使う学生の理解不足点を特定する必要がある．

2.2 既存研究

学生の理解状況を特定するためにコンパイラのエラーメッセージやプログラムの関数呼出しの記録を解析した研究がある [1][2]．これらの研究は, コードのエラーやプログラムの振舞いに関するものである．そのためプログラムが適切なコードで書かれているかは評価しない．これらの研究では, 学生の書いたソースコードに無作法コード片が含まれるときに学生の理解不足点を推定し, 学生に正しい知識を獲得される指導については論じられていない．

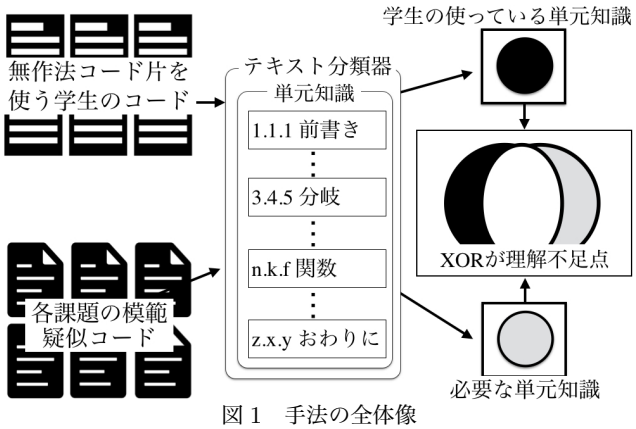

図 1 手法の全体像

3. 無作法コード片を使う学生の理解不足点

3.1 テキスト分類器による理解不足点の特定

無作法コード片を含むソースコードを書く学生の理解不足点を学生の書いたソースコードと模範解答から推定する手法を提案する．図 1 に手法の全体像を示す．本手法では, 学生が参照する教科書全文をテキストデータとして利用する．教科書のテキストデータを各節ごとに分割し教師データとしてテキスト分類器に学習させる．このテキスト分類器は与えられた文章と各節の関連度合いを確率で評価する．以降各節を単元知識と呼ぶ．

まず, ある課題のコメントおよびコードからなる模範解答を, テキスト分類器に評価させ, 単元知識の分類確率を算出する．このとき, ある閾値以上の確率をもつ単元知識の集合を抽出する．

つぎに, その課題の無作法コード片を含むソースコードを評価させ, 同様に閾値以上の確率をもつ単元知識の集合を抽出する．これらのソースコードには, 同一の課題について書かれたものであるため, 共通する単元知識が存在する．共通する単元知識は学生が理解している, もしくは, 正しく使えている内容であると考えられる．

逆に, 評価結果模範解答から得られた単元知識の集合と, 無作法コード片を含むソースコードから得られた単元知識の集合の排他的論理和にあたる部分は, あるべき単元知識が現れていないか, あってはならない単元知識が現れていることを意味する．これらが学生の理解不足点であると考えられる．

3.2 テキスト分類器の構築

学習者の書いたソースコードと模範解答を解析するために, テキスト分類器を構築する．テキスト分類器の分類アルゴリズムにはナイーブベイズ分類 [3] を用いる．教師データとして, プログラミング演習の授業で採用されている教科書の各節を使う．各節の文章をわかち書きかつ基本

† 立命館大学情報理工学研究科
‡ 立命館大学情報理工学部

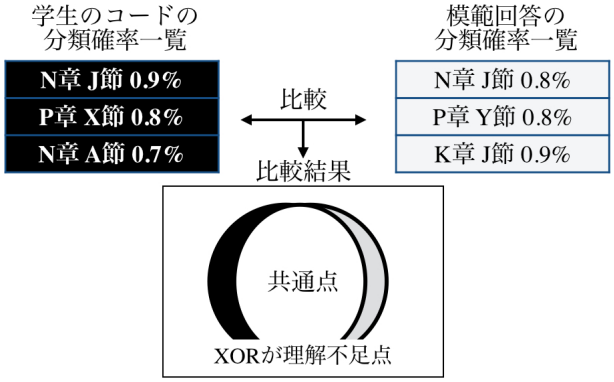

図 2　理解状況の比較

形に変換し, 各単語の出現頻度を計算する. 節ごとの計算結果を教師データとする.

このテキスト分類器は文章があたえられたとき, その文章がある節に分類される確率をすべての節について計算する. 本論文では分類確率がその文章と各節の関連度合いとみなす. ある節と関連度合いの高い文章はその節の内容によって文章の特徴を説明できる. ここで文章として, 模範解答と学生の書いたソースコードをテキスト分類器に与えることを考える.

テキスト分類器で模範解答やソースコードを評価できる理由として次のようなものがある. プログラミング演習授業で扱う内容は教科書をもとに作成されているため内容が似ている. 課題で作成するソースコードは簡単なものであるため, 一つのソースコードで使われる文法要素や関数などの構成要素が限られている. 多くの教科書は,1 つの節には 1 つの内容しかない.

3.3　理解状況の評価

テキスト分類器により算出された模範解答の分類確率一覧と学生の書いた無作法コード片を含むソースコードの分類確率の一覧を比較する. 分類確率はすべての単元知識について計算されるため, 課題とは関係のないものを取り除く必要がある. 取り除いたものを教師データとしてテキスト分類器を学習させる.

模範解答の分類確率評価は, 模範解答の解答に必要となる単元知識が何なのかを確率的に表現している. したがって, 模範解答のプログラムを実装するためには分類確率の高い単元知識が必要となる. 学生の書いたソースコードを分類させた場合, 分類確率は学生のソースコードで使われている単元知識となる.

図 2 に示すように, これらは同一の課題について書かれたソースコードであるため共通部分が多くあると考えられる. 共通部分の単元知識を, 学生は既に使えているため, 現時点では大きな問題はない.

排他的論理和にあたる部分が学生にとって問題のある単元知識の集合になる. 学生が書いた無作法コード片を含むソースコードのみで分類確率が高い単元知識は学生が固執してしまっている単元知識である. 模範解答でのみ分類確率が高い単元知識は学生が使えていない理解不足点である. このようにして, 学生の書いた無作法コード片を含むソースコードから学生の理解不足点を推定できる.

4.　テキスト分類器による模範解答の評価

手法適用のための前処理としてテキスト分類器によって模範解答を評価したときに, 適切な節の分類確率が高くなることを確認するための評価テストを実施した. 「苦しんで覚える C 言語」[4] を教科書の例としてテキスト分類器を学習させた. この教科書の 198 節の文章を教師データとして学習させる.

課題の例として入力された値に対して条件分岐させる課題および自作関数を使い 1 次元ベクトルの内積を計算させる課題を評価した. これら課題では課題の解答に必要となる関連知識を教員が事前に定義している. 条件分岐させる課題では, 入出力, 条件分岐, 数学的表記, 浮動小数点型,C 言語プログラムの基本構造, 基本的な関数の利用, 変数,if-else などが定義されている. 自作関数を使った内積計算には, 入出力, ループ構文, ループ終了条件, 配列, 戻り値, 数学的表記, 浮動小数点型,if-else, 配列の初期化, 引数, マクロなどが定義されている. これらの課題の模範解答をテキスト分類器にかけ, 単元知識の分類確率を算出する.

分類確率の高い 10 の単元知識のみ参照した. その結果, 条件分岐させる課題の場合 8 個中 6 個が単元知識として出現した. 出現しなかった関連知識は数学的表記と C 言語プログラムの基本構造であった. 内積の計算をさせる課題では 11 個中 8 個が出現した. 出現しなかった関連知識は数学的表記, 浮動小数点型,define マクロであった. これらの関連知識が単元知識として出現しなかった理由としてこれらは一般的な内容であることが考えられる. 一般的な内容の場合, それらについては特徴が出ないため分類時に影響がでない. そのため, これらの関連知識が出現しなかったと考えられる.

以上より本テキスト分類器を用いることで, 各課題の文法的, 構文的特徴を単元知識として概ね表現できた.

5.　おわりに

本論文では学生の理解不足点を模範解答および無作法コード片を含むソースコードをテキスト分類器で評価し, 学生の理解不足点を推定する手法を提案した. テキスト分類器が適切に模範解答と関連する単元知識を推定できることを確認した. 今後の課題として, 学生のソースコードおよび模範解答の評価結果を比較し理解不足点を推定する必要がある.

参考文献

[1] 西 輝之, 劉 渤江, 横田 一正, デバッガの連携による C 言語学習支援システムの提案, 電子情報通信学会技術研究報告. ET, 教育工学,Vol.106, No.583, pp.173-178, 2007.

[2] 谷川 紘平, フォン ディン ドン, 原田 史子, 島川 博光 C 言語関数呼出しの記録を用いた演習過程での習得項目の把握 電子情報通信学会論文誌. D, 情報・システム J95-D(12), 2079-2089, 2012

[3] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[4] MMGames, 森口将憲, 苦しんで覚える C 言語, https://9cguide.appspot.com, 2011.