

ジニ係数を用いた空欄の有効性評価による空欄補充問題の洗練

Evaluation of Blank Effectiveness
Using Gini Index in Fill-in-Blank Tests五島僚佑[†]
Ryosuke Goshima浅井創[‡]
So Asai島川博光[†]
Hiromitsu Shimakawa

1. はじめに

プログラミングの理解度を判定するために学習者がスクラッチから書いたプログラムが評価されることが多い。しかし、少数の指導者が多数の学習者のソースコードを採点するには相当な時間を要し、すぐに理解度を明らかにすることができない。そのため、指導者は学習者の理解度を把握するために空欄補充問題を用いることがある。

空欄補充問題は採点箇所を限定することができるため採点が容易である。空欄補充問題は学習者の理解度を高精度に判定できることが求められる。そのために空欄は適切な位置に設けなければならない。

本研究では、指導者が生成した空欄補充問題において、学習者の理解の有無を判別できる空欄を検出する。本手法を用いることで、空欄補充問題において有効な空欄を検出できるため、空欄補充問題を洗練することができる。

2. プログラミング学習

2.1 各問題形式の学習効果

プログラミング学習では学習者の理解度を測るために試験が実施される。その問題形式としておもに次の3種類の問題形式が用いられる。

全文記述問題 学習者は問題文や条件を読み、それらを満たすソースコードをすべて記述する。指導者が採点することで、学習者の理解度を細かい点まで判定することができるが、回答の多様性があり、採点が難しい。また、少数の指導者が多数の学習者に指導する場合、採点に相当の時間を要し、すぐに学習者の理解度を明らかにすることができない。

空欄補充問題 学習者はソースコードに設けられた空欄にプログラムとして正しく動作するコードを記述する。設定された空欄は、データや制御の流れを遮断する。空欄に補充することは、プログラムが正しく動作するように遮断された部分を補完することである。学習者は、見えているソースコードからデータや制御の流れに関する情報を集めることが必要となる。空欄以外の部分に対する理解がなければ記述できないような空欄であると、理解のある学習者のみが正答できる問題となる。なお、採点箇所を限定することができるので、全文記述問題より採点が容易である。

選択問題 学習者はソースコード中に設けられた空欄にプログラムとして正しく動作するコードを選択肢から選ぶ。空欄補充問題と似ているが、選択肢が見えることで回答を憶測されるため、理解度の推定が難しい。

本研究では、採点が容易かつ理解度の推定ができる空欄補充問題を扱う。

2.2 空欄補充問題の生成

空欄補充問題の作問は、空欄に設定する位置が重要となる。100人中100人が正答できるような位置に空欄を設定した問題は、全員が正答してしまっているため、学習者を理解の有無で判別することができない。一方、ほぼ半数が正答できるような位置に空欄を設定した問題は、学習者を理解の有無で二等分できるため、理解の有無を判別するうえで有効である。

どの部分を空欄にするべきかを自動で決定し、空欄補充問題を生成する手法がいくつか提案されている [4, 3, 5]。柏原ら [4] は、プログラムにおける処理の意味内容には踏み込まず処理の要所を同定し、形式的に空欄を設定する手法を提案している。内田 [3] は、キーワードや識別子を自動的に抽出し、空欄に設定する手法を提案している。設定する空欄は毎回ランダムであるため、学習者は反復練習ができる。しかし、これらは学習者や指導者それぞれの状況が考慮されていない。有安ら [5] は、指導者の意図に沿った演習のための空欄補充問題の生成手法を提案している。また、学習者の回答結果により理解が不十分なところを見つけ、次の出題時に各学習者に対し、おのおのの苦手な部分の問題を重点的に学習させるフィードバック機能も構築している。しかし、空欄補充問題の質は保持されず、新しい学習者に対してかならずはじめに学習者に適応していない空欄補充問題を解かさなければならない。空欄の有効性の評価により問題を洗練し、新しい学習者にも質の高い空欄補充問題を提供する必要がある。

3. 空欄の有効性の評価手法

3.1 理解度を判定可能な空欄の選択

本研究では、数ある空欄の中から学習者の理解の有無を判別可能な空欄を検出する手法を提案する。手法の全体像を図1に示す。過去の学習者が全文記述問題で書いたソースコードおよび空欄補充問題の回答履歴を用いる。本研究では同じ教育機関における過去の学習者と新しい学習者には同じ傾向があると仮定する。全文記述問題で書いたソースコードの正誤を目的変数、空欄補充問題の回答履歴を説明変数として決定木を生成する。ソースコードの正誤は、指導者により評価される。実行結果が正しいことだけではなく、ソースコードが指導者の想定範囲内である必要がある。全文記述問題において、ソースコードが正しい学生はその単元に関して理解があり、そうでない学生は理解がないとする。決定木生成アルゴリズムは、最も不純度が低くなるようにノードを分岐させ、それ以上低くならないノードで分岐を終了する。よって、決定木の根元に近い分岐に用いられる問題は、学習者の理解の有無を正確

[†]立命館大学情報理工学部[‡]立命館大学大学院情報理工学研究科

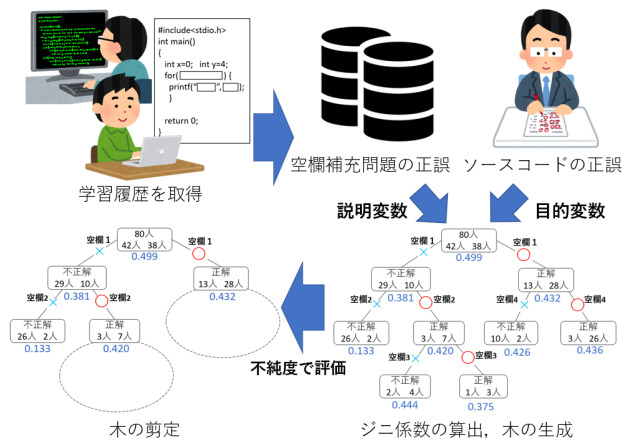


図 1: 手法概要図

に判別できる問題となる。本手法を用いることで、学習者の理解を測るのに有効な空欄を見つけることができ、空欄補充問題の質が洗練される。これにより、新しい学習者の空欄補充問題の正誤だけで、学習者が全問記述問題を回答するときと同様の理解度を測れる。

3.2 学習履歴の取得

学習者の理解の有無を判別する決定木を生成するために、過去の学習者の学習履歴を取得する。全問記述問題の正誤と空欄補充問題の回答状況を同定するために、学習履歴として学習者が全文記述問題で書いたソースコードおよび空欄補充問題の回答の正誤を取得する。全文記述問題のソースコードは指導者により評価される。評価は実行結果の正誤だけではなく、指導者の想定範囲内のソースコードであることに基づく。例えば、ポインタの知識を問う問題であるにもかかわらず、配列を用いて回答する学習者はその問題が不正解であると評価される。そのような学生はポインタに対して理解がないものとして扱われる。空欄補充問題は学習单元ごとに課題が独立しており、それぞれの課題は1~17箇所の空欄が設定されている。これは、学習者の理解の有無を学習单元ごとに取得するためである。

3.3 決定木の作成

分類や予測をするために決定木分析が広く使われる。決定木分析はモデルの生成過程が明らかでないため、理解しやすいルールが生成される。また、計算時間が極めて短く、予測や分類においてどの項目が最も重要かを明確に示すことができる。本手法では、指導者によって評価されたソースコードの正誤を目的変数、空欄補充問題の正誤を説明変数として決定木を生成する。なお空欄補充問題は学習单元ごとに課題が独立しているため、決定木は課題ごとに生成する。決定木の分岐では、分岐により学習者を二等分できる説明変数を選ぶことが重要である。例えば、100人の学習者を理解の有無で分けるさい、10人と90人に分けるより50人と50人に分けるような説明変数を設定する。また、生成された決定木の各葉ノードにはそれぞれ同様な理解度を持つ学習者が分類される。

3.4 ジニ係数による評価

本研究では、決定木の不純度としてジニ係数を用いる。ジニ係数はノードにおける不純度を定量的に表す。決定木生成アルゴリズムで用いられる不純度の尺度として、ジニ係数と情報エントロピーがある。これらにより生成された決定木に明確な差はみられないが、ジニ係数は計算が比較的容易である [1, 2]。ジニ係数 G は t を分岐前のノード、 i を分岐後のノード、 p を分割された学習者がノードに属する比率として次式で求められる [1]。

$$G = 1 - \sum_{i=1}^2 [p(i | t)]^2$$

ジニ係数の値が大きい場合は、ノード内に理解のある学習者とそうでない学習者が混在している状態である。この場合は学習者の理解の有無を正確に分類できていない状態である。ジニ係数の値が小さい場合は、ノード内に理解のある学習者もしくはそうでない学習者のうち片方だけが集中的に集まっている状態である。この場合は学習者の理解の有無を正確に分類できている状態である。決定木は最も不純度が低くなるように分岐させているので、根元では分岐前後のジニ係数に大きな差が生じる。分岐前後のジニ係数の差を見ることで、空欄の有効性を評価できる。ジニ係数の差が大きい空欄は、学習者の理解の有無を判別できている空欄として評価され、差が小さい空欄は判別できていない空欄として評価される。また、ジニ係数の差が閾値を下回っていた場合、決定木をその子孫ノードで剪定する。これにより、学習者の理解の有無を判別できていない空欄が除かれ、空欄補充問題の質が洗練される。

4. おわりに

本論文では、空欄補充問題における空欄の有効性を評価する手法を提案した。本手法を用いることで、空欄補充問題において有効な空欄を検出できるため、空欄補充問題を洗練することができる。今後は、本手法の有用性を実験により検証する。

参考文献

- [1] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.
- [2] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, 2004.
- [3] 内田保雄, "初級プログラミング学習のための自動作問システム," 情報処理学会研究報告コンピュータと教育 (CE), vol. 2007, no. 123, pp. 109–113, 2007.
- [4] 柏原昭博, 久米井邦貴, 梅野浩司, and 豊田順一, "プログラム空欄補充問題の作成とその評価," 人工知能学会論文誌, vol. 16, no. 4, pp. 384–391, 2001.
- [5] 有安浩平, 池田絵里, 岡本辰夫, 國島丈生, and 横田一正, "学習者に合わせた c 言語演習穴埋め問題の自動生成," in *DEIM Forum 2009*, vol. 5, 2009.