

個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援

田口 浩[†] 糸賀 裕 弥^{††} 毛利 公 一^{††}
山本 哲 男^{††} 島川 博 光^{††}

本論文では、プログラミング教育において、学習者ひとりひとりの理解状況に応じて演習課題を出題する手法を提案し、それを実際のプログラミング演習科目へ適用した結果を示す。本手法では、豊富に用意された演習課題の中から、過去の他の学習者の演習履歴に基づいて各演習課題の達成度を推測し、各学習者に最適な演習課題を選出して出題する。学習意欲が低い学習者には学習の継続を、学習意欲が高い学習者にはプログラミング能力の向上を優先して出題を行うので、本手法は両者に対して効果的である。本手法を大学でのC言語プログラミング演習科目に適用した結果、選出された演習課題を解いた学習者の学習継続率が11.3ポイント向上し、理解をより深めさせることもできた。

Programming Training of Students According to Individual Understanding and Attitude

HIROSHI TAGUCHI,[†] HIROYA ITOGA,^{††} KOICHI MOURI,^{††}
TETSUO YAMAMOTO^{††} and HIROMITSU SHIMAKAWA^{††}

This paper proposes a method to recommend a C-language programming exercise to each student according to its understanding. The understanding of each student for a specific programming exercise is assumed to be evaluated by a score. For a specific student, a score of a specific programming exercise is inferred, using score histories of himself and other students who addressed the same programming exercise sets in the past. The method selects an appropriate programming exercise to each student, based on the inferred score and the attitude toward the programming. It recommends an easy one to a student losing motivation, while a tough one to a positive student. The application of the method to an actual university course has proved that students addressing recommended programming exercises are superior to ones rejecting the recommendation in the exercise continuity by 11.3 points, as well as the understanding.

1. はじめに

現代の高度情報化社会が求める情報処理技術者にとって、実用的なプログラミング能力は不可欠であり、大学をはじめとする教育機関でプログラミング技法を学ぶ学生が急増している。これにともない、的確かつ効率的なプログラミング教育の必要性が増している。

汎用性が高く、オペレーティングシステムの記述からアプリケーションの開発まで幅広い分野で用いられているC言語のプログラミング技法は多くの教育機

関で教えられている。しかし、C言語にはポインタや構造体といった特有の概念が存在し、それらを含んだ学習内容は初級者が学習を進めるうえで大きな障壁となっている。そのため、教員が個々の学習者の理解状況を正確に把握し、理解に苦しんでいる学習者を中心にきめ細かい指導を行わなければならない。

現在の大学などにおけるプログラミング演習科目では、30~40名程度の学習者からなるクラスを1名の教員と数名のティーチング・アシスタントで指導することが多い。そして、ほとんどの場合、すべての学習者に対して同一の演習課題が課されており、個々の学習者に合わせた指導が行われているとはいえない。そのため、クラスで共通に進められる演習内容に理解が追いつかず、学習に対する意欲を失い、挫折してしまう学習者が少なくない。

この問題点を解決するためには、学習者ごとに理解

[†] 立命館大学大学院理工学研究科
Graduate School of Science and Engineering,
Ritsumeikan University

^{††} 立命館大学情報理工学部
Department of Information Science and Engineering,
Ritsumeikan University

状況を把握したうえで、最も学習効果が高まると思われる演習の進め方を判断し、それに沿って個別に最適な演習課題を出題することが望ましい。しかし、プログラミング教育における教員の負担は、現状においても相当大きく、これに加え、各々の学習者に最適な演習課題を選ぶ負荷を課すのは現実的でない。

本論文では、教員に多大な負担を強いることなく、個人に合わせた C 言語プログラミング教育を実現するために、それぞれの学習者にとって最も学習効果が高まると思われる演習課題を個別に選出して出題する手法を提案する。我々の手法ではあらかじめ、学習内容ごとに難易度に幅を持つ演習課題を豊富に用意しておく。学習者が新しい学習内容に取り組みとうとするさいに、その学習内容についての演習課題の中から 1 題が選出される。選出のためにまず、出題の候補となる各演習課題に対する学習者の達成度が推測される。そして、学習者がこれまでに作成したプログラムなどから学習意欲の高低を判定し、その結果に基づいて出題課題を決定する。具体的には、学習意欲が低ければ、推測される達成度が最も高い、その学習者にとってやさしいと思われる演習課題を出題する。学習意欲が高ければ、推測される達成度が最も低い、その学習者にとって難しいと思われる演習課題を出題する。この手法は、

- 個々の基礎的学習内容やそれらの組合せに対する学習者の理解状況を反映した演習課題の達成度を、過去の他の学習者の演習履歴と照らし合わせることで、細かい粒度で推測する、
- 学習意欲が低下しつつある学習者には、理解しやすいと推測される演習課題を与え、学習意欲を失わずに学習を継続させることができる、
- 学習意欲が高い学習者には、プログラミング能力の向上を期待して、難しく、それゆえ、やりがいがある演習課題を割り当てる、

という特長を持ち、学習者ひとりひとりに合わせた効果的なプログラミング演習の実施が可能となる。新たな演習課題を出題するさいに、毎回学習意欲の高低を判定するので、学習意欲が低下しつつある学習者を早期に発見し、対応することができる。我々は本手法を用いて演習課題を出題する機能を持つプログラミング演習支援システムを構築し、大学での C 言語プログラミング演習科目に適用した。その結果、本手法によって選出された演習課題を解いた場合のほうが、それ以外の演習課題を解いた場合に比べ、学習者が継続して学習を行った割合が 11.3 ポイント向上した。さらに、選出された演習課題を解いた回数が多い学習者ほど、演習全体を通して達成度が高かった。

以下、2 章で C 言語プログラミングの学習内容と、実際の学習者に対して実施した学習履歴調査について述べる。3 章では出題する演習課題の選出手法を説明し、4 章で本手法を用いたプログラミング演習支援システムの実現法について述べる。5 章では大学で実際に開講されているプログラミング演習科目で本手法を適用した結果を示す。そして、6 章で関連研究との比較を行ったうえで、最後にまとめを行う。

2. C 言語プログラミング学習

2.1 C 言語の基礎的学習内容とその結合

文献 1) によれば、C 言語プログラミングの基礎的な学習内容は、図 1 に示すように、8 つの単元に大きく分けられる。C 言語プログラミングを初めて学ぶ学習者はまず、これらの単元を順に学習する必要がある。基礎的な内容を一通り学習した学習者はプログラムを作成することができるようになるが、この時点で作成できるプログラムは、単純な処理を行うだけの簡単なものに限られ、一般的な問題を解決できるものではない。これは、学習者は「配列」や「ポインタ」、「構造体」といった個々の内容は理解できていても、それらを組み合わせる扱おうことがまだできないからである。

一般的な問題を解決するデータ構造やアルゴリズムの実現のためには、学習者は複数の基礎的内容を連携させる手法も学習しなければならない。問題解決のための手法の実現を実習することによって、学習者は個々の基礎的学習内容を深く理解し、それらを有機的に結合させることを学ぶ。したがって、問題解決手法の学習の過程において、演習課題を豊富に用意し、その中から各学習者に対して、理解状況に応じた最適な演習課題を与えることが求められる。

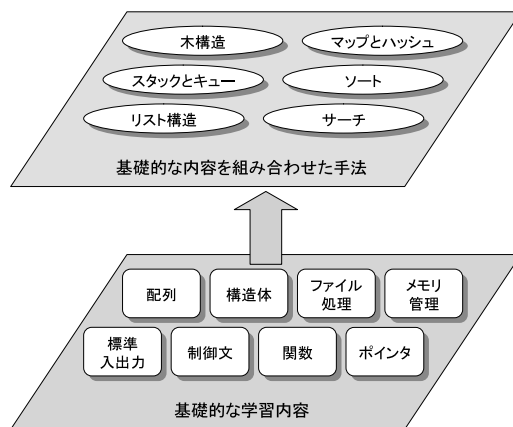


図 1 C 言語プログラミングの学習内容

Fig. 1 Contents of C programming exercise courses.

2.2 演習科目における学習履歴調査の実施

プログラミング教育では、学習者の理解を深め、プログラミング能力を向上させることが最も重要であるが、そのためには、学習者の学習意欲を高く維持することが必要である。ゆえに、学習者の理解状況に応じた最適な演習課題を選定するさいも、学習者の学習意欲を低下させないように考慮しなければならない。個々の学習者に対して個別に最適な演習課題を出題する手法を考察するため、大学で実際に開講されているC言語プログラミング演習科目における受講者の理解状況とその履歴を調査した。

調査は、2003年9月から2004年1月にかけて立命館大学理工学部情報学科で開講された1回生配当科目「情報処理演習」のある1クラスの受講者46名に対して実施した。この科目はC言語プログラミングの初心者を対象として開講されており、図1に示したC言語の基礎的な学習内容のうち、「メモリ管理」単元を除く7つの単元について、1単元あたり1~3週かけてプログラミング演習を行い、全15週の演習でそれらの学習内容を一通り学習する。ここでは個人別の演習課題の出題は行わず、すべての受講者に共通の演習課題を各週6~10題ずつ出題し、次週までにプログラムを作成させた。プログラム提出時に、各演習課題に対する自己評価を受講者に回答してもらった。自己評価は

- (1) 自信満々
- (2) 大体理解できたと思う
- (3) 自力で完成させたが、理解できたか不安
- (4) 教員らから助言を受けて何とか完成させた
- (5) 何をすればよいのか分からなかった

の5つの選択肢の中から最もあてはまるものを1つだけ選択してもらうという形式をとった。

学習履歴調査で得られたデータより、まず、C言語プログラミングの初心者である受講者がどの単元で理解に苦しんでいるのかを調べた。5つの選択肢のうち、選択肢(3)、(4)、(5)が選択された場合は、受講者が理解に苦しんでいると考えられる。図2のグラフは、各単元の演習課題に対する自己評価として、選択肢(3)、(4)、(5)のいずれかが選択された割合を示したものである。このグラフより、「ポインタ」単元で特にその割合が高いことが分かる。

理解に行き詰まっている学習者は学習意欲が低下しつつある可能性が高く、その状態が長く続けば、学習意欲を完全に失ってしまいかねない。そこで次に、理

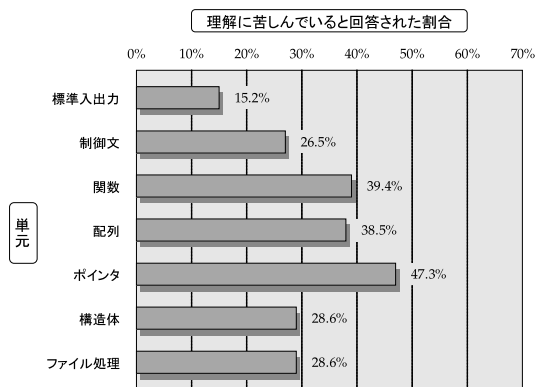


図2 学習履歴調査結果

Fig. 2 Findings in a programming exercise class.

解に苦しんでいる受講者の多い「ポインタ」単元に焦点を絞って調査結果の分析を行う。「ポインタ」単元のいずれかの演習課題に対して選択肢(5)を選択した受講者は13名であった。これらの受講者が提出した当該課題のプログラムを見てみると、大きく以下の2種類のパターンに分かれるということが分かった。

- ほぼ白紙同然か、講義資料や教科書に書かれている例題の一部をそのまま写して提出している。
- 未完成だったり、誤っていたりするものの、自分なりにプログラムを作成して提出している。

前者の多くは、その後の演習課題に対しても同様の状態のプログラムを提出しているか、それを最後にプログラムを提出しなくなっていた。これは、「ポインタ」単元の学習内容の理解に行き詰まってしまったため、プログラミング学習に対する意欲が失われたものと考えられる。よって、このような学習者に対しては、できるだけ理解に行き詰まらないように、きわめて理解しやすい演習課題を出題するほうが効果的であると推測される。一方、後者は、理解に苦しんではいないものの、学習意欲は低下しておらず、それ以降の演習にも積極的に取り組み、理解しようと試みている場合がほとんどであった。したがって、このような学習者に対しては、難しく、やりがいがある演習課題を出題し、高い学習意欲を利用するほうが効果的であると推測される。

3. 協調フィルタリングを用いた課題の出題

3.1 協調フィルタリングによる達成度の推測

種々の基礎的な学習内容を組み合わせるべきプログラミング演習課題の達成度には、ある特定の演習課題Aの達成度が低い学習者は、似た学習内容を用いる演習課題Bの達成度も高くないといった傾向が存在す

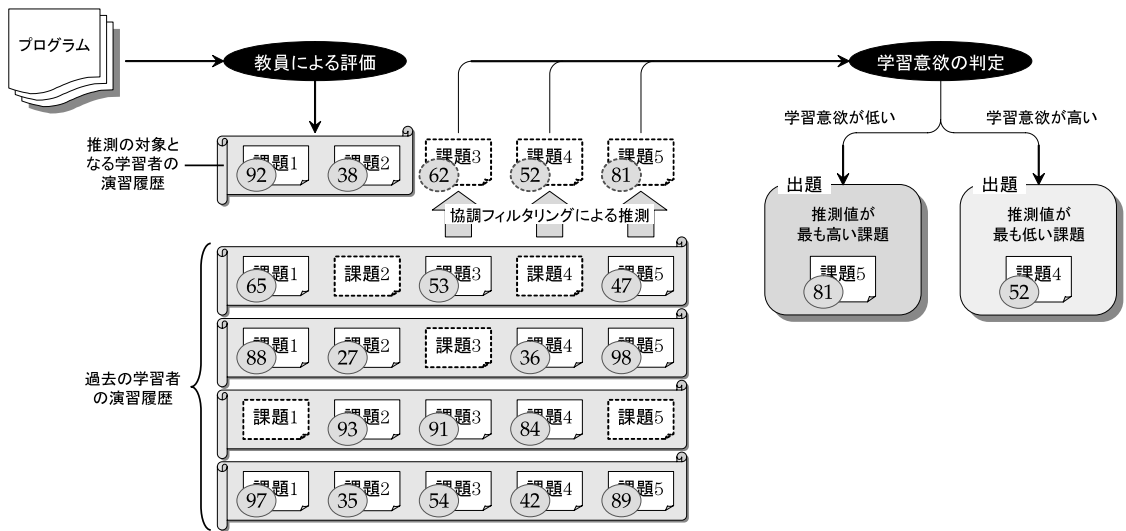


図3 協調フィルタリングを用いた演習課題の出題

Fig. 3 Selection of the most appropriate exercise with the collaborative filtering algorithm.

ると考えられる。また、特定の学習内容を苦手とする学習者は、総じて、その学習内容の理解が必須となる演習課題を解けないという傾向がある。たとえば、ポインタを苦手とする学生は、リスト構造の演習課題を解くことに苦しむ。そこで、同じ演習内容でプログラミングを学習した過去の学習者のデータに基づいて、現在学習を進めている学習者の傾向を推測したうえで、各学習者にとって最も学習効果が高まる演習の進め方を判断することを考える。

ユーザの物事に対する傾向や嗜好を過去の行動という形で記録し、そのユーザと似たような行動をとっている他のユーザの情報をもとに、そのユーザの今後の傾向や嗜好を推測するための手法として、協調フィルタリング²⁾がある。この手法を用いた、書籍の購入を支援する Web サイトなどが実用化されている³⁾。

我々の提案する手法では、あらかじめ、「単方向リスト構造」や「クイックソート」といった学習内容ごとに難易度や実現方法の異なる演習課題を豊富に用意しておく。学習が始まると、学習者は与えられた演習課題に対してプログラムを作成し、完成したプログラムを自己評価とともに提出する。一方、教員は学習者が提出したプログラムから学習者の理解状況と学習への取り組み姿勢を評価する。豊富に用意された演習課題の中から出題する演習課題を選定するために、まず、図3に示すように、学習者がこれから取り組む学習内容に対応する各演習課題の達成度を、既存の協調フィルタリング手法を用いて推測する。

協調フィルタリングによる推測を行うためには、ま

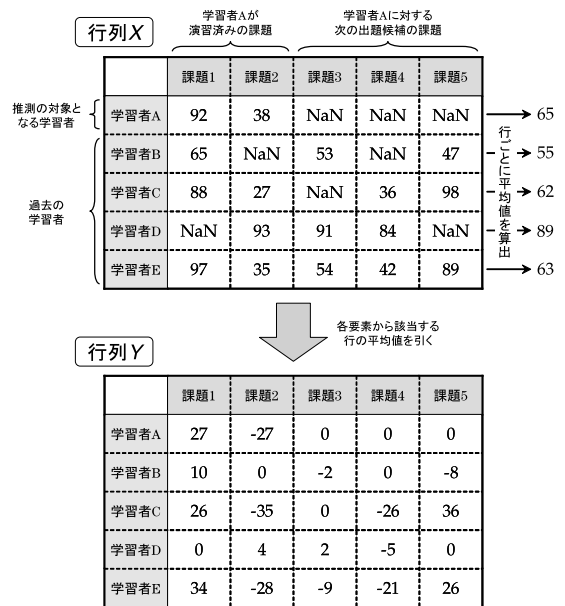


図4 行列の例

Fig. 4 Example of matrices.

ず、推測の対象となる学習者のこれまでの演習履歴と、過去の他の学習者の演習履歴から、図4に示すような、行を学習者、列を演習課題、要素を得点とする $m \times n$ 行列 X を作成する。ここで、 m は学習者数、 n は演習課題数である。得点とは、教員による評価の結果に基づき、演習課題に対する学習者の達成度を数値化したものである。得点は0以上100以下の値をとる。ただし、未出題の演習課題については、その得点を、未

確定を表す NaN (Not a Number) 値とする。

学習者の中には、どの演習課題の得点も比較的高い者もいれば、全般的に得点の低い者も存在するため、実際の得点の値そのものでは学習者間の得点の類似性を表すことができない。そこで、図 4 のように、学習者ごとに得点の平均値を求め、各演習課題の得点から平均値を引いた値によって学習者のさまざまな演習課題にわたっての得点推移を表現する。つまり、行列 X の行ごとに要素の平均値を算出し、各要素から該当する行の平均値を引いた行列 Y を作成する。ただし、平均を計算するさいは、NaN 値は計算の対象外とする。また、行列 X において NaN 値であった要素に対応する行列 Y の要素は 0 とする。したがって、行列 X, Y の各要素をそれぞれ x_{ij}, y_{ij} としたとき、 y_{ij} は以下の数式によって求められる。

$$y_{ij} = \begin{cases} 0 & (x_{ij} = \text{NaN 値}) \\ x_{ij} - a_i & (x_{ij} \neq \text{NaN 値}) \end{cases}$$

ここで、 a_i は行列 X の第 i 行の NaN 値でない要素の平均値である。

次に、この行列 Y をもとに、協調フィルタリングを用いて、出題候補となっている各演習課題の得点を推測する。処理の概要は以下のとおりである。

- (1) 行列 Y の各行は、それぞれの学習者の特徴を表すベクトルであり、それら m 個の行ベクトルが構成する部分空間がすべての学習者の特徴を表す。特異値分解⁴⁾を用いてこの部分空間をその特徴を表す m 個以下の行ベクトルで構成される空間で近似する。
- (2) 推測の対象となる学習者について、行列 X の行に相当するベクトルが与えられたとき、その各要素からその行の平均値を引いた行ベクトル t を作成する。このとき、NaN 値の扱いは上述のとおりとする。その学習者がすでに解いた演習課題の得点は変化しないものとする、 m 次の空間内で t の動きうる範囲は限定される。そして、未出題の演習課題の得点がとる値に制限はないものとしたうえで、 t が動きうる点のうち、(1) で近似した空間に最も近い点を最小 2 乗法⁴⁾を用いて求める。この求められた点を t が今後到達する点であると見なす。その点のベクトルの各要素に、最初に与えられた行の要素の平均値を加え、未出題の各演習課題の得点を推測する。

推測された得点は、過去の他の学習者の演習履歴と照らし合わせることによって得られる、個々の基礎的

学習内容やそれらの組合せに対するその学習者の理解状況を反映した得点である。

本手法は、出題する演習課題を決定するうえで、協調フィルタリングによる推測結果だけでなく、その学習者の現時点の学習意欲の高低を利用する。学習意欲の判定には、その学習者がこれまでに作成したプログラムに対して教員が評価した演習への取り組み姿勢を用いる。図 3 に示すように、学習意欲が低い場合は、推測された得点が最も高い、その学習者にとって理解しやすいと思われる演習課題を出題し、学習意欲が失われることを防ぐ。学習意欲が高い場合は、解けたときに大きな達成感が味わえるように、推測された得点が最も低い演習課題を出題する。これにより、すべての学習者に対して効果的な演習課題を与えることができる。

3.2 演習課題の達成度の数値化

協調フィルタリングを用いた推測を行うためには、演習課題の達成度を数値化する必要がある。そのため、演習課題ごとに「ファイルを読み込み用にオープンする」や「単方向リスト構造から i 番目の要素を削除する」といった評価観点をあらかじめ 10~20 個ずつ設定しておく。さらに、各評価観点の当該演習課題における重要度を 1~10 の 10 段階で定めておく。

教員は演習課題に設定されているそれぞれの評価観点について、「理解できている」、「理解できていない」、「どちらともいえない」のいずれかを選択することにより、学習者が作成したプログラムの評価を行う。理解できているか判断しかねる場合や、その評価観点に記されている内容を用いずにプログラムを作成している場合に「どちらともいえない」を選択するものとする。

このような方法で教員による評価が行われた、 n 個の評価観点が設定されている演習課題の得点 S は、以下の数式によって求められる。

$$S = \frac{\sum_{i=1}^n w_i e_i}{\sum_{i=1}^n w_i} \cdot 100, \quad e_i = \{0, 0.5, 1\}$$

ここで、 w_i, e_i はそれぞれ i 番目の評価観点の重要度、評価結果である。「理解できている」が選択された場合は $e_i = 1$ 、「理解できていない」が選択された場合は $e_i = 0$ 、「どちらともいえない」が選択された場合は $e_i = 0.5$ とする。ゆえに、教員による評価結果に基づいて、演習課題の達成度を表す得点が 100 を最大値として算出される。

3.3 推測結果の利用法

協調フィルタリングを用いて得られた、出題候補に

なっている各演習課題の得点の推測値から、次に出題する演習課題を決定する場合、「最も推測値の高い演習課題を出題する」と「最も推測値の低い演習課題を出題する」の2通りの方法が考えられる。それぞれの方法には、以下に示すような特徴がある。

(1) 最も推測値の高い演習課題を出題する

長所 学習者が比較的理解しやすいであろう演習課題を出題することになるので、その演習課題で行き詰まる可能性は低く、学習者のやる気を失わせない。

短所 向上心の高い学習者にとっては、出題された演習課題が簡単すぎて物足りなさを感じてしまう可能性がある。このような場合、学習効果が高いとはいえない。

(2) 最も推測値の低い演習課題を出題する

長所 学習者が難しいと思うであろう演習課題を出題することになるので、その演習課題をやりとげたとき、学習者のプログラミング能力を大きく伸ばすことができる。

短所 出題された演習課題が難しすぎて理解することができず、プログラムを完成させることが困難となり、学習者がやる気を失ってしまう可能性がある。

これら2つの方法はいずれも一長一短があり、一概にどちらのほうが効果的であるということとはできない。そこで、2.2節で述べた学習履歴調査の結果をふまえると、複数の演習課題の中から出題する演習課題を選定するさいには、学習者の理解状況に加え、学習への取り組み姿勢も考慮したうえで、2つの決定方法のいずれかを選択するべきであるといえる。

3.4 調査結果に基づく演習課題出題手法

調査結果より導かれた手法を実現するために、教員がプログラムを評価するさいに、演習課題ごとに設けられた各評価観点の評価に加え、演習に意欲的に取り組んでいるかどうかの評価も行う。学習への取り組み姿勢は、演習課題ごとに「意欲的である」、「意欲的でない」、「どちらともいえない」のいずれかを選択することにより評価する。また、学習意欲を判断するために、学習者には各演習課題に対する自己評価を

- (1) 完全に理解できた
- (2) 大半は理解できた
- (3) どちらともいえない
- (4) あまり理解できなかった
- (5) まったく理解できなかった

の中から選択してもらおう。自己評価として選択肢(4)や(5)を選択しているが、頑張っ

てプログラムを完成させようとしている場合や、選択肢(1)や(2)を選択して、課題文で指定されていない機能を実現している場合などに教員は「意欲的である」を選択する。一方、選択肢(4)や(5)を選択しており、ほぼ白紙同然か、講義資料などに書かれている例題の一部をそのまま写しただけのプログラムを提出している場合は「意欲的でない」が選択される。それ以外の場合は「どちらともいえない」が選択されるものとする。

次に、教員による学習への取り組み姿勢の評価に基づき、学習者ごとの学習意欲の数値化を行う。 n_{req} をこれまでに最低限提出するべき課題数とする、これまでに n 題の演習課題に対してプログラムを提出している学習者の現時点の学習意欲 A は、以下の数式によって求められる。

$$A = \frac{\sum_{i=1}^n a_i}{n_{req}}, \quad a_i = \{1, 3, 5\}$$

ここで、 a_i は i 番目の演習課題における学習意欲であり、「意欲的である」が選択された場合は $a_i = 5$ 、「意欲的でない」が選択された場合は $a_i = 1$ 、「どちらともいえない」が選択された場合は $a_i = 3$ とする。算出された学習意欲に対して閾値を設定しておき、出題時の学習意欲が閾値よりも低ければ、出題候補となっている演習課題のうち、協調フィルタリングによる得点の推測値が最も高い演習課題を出題する。閾値以上であれば、推測値が最も低い演習課題を出題する。これにより、学習者ごとに最適な演習課題を出題することが可能となり、学習者ひとりひとりの理解状況に合わせた効果的なプログラミング演習が実施できる。

4. 個人に合わせた演習支援システムの実現

4.1 演習支援システムの構成

我々の提案する手法を用いて学習者ごとに異なる演習課題を出題するプログラミング演習支援システムを構築した。本システムは大学などでのC言語プログラミング演習科目における利用を想定している。

本システムの構成図を図5に示す。本システムはWebブラウザをインターフェースとするWebアプリケーションである。サーバサイドはJavaを開発言語とし、サーバレットを用いて実装しており、単一のサーバ上ですべて実行される。動作環境として、OSはMicrosoft Windows 2000 Server、WebサーバにはApache Tomcat 4.1を、DBMSにはOracle Database 9iを採用した。また、サーバにはデータウェアハウスシステムDIAPRISMリコメンデーションシステム⁵⁾(以後、リコメンデーションシステムと略す)

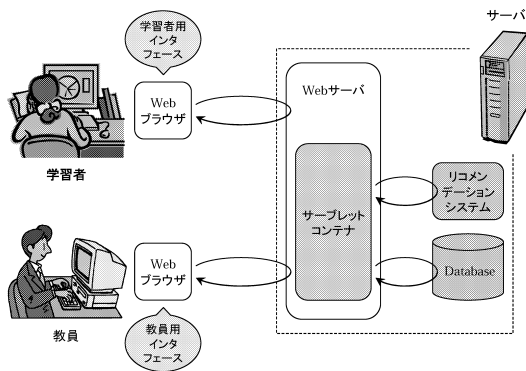


図 5 演習支援システムの構成

Fig. 5 Configuration of the exercise support system.

がインストールされており、これを利用して協調フィルタリングによる推測を実行する。

4.2 インタフェースと機能

本システムでは、学習者および教員に対してそれぞれ専用のインタフェースが用意されており、個別に定められたユーザ ID とパスワードによってそれらを利用することができる。

学習者用インタフェースでは、演習課題が出題されるだけでなく、作成したプログラムを電子ファイルのまま提出することが可能である。提出時にはその演習課題に対する自己評価を、3.4 節で述べたように選択式で回答するとともに、コメントを自由記述することができる。これらの内容は提出したプログラムとともに教員側に提示される。

教員用インタフェースでは、プログラムの提出状況、学習者がプログラム提出時に回答した自己評価の集計を閲覧することが可能である。さらに、学習者が提出したプログラムの評価を行うことができる。評価のさいには、提出されたプログラムに加え、学習者の自己評価とコメントがあわせて表示されるので、それらを参考にして、演習課題ごとに定められた各評価観点の評価、および学習への取り組み姿勢の評価を行う。教員も評価時にコメントを自由記述ことができ、その内容は学習者側に提示される。

本システム上で行われた、プログラムの提出や評価といった作業の内容はすべてデータベースに記録される。それらのデータを学習者ごとに整理し、学習履歴として提示することにより、学習の過程を可視化した。学習者は自身の、教員は全学習者の学習履歴を参照することができる。ゆえに、教員は各学習者の理解状況を的確に把握できる。学習過程の可視化機能は、演習の進捗が他より著しく滞っている学習者を見つけられるなど、適切な指導のために役立つ。

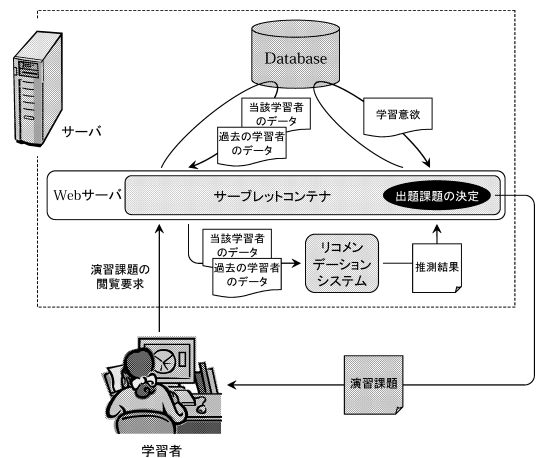


図 6 処理の流れ

Fig. 6 Work flow.

4.3 演習課題の出題

それぞれの学習者に個別の演習課題を出題するための処理の流れを図 6 に示す。まず、協調フィルタリングによる推測を実行するために、推測の対象となる学習者と、本システムを用いて学習した過去のすべての学習者についてのデータを、入力データとしてリコメンデーションシステムに渡す。過去の学習者のデータは、現在学習を進めている学習者のデータと共通のデータベースで一元管理されている。リコメンデーションシステムはこれらの入力データに基づいて、当該学習者の次の出題候補となっている各演習課題の得点を推測して出力する。出力された推測値の一覧と、当該学習者の現在の学習意欲から出題する演習課題を決定し、学習者に提示する。この処理は、学習者が各回の演習課題を初めて閲覧しようとするときに実行される。

5. 評価

5.1 演習科目への実適用

2004 年 9 月から 2005 年 1 月にかけて立命館大学情報理工学部情報システム学科で開講された 1 回生配当科目「プログラミング演習 2」において、我々が構築したプログラミング演習支援システムを実際に導入し、本手法の有効性を検証した。

この科目では、C 言語の基本的な学習内容を一通り学習した学生を対象に、データ構造とアルゴリズムといった問題解決のための手法を習得することを目標として、全 14 週のプログラミング演習を行う。当科目は 36 名程度ずつの 4 クラス、計 146 名が受講し、全受講者が本システムを利用して演習を行った。同学科では、当科目で演習するプログラミング技法の知識を講義形式で説明する科目である「データ構造とアルゴ

表 1 推薦課題の選択結果別の学習継続率
Table 1 Learning continuation rate.

		週	3	4	5	6	7	8	9	11	12	13	14
該当者数 (名)	(1) 推薦課題を選択		70	49	31	33	33	21	19	39	14	27	12
	(2) 他の課題を選択		48	22	29	22	0	25	27	—	—	0	10
	(3) いずれも未提出		28	75	86	91	113	100	100	107	132	119	124
学習継続率 (%)	(1) 推薦課題を選択		65.7	87.8	90.3	66.7	93.8	100.0	73.7	33.3	92.9	70.4	58.3
	(2) 他の課題を選択		52.1	72.7	75.9	45.5	—	84.0	74.1	—	—	—	60.0
	(3) いずれも未提出		0.0	1.3	5.8	0.0	13.3	4.0	5.0	0.9	10.6	2.5	1.6

リズム」が同時期に当科目と連動して開講された。当科目では受講者に、「データ構造とアルゴリズム」において毎週提示される1~3題の例題のプログラム作成に加え、当科目で独自に用意されている各週1~6題の演習課題の中から各週1題以上を選択してプログラムを作成するように求めた。今回は、本手法を用いて決定された演習課題は最も取り組んでほしい推薦課題と位置づけ、その演習課題だけでなく、すべての演習課題を受講者に提示し、推薦課題以外の演習課題も選択可とした。

3.4節で述べた方法によって学習者の学習意欲を数値化すると、学習意欲が特に高くも低くもない学習者はその値が3.0となる。本手法では、学習意欲が高いと判定された学習者には、難しく、やりがいがある演習課題を選出するため、そのような学習者を学習意欲が高いと判定すると、逆効果になってしまう可能性がある。そこで今回は、提出されたプログラムに学習意欲の高さが顕著に表れている学習者のみを学習意欲が高いと判定するために、学習意欲の高低を決める閾値を、中間値である3.0よりもやや高めめの3.8とした。

5.2 学習意欲の継続

表1は、前回の演習における推薦課題の選択結果によって、週ごとに受講者を以下の3つのグループに分け、各グループに該当する受講者数と、その中で当該週の演習課題を1題以上解いて提出した割合をグループ別に示している。本論文ではこの割合を学習継続率とよぶ。

- (1) 推薦課題を選択した受講者
- (2) 推薦課題以外の演習課題を選択した受講者
- (3) いずれの演習課題も未提出の受講者

第1週の演習では、受講者に関するデータがまだ存在しないので、推薦課題の提示を行っていない。そのため、第2週は対象外としている。第10週の演習課題は全受講者に対して提出を求めたレポート課題となっているため、第10週も対象外とした。また、第9週および第11週の演習では演習課題がそれぞれ1題ずつしかなかったため、第11週と第12週についてはグループ(2)の該当者数が示されていない。

当科目では、上述のとおり、当科目独自の演習課題のほかに、「データ構造とアルゴリズム」において提示される例題のプログラムの作成も受講者に課しており、各週のすべての例題とレポート課題を解いて提出することを単位認定の最低条件とした。そのため、グループ(3)に該当する受講者が週を追うごとに増加している。そのような受講者に対し、各週の演習課題を1題以上解いて提出した受講者は真に学習意欲が高く維持されていたといえる。

表から分かるように、第9週と第14週を除いては、前週の演習で推薦課題を選択したグループ(1)のほうが、他の演習課題を選択したグループ(2)に比べて学習継続率が高くなっている。第9週と第14週ではグループ(2)の値のほうが高いが、ほぼ同数であり、グループ(1)のほうが大きく下回ることはなかった。すべての週について平均すると、推薦課題を選択して解いた場合のほうが、学習継続率が11.3ポイント向上している。したがって、本手法を用いて学習者ごとに選定された演習課題を受講者が解くことによって、学習意欲を向上させ、学習を継続させることができたといえる。

前週の演習でいずれの演習課題も提出しなかった場合の学習継続率は、すべての週において著しく低い。これより、一度理解に行き詰まって学習意欲を失ってしまうと、途中で挫折するおそれが非常に高いことが分かる。よって、学習者の学習意欲を低下させないように演習を進めることが重要であると再確認された。

5.3 演習課題の達成度

図7のグラフは、受講者ごとにすべての演習を通じた得点の偏差値を求め、全受講者を推薦課題の選択回数によってグループ分けし、各グループに該当する受講者の偏差値の平均を示している。推薦課題の提示は、上述のとおり、全14週の演習のうち、第1週と第10週を除く12回の演習で行った。各受講者の偏差値を求めるために、まず、受講者が解いたすべての演習課題の得点の偏差値を個別に求めた。そして、週ごとに偏差値の平均値を算出し、さらにすべての週の偏差値を平均したものをその受講者の偏差値とした。こ

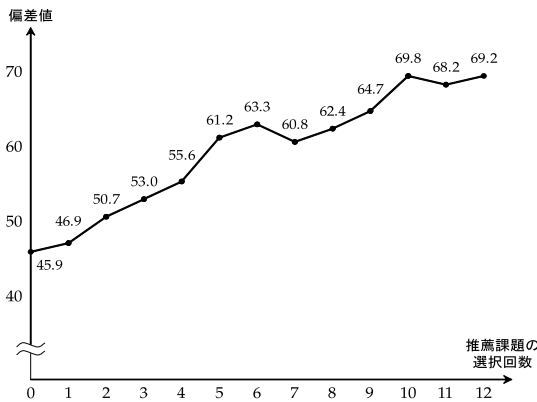


図 7 推薦課題の選択回数別の偏差値

Fig. 7 Deviation value of exercises.

表 2 推薦課題の選択回数別の推薦分布

Table 2 Distribution of recommended exercises.

選択回数	最も推測得点の低い	
	最も推測得点の低い 課題が推薦された回数	最も推測得点の高い 課題が推薦された回数
12	12.0 回	0.0 回
11	12.0 回	0.0 回
10	12.0 回	0.0 回
9	12.0 回	0.0 回
8	10.8 回	1.2 回
7	7.7 回	4.3 回
6	7.1 回	4.9 回
5	12.0 回	0.0 回
4	7.6 回	4.4 回
3	7.7 回	4.3 回
2	5.3 回	6.7 回
1	3.0 回	9.0 回
0	2.4 回	9.6 回

のとき、いずれの演習課題も提出されていない週については、その週の学習内容が理解できなかったものと見なし、その週の演習課題の得点をすべて 0 とし偏差値を算出した。また、表 2 は、推測された得点が最も低い演習課題と最も高い演習課題のそれぞれが推薦された回数の平均値をグループ別に示している。

グラフと表から分かるように、推薦課題を選択した回数が多いほど、推測得点の最も低い演習課題が推薦された回数も多く、偏差値も高くなっている。特に、12 回中 10 回以上選択した場合は、すべての週において推測得点が最も低い演習課題が推薦されたにもかかわらず、それらの演習課題に取り組むことによって著しく高い偏差値を示した。したがって、本手法を用いて選出された演習課題を受講者が解くことにより、他の演習課題を解いた場合よりも受講者の理解を深め、演習課題の達成度を高めることができたといえる。

このように、本手法を実際の C 言語プログラミング演習科目に適用した結果、本手法は豊富に用意され

た演習課題の中から、学習者の学習意欲を高めるとともに、学習内容の理解を向上させるために最も適した演習課題を選出できることが分かった。したがって、本手法は個々の学習者に合わせたプログラミング教育を実現するうえで有効であるといえる。

6. 関連研究

本論文で提案する手法の目的は、学習者の理解状況と学習意欲に基づいて、各学習者の学習意欲を高く維持しつつ、理解を深めさせるために最も効果的な演習課題を個別に選定して出題することである。

これまでに、学習者ごとに異なるプログラミング演習課題を出題する手法⁶⁾や、電子教材を用いて理解状況に応じた練習問題を自動生成するシステム⁷⁾が提案されている。前者では、あらかじめ各演習課題に難易度を設定しておき、達成状態に応じて適当な難易度の演習課題を出題する。しかし、同じ演習課題であっても理解のしやすさは学習者によって異なるので、学習者の特徴を見極めずに出題課題を決定するのは問題がある。後者では、1 つの出題内容に対して出題形式を変えることによって問題が変更されるので、問題の内容は大きく変わらない。

文献 8) は理解に行き詰まっている学習者を自動的に検知する手法を提案している。しかし、行き詰まりを検知した後の処置は教員に委ねられており、それを支援する形式的手法やツールは提供されていない。

そのほかにも、e-Learning 環境において個人に合わせた教材などを選定するための研究は数多く行われている^{9)~11)}。しかし、これらはプログラミング教育以外の分野を対象としており、教材などの良し悪しを、それを利用した学習者に多くとも 5 段階で評価してもらい、他の学習者への選定に利用している。プログラミング教育では、過去の学習者の主観ではなく、学習者の理解の程度に応じた演習課題の選定を行う必要がある。本手法では、学習者の現時点での理解状況を反映した各演習課題の達成度を 0~100 の細かい粒度で推測し、個々の学習者により高い精度で適合する演習課題を選出している。

7. おわりに

本論文では、プログラミング教育において、学習内容ごとに豊富に用意された演習課題の中から、各々の学習者にとって最も学習効果が高まるとされる演習課題を個別に選出して出題する手法を提案した。

出題課題を選出するにあたり、協調フィルタリングを用いて、過去の他の学習者の演習履歴に基づき、個々

の基礎的学習内容やそれらの組合せに対する学習者の理解状況を反映した各演習課題の得点を推測する。さらに、その学習者がこれまでに作成したプログラムに対して教員が評価した演習への取り組み姿勢から、学習意欲の高低を判定する。そして、これらの結果から、学習意欲が低下しつつある場合は、得点が高いと推測される演習課題を出題して、学習意欲を失わせることなく学習を継続させる。一方、学習意欲が高い場合は、プログラミング能力をより向上させるために、難しく、やりがいがある演習課題を出題する。これにより、教員に多大な労力を要求することなく、学習者ひとりひとりに合わせた効果的なプログラミング教育が実現できる。

本手法を大学で実際に開講されている C 言語プログラミング演習科目に適用した結果、選出された演習課題を解くことによって、それ以外の演習課題を解いた場合よりも学習継続率が 11.3 ポイント向上した。さらに、選出された演習課題を解いた回数が多い学習者ほど、演習全体を通して達成度が高かった。

今後は、学習者が提出したプログラムの評価方法について改良を行い、本手法の有効性を高める予定である。

参考文献

- 1) Kernighan, B.W. and Ritchie, D.M.: *The C Programming Language*, 2nd edition, Prentice Hall (1988). 石田晴久 (訳): プログラミング言語 C 第 2 版, 共立出版 (1989).
- 2) Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proc. 1994 ACM Conference on Computer Supported Cooperative Work*, pp.175-186 (1994).
- 3) Linden, G., Smith, B. and York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering, *IEEE Internet Computing*, Vol.7, No.1, pp.76-80 (2003).
- 4) Strang, G.: *Linear Algebra and its Applications*, Academic Press (1976). 山口昌哉 (監訳), 井上 昭 (訳): 線形代数とその応用, 産業図書 (1978).
- 5) 三菱電機インフォメーションテクノロジー株式会社: DIAPRISM リコメンデーションシステム. <http://www2.mdit.co.jp/service/diaprism/dia/tjirei6-2.pdf>
- 6) 竹田尚彦, 川口清志, 浅見美紀, 佐合尚子: プログラミングの個別指導のための演習問題サーバ, 情報処理学会研究報告コンピュータと教育, No.2000-CE-58, pp.21-28 (2000).
- 7) 菅沼 明, 峯 恒憲, 正代隆義: 学生の理解度と問題の難易度を動的に評価する練習問題自動生成システム, 情報処理学会論文誌, Vol.46, No.7, pp.1810-1818 (2005).
- 8) 中村喜宏, 赤松則男, 桑原恒夫, 玉城幹介: 操作時間間隔の変動に着目した CAI 学習の行き詰まり検知方法, 電子情報通信学会論文誌, Vol.J85-D-I, No.1, pp.79-90 (2002).
- 9) Tang, T.Y. and Mccalla, G.: Smart Recommendation for an Evolving E-Learning System, *Proc. 11th International Conference on Artificial Intelligence in Education*, pp.699-710 (2003).
- 10) O'Riordan, C., Griffith, J. and Scully, M.: Support for e-learning with Information Retrieval Techniques, *Proc. 4th Annual Conference of the LTSN Centre for Information and Computer Sciences*, pp.180-184 (2003).
- 11) Zaiane, O.R.: Building a Recommender Agent for e-Learning Systems, *Proc. International Conference on Computers in Education*, pp.55-59 (2002).

(平成 17 年 12 月 26 日受付)

(平成 18 年 11 月 2 日採録)



田口 浩 (学生会員)

昭和 57 年生。平成 16 年立命館大学理工学部情報学科卒業。平成 18 年同大学大学院理工学研究科博士前期課程修了。現在、同大学院理工学研究科博士後期課程在学中。プログラミング教育支援の研究に従事。



糸賀 裕弥 (正会員)

昭和 47 年生。平成 7 年筑波大学第 3 学群工学システム学類中途退学。平成 14 年同大学大学院博士課程工学研究科修了。博士 (工学)。平成 13 年茨城県工業技術センター流動研究員。平成 15 年立命館大学理工学部情報学科助手。平成 16 年より立命館大学情報理工学部情報システム学科講師。要求工学・情報セキュリティ技術に関する研究に従事。ソフトウェア工学, システムソフトウェア, コンピュータヒューマンインタフェースの各分野における情報セキュリティ技術にも興味を持っている。筑波大学先端学際領域研究センター (TARA) マルチメディア情報アспект「ユビキタス環境における情報提示・操作技術の研究」客員研究員。ACM, 電子情報通信学会, 日本ソフトウェア科学会各会員。



毛利 公一（正会員）

昭和 47 年生。平成 6 年立命館大学理工学部情報工学科卒業。平成 8 年同大学大学院理工学研究科修士課程情報システム学専攻修了。平成 11 年同大学院理工学研究科博士後期課程総合理工学専攻修了。同年東京農工大学工学部情報コミュニケーション工学科助手。平成 14 年立命館大学理工学部情報学科講師。平成 16 年同大学情報理工学部情報システム学科講師となり、現在に至る。この間、平成 17～18 年イリノイ大学アーバナ・シャンペーン校客員研究員。工学博士。オペレーティングシステム、プライバシー保護技術、計算機ネットワーク等の研究に従事。電子情報通信学会、日本ソフトウェア科学会、ACM、IEEE-CS 各会員。



山本 哲男（正会員）

昭和 50 年生。平成 9 年大阪大学基礎工学部情報工学科卒業。平成 14 年同大学大学院基礎工学研究科博士後期課程修了。博士（工学）。同年科学技術振興事業団研究員。平成 16 年より立命館大学情報理工学部情報システム学科講師。ソフトウェア検索およびソフトウェア再利用の研究に従事。



島川 博光（正会員）

昭和 36 年生。昭和 59 年京都大学工学部情報工学科卒業。昭和 61 年同大学大学院博士前期課程修了。同年三菱電機（株）入社。平成 11 年京都大学大学院博士後期課程修了。平成 14 年立命館大学理工学部情報学科教授。平成 16 年より立命館大学情報理工学部情報システム学科教授。博士（工学）。データベース、プログラミング教育、ユビキタス環境の研究に従事。ACM、IEEE 各会員。