

1. はじめに
2. 関連研究
3. 提案手法
4. 評価方法の概要
5. 評価
6. まとめ

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

情報処理学会

堀 由隆

May 31, 2024

1.1 本研究の背景

2/11

課題

汎用性の高い C 言語は、多くの教育機関で教えられているが、ポインタや構造体など特有の概念が存在するため、初級者が学習を進める上で大きな障壁となっている。大学などでのプログラミング演習科目は、ほとんどの場合生徒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ない。

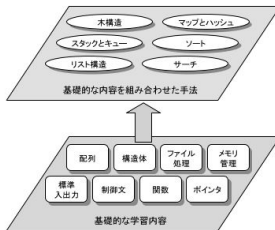


図 1 C 言語プログラミングの学習内容

Fig. 1 Contents of C programming exercise courses.

Figure 1: c 言語の分野

1. はじめに
2. 関連研究
3. 提案手法
4. 評価方法の概要
4. 評価方法の概要
5. 評価
5. 評価
6. まとめ

1.2 本研究の目的

3/11

目的

各学習者にとって最適な演習課題を出題をする. 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, 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

提案手法

協調フィルタリングを使ったシステム

- 1. はじめに
- 2. 関連研究
- 3. 提案手法
- 4. 評価方法の概要
- 4. 評価方法の概要
- 5. 評価
- 5. 評価
- 6. まとめ

1. はじめに
2. 関連研究
3. 提案手法
4. 評価方法の概要
4. 評価方法の概要
5. 評価
5. 評価
6. まとめ

協調フィルタリング



ユーザの傾向や嗜好を過去の行動という形で記録し、そのユーザー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

3.1 提案手法の概要

5/11

提案手法

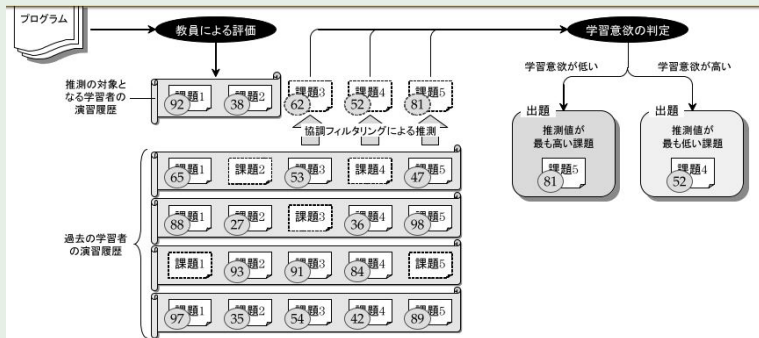


図 3 協調フィルタリングを用いた演習課題の出題

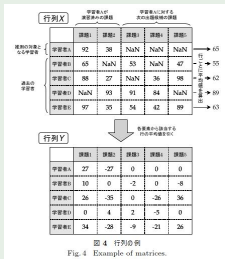
Fig.3 Selection of the most appropriate exercise with the collaborative filtering algorithm.

1. 協調フィルタリングによる推測
2. 教員による評価
3. 学習意欲の判定

3.2 提案手法の概要

6/11

1. 協調フィルタリングによる推測



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

a_i …行列 X の行の平均値

- 1. はじめに
- 2. 関連研究
- 3. 提案手法
- 4. 評価方法の概要
- 4. 評価方法の概要
- 5. 評価
- 5. 評価
- 6. まとめ

2. 教員による評価

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

課題における評価観点 n をあらかじめ 10~20 こづつ設定し、重要度 w を 1~10 の 10 段階で定める。そして教員は評価観点について、 e_i の値を「理解できている」、「理解できていない」、「どちらともいえない」から選択する。すると演習課題の得点 S が 100 を最大値として算出される

3.4 提案手法の概要

8/11

3. 学習意欲の判定

学習者に課題に対して以下から自己評価をしてもらう

- (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

そして以下の式から学習意欲Aを求める

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

(1),(2) 意欲的である $[a_i] \cdots 5$

(3) どちらでもない $[a_i] \cdots 3$

(4),(5) 意欲的でない $[a_i] \cdots 1$

$n \cdots$ 課題数

$n_{req} \cdots$ 提出するべき課題数

1. はじめに
2. 関連研究
3. 提案手法
4. 評価方法の概要
4. 評価方法の概要
5. 評価
5. 評価
6. まとめ

被験者

立命館大学生, 4 クラス, 計 146 名が 14 週に渡り本システムを導入した「プログラミング演習 2」を受講した

結果

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

9 週と 14 週を除いては, (1) のグループの方が (2) のグループよりも学習継続率が高くなっている. 全ての週について平均すると, 推薦課題を選択して解いたほうが, 学習継続率が 11.3 ポイント向上している

1. はじめに
2. 関連研究
3. 提案手法
4. 評価方法の概要
4. 評価方法の概要
5. 評価
5. 評価
6. まとめ

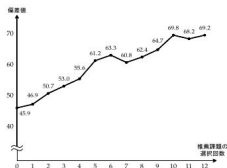


図 7 推薦課題の選択回数別の偏差値

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 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 問	0.0 問
11	12.0 問	0.0 問
10	12.0 問	0.0 問
9	12.0 問	0.0 問
8	10.8 問	1.2 問
7	7.7 問	4.3 問
6	7.1 問	4.9 問
5	12.0 問	0.0 問
4	7.6 問	4.4 問
3	7.7 問	4.3 問
2	5.3 問	6.7 問
1	3.0 問	9.0 問
0	2.4 問	9.6 問

結果

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が多く、偏差値も高くなっている

まとめ

推測得点が低い演習課題（難しい課題）が推薦された場合、それらの演習課題に取り組むことによって高い偏差値が示された。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。

1. はじめに
2. 関連研究
3. 提案手法
4. 評価方法の概要
5. 評価
6. まとめ

Temporary page!

\LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because \LaTeX now knows how many pages to expect for this document.