

# プログラミング演習支援システム CAPES の 演習履歴を用いた解答時間と難易度の推定

## Estimating Answer-Times and Difficulties of Exercises with Exercise History in a Computer-Aided Programming Exercise System

飯田 真也<sup>\*1</sup>, 立岩 佑一郎<sup>\*2</sup>, 山本 大介<sup>\*2</sup>, 高橋 直久<sup>\*2</sup>  
Shinya IIDA<sup>\*1</sup>, Yuichiro TATEIWA<sup>\*2</sup>, Daisuke YAMAMOTO<sup>\*2</sup>, Naohisa TAKAHASHI<sup>\*2</sup>

<sup>\*1</sup> 名古屋工業大学情報工学科

<sup>\*1</sup>Department of Information Engineering, Nagoya Institute of Technology

<sup>\*2</sup> 名古屋工業大学大学院 工学研究科

<sup>\*2</sup> Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology  
Email: {iida@moss.elcom., tateiwa@, yamamoto.daisuke@, naohisa@}nitech.ac.jp

あらまし：本稿では、本大学で運用しているプログラミング演習支援システム CAPES を用いて、指導者がプログラミング演習を作成する際、よりよい演習を作成するために、過去問を参考し、利用できるようにする要素として、「解答時間」や「難易度」を、CAPES の演習問題の過去の受講者のデータより推定するシステムを実現する。

キーワード： e-Learning プログラミング演習, 演習履歴, 指導者補助, クラスタ分析

### 1. はじめに

我々はプログラミング演習支援システム CAPES<sup>(1)</sup>を提案し、学部の演習に適用している。CAPES は受講者が提出した答案プログラムを自動評価する WEB システムである。この時の答案の提出時刻や、正誤判定結果などの演習履歴のデータを詳細に収集し、指導者に提供する。指導者はこの CAPES に演習を作成、登録して利用する。しかし、作成した演習を受講者が取り組むとき、想定した時間内に終了するような演習を構成するのは困難である。そこでこの問題を解決するために以下のアプローチをとる。**アプローチ** CAPES には多数の過去問が存在する。これらの必要時間や難易度等が分かれば、その過去問を参考に類題を作成、または再利用することで、より適切な演習を提供できる。しかし多数の問題を手作業で調べていくのは困難である。そこで CAPES の過去問の受講履歴から、受講者の正解までにかかる時間「解答時間」、またそれを用いた「難易度」を推定するシステムを提案する。

### 2. 提案システムの実現法

実現にあたり、個人ごとの解答時間を演習履歴から取得するのは困難であるという問題がある。この理由として、CAPES を用いた演習では、授業時間として演習の時間をとる。演習で行う課題の締切自体は 1 週間後等に設定可能である。よって受講者は課題に取り組む際に中断して、間が空く可能性がある。また、取り組む仮定で、複数の課題を並行して取り組む可能性がある。そこで以下の方法により、解答時間を推定する。

#### 2.1 解答時間推定法

CAPES では受講者ごとにログイン等の特定のイベントの発生した時刻等を記録している。今回使用するイベントについて、表 1 で示す。

表 1 CAPES イベント

問題要求	受講者が CAPES にて問題を閲覧する際に観測される。課題情報も保持しており、課題ごとに取得可能。
ログイン	受講者が CAPES にログインした際に記録される。課題情報は持っていない。
正誤判定	受講者が CAPES に答案を提出し、正誤評価を行う際に観測される。課題情報、答案の正誤情報を含み、課題ごとの取得や、正解、不正解別に取得も可能。

これらのイベントを用いて、受講者の問題の取り組み方を想定し、受講者の「解答モデル」を作成する。それに基づき解答時間を推定する。

#### 2.2 解答モデル

##### モデル 1：全時間モデル

初回の問題要求から正解の正誤判定まで、すべての時間で取り組んだと見なすモデル。

##### モデル 2：非取組時間除外モデル

中断などで間が空く問題の解決の為、全時間モデルの取組時間のうち、連続したログイン間の時間と、正誤判定の後のログインまでの時間を、取り組んでいないとみなし、除外するモデル。

全時間モデルでは初回の問題要求のイベントより、正誤判定イベントのうち、正解の正誤判定との間の時間すべてを解答時間としてとる。非取組時間除外モデルでは全時間モデルの解答時間のうち、ログインイベントから正誤判定イベントまでの、ログインから提出するまでは取り組んでいると見なし、提出から次のログイン間、つまり正誤判定イベントからログインイベントまでの時間を除外した部分の時間をとる。

各種モデルが推定する解答時間の例を、ある受講者のイベント例と共に図 1 に示す。

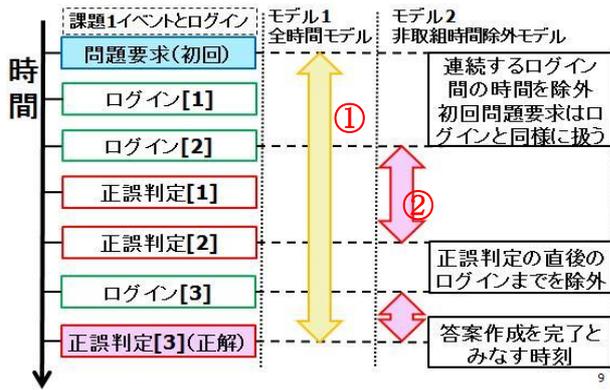


図1 イベントと答案作成モデルの例

図1ではイベント一覧と各モデルの取り組みとみなす時間の例を表している。イベントとしては初回の問題要求とログイン、正誤判定からなっている。イベントの観測する範囲として、初回問題要求から正解の正誤判定までの範囲を使用しており、この中で全時間モデルと非取組時間除外モデルを当てはめる。図1の①では全時間モデルの範囲を表しており、初回の問題要求から正解の正誤判定までの範囲をとっている。非取組時間除外モデルでは、初回の問題要求もログインイベントと同等に扱い、ログイン[2]までの時間を除外、また正誤判定[2]とログイン[3]の間も除外し、残りの②の部分の時間を取り組んだ時間とみなしている。

### 2.3 難易度推定法

推定した解答時間を使用し、問題の難易度を推定する。その手法として各種問題に解答時間の受講者平均を値として持たせ、その値を用いて、クラスター分析により問題を分類する。その分類結果からグループ単位でソーティングし、問題がどのグループに属するかによって難易度を推定するという手法をとる。

## 3. プロトタイプシステムの結果と考察

### 3.1 解答時間推定の結果と考察

解答時間推定の為に、2種類の解答モデルを作成し、それぞれの解答時間を推定した。

図2はある演習問題に対して解答時間を推定した結果の人数分布を示したグラフである。

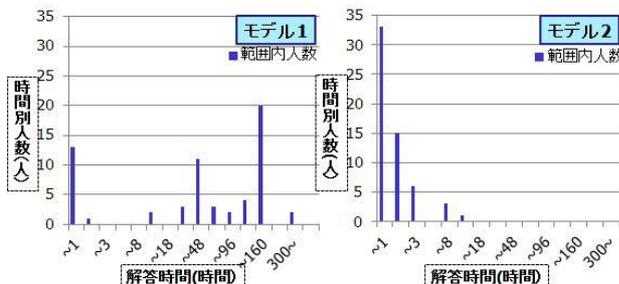


図2 モデル別解答時間分布グラフ

横軸が解答時間、縦軸がその解答時間の範囲での人数による度数となっている。この演習問題では出題からの提出締切が1週間後となっており、全時間モデルでは解答時間が160時間付近の人数が最も多くなっている。一方非取組時間除外モデルでは全体

的に解答時間が短くなっており、大半が1時間以内という結果となった。他の演習問題に対して、全時間モデルの解答時間が全体的に短い場合以外は似たような結果となった。短い場合は両モデルともに短時間となり、モデルごとの解答時間の差が小さくなった。

以上の結果より、各モデルの取れる時間の違いとして、全時間モデルでは提出時期に大きく影響を受け、非取組時間除外モデルではその提出までの中断時間を除外したことで全体的に短時間の時間が推定できると判断した。またいくつかの演習課題の解答時間の分布から、以下のことを考察した。

- 簡単な問題だとモデルごとに差が小さいが、難しい問題だと差が大きくなる。
- 難しい問題ほど受講者全体で、正解する時刻の幅が大きくなる。

### 3.2 難易度推定の結果と考察

各演習問題を、モデル別に推定した解答時間からクラスター分析により分類し、結果をTAによる難易度評価と比較した。今回は5段階の難易度とした。それをグラフ化したものを図3に示す。

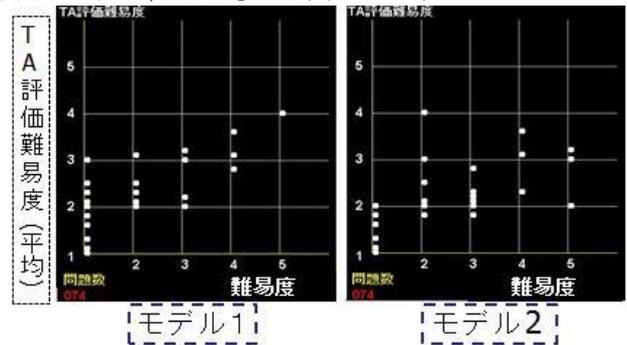


図3 難易度比較評価

横軸が解答時間によって演習問題を分類した際、5つのグループに分類し、各グループ内平均を取り、その値の小さいグループから1~5の値を割り振ったものである。値が大きいほど、解答時間の長い演習問題のグループとなる。縦軸はTAによる1~5の5段階評価の評価難易度をTA全体の平均を出したものを使用する。この結果を見ると、見た目では正の相関が多少あるように見え、特に全時間モデルのほうがより正の相関に近く見える。

## 4. おわりに

今回、提案システムとして、解答モデルによる解答時間推定、およびそれを利用した難易度推定システムとその評価機構のプロトタイプシステムを実現した。今後の予定としては、さらに別の解答モデルを作成し、より正確な解答時間の推定法を模索、また実測方法も試行し、それとの比較評価等も行っていく。また解答時間以外の要素も用いて難易度を推定するシステムを模索する。

### 参考文献

- (1) 中島秀樹：プログラミング演習支援システム CAPES の実現法に関する研究，名古屋工業大学 高橋・片山研究室 修士論文，2006