

国立大学法人電気通信大学 / The University of Electro-Communications

# ダイナミックアセスメントのための隠れマルコフ IRTモデル

著者	堤 瑛美子, 宇都 雅輝, 植野 真臣
雑誌名	電子情報通信学会論文誌. D, 情報・システム
巻	J102-D
号	2
ページ	79-92
発行年	2018-02-01
URL	<a href="http://id.nii.ac.jp/1438/00008848/">http://id.nii.ac.jp/1438/00008848/</a>

doi: 10.14923/transinfj.2018JDP7022

## ダイナミックアセスメントのための隠れマルコフ IRT モデル

堤 瑛美子<sup>†a)</sup> 宇都 雅輝<sup>†b)</sup> 植野 真臣<sup>†c)</sup>

Item Response Theory for Dynamic Assessment

Emiko TSUTSUMI<sup>†a)</sup>, Masaki UTO<sup>†b)</sup>, and Maomi UENO<sup>†c)</sup>

あらまし 教育の最も難しい問題は、教師は学習者に教えすぎても、教えなさすぎても学習者の十分な発達は望めないということである。そのために、教師は個々の学習者の理解度や最適な支援の度合いを予測することが重要な課題となっている。足場かけによる学習者のパフォーマンスを予測するために、項目反応理論を用いて最適な予測正答確率になるようにヒントを提示する足場かけシステムが開発されている。しかし、従来の項目反応理論では、学習者の能力変化がモデルに考慮されておらず、正確な正答確率を予測できないために、最適なヒント数を予測できていない可能性がある。本研究では、学習者の能力が時間変化していくプロセスを項目反応理論に組み込み、能力が隠れマルコフ過程に従って変化すると仮定した新しい項目反応モデルを提案する。提案モデルでは、能力値が継続する時間（課題数）を表すウィンドウサイズと能力の変動の程度を反映する変動パラメータをもち、これらの最適値がデータから推定されるために、学習者の真の能力変化を反映でき、予測精度を向上させることが期待される。実データを用いて、本提案の有効性を示す。

キーワード アダプティブ・ラーニング、ダイナミックアセスメント、項目反応理論、隠れマルコフモデル

### 1. ま え が き

近年、教育の現場では、教師は学習者に教えすぎても、教えなさすぎても学習者の十分な発達は望めないという問題が注目されている。Vygotsky [1], [2] は、学習者が自力で解決できない課題でも、教師の支援によって学習者の成長を促すことができる「最近接発達領域」(ZPD; Zone of Proximal Development) の考え方を導入した。最近接発達領域の考え方に従って、Bruner [3] や Wood et al. [4], Collins [5] は、学習者の発達を促すためには、学習者が高次の課題に直面した際に教師が学習者の能力に応じて適度に支援をする「足場かけ」が重要であることを示している。

足場かけでは、学習者の ZPD に関する能力を正確に測定し、教師が支援した後の学習者のパフォーマンスを予測する必要がある。すなわち、優秀な教師は、問題

解決において支援後の学習者のパフォーマンスを予測し、最小限の支援となる足場かけを与えていると考えるのである。しかし、従来行われてきた能力評価や学習者への支援は教師の経験や勘によるものであり、学習者ごとに正確な支援を行うことは非常に困難であった。そこで、Brown and Ferrara [6] や Collins [5] は、学習者の学習履歴データを用いて ZPD における学習者の能力を客観的に測定できるように、ダイナミックアセスメントと呼ばれる新たな評価手法を開発した。彼らのダイナミックアセスメントは、学習者の支援に段階的なヒントを用いることで課題を達成するまでに利用したヒント数から学習者を評価した。彼らが行った実験では、少数のヒントで課題に正答した学習者ほど学習効率が良いことが示された。その後、Wood [7] は、学習者の成長を促す支援を行うには、それまでの課題を達成するまでに利用したヒント数を評価することが重要であることを示している。しかし、これらの手法には以下の問題がある。1) 難易度の異なる課題の特性が評価に反映されておらず、能力評価としての信頼性が低い。2) 課題ごとの異なるヒントの特性が評価に反映されておらず、能力評価としての信頼性が低い。

この問題を解決するために、植野・松尾 [8], Ueno and Miyazawa [9], [10] は、課題解決におけるヒント

<sup>†</sup> 電気通信大学大学院情報理工学研究所, 調布市  
Graduate school of Informatics and Engineering, The  
University of Electro-Communications, 1-5-1 Chofugaoka,  
Chofu-shi, 182-8585 Japan

a) E-mail: tsutsumi@ai.lab.uec.ac.jp

b) E-mail: uto@ai.lab.uec.ac.jp

c) E-mail: ueno@ai.lab.uec.ac.jp

DOI:10.14923/transinfj.2018JDP7022

を与えた後の学習者の反応についての項目反応理論 (Item Response Theory, IRT と呼ぶ) を提案し, ダイナミックアセスメントの信頼性が向上したことを報告している. 更に, 彼らは提案した項目反応理論を用いて, ヒントを提示した後の学習者の課題へのパフォーマンス (正答確率) を予測し, 最も学習効果が高くなるように適応的にヒントを与えるアダプティブ・ラーニング・システムを開発している. この手法の特徴は, ヒントを与えたときの学習効果が最適になるような正答確率  $P_s$  が存在すると仮定していることである. すなわち, システムは学習者の課題への予測正答確率が設定された  $P_s$  に近づくようにヒントを選択し, 学習者に提示することにより, 学習効率が向上すると仮定している. Ueno and Miyazawa [10] では,  $P_s$  を様々に変化させてヒント提示した学習者グループのプレテストとポストテストの差異を比較した結果,  $P_s = 0.5$  に設定したグループの学習効果が最大となった. このことから, 学習効果を最大化する足場がけのためには, ヒント提示後の学習者の正答確率を高精度に予測することが重要であることが示唆される.

しかし, Ueno and Miyazawa [10] では, 長期の学習過程に対して予測正答確率の誤差が増加することが指摘されている. ここで用いられている IRT モデルは, 能力の変化の度合いが考慮されていないために, 支援後の予測正答確率が正しく推定されていない可能性が高い. このため, 学習者に提示される適応的ヒントが足りなかったり, 必要以上に提示されてしまう問題がある. 精度の高いパフォーマンスの予測を行うためには, 学習過程で学習者の真の能力値が変化することを考慮し, 推定に用いるデータのある時点以前で忘却させる必要がある. しかし, データを忘却させることで能力推定に用いられるデータ数が少なくなり, 過学習が起こって, 過大評価または過小評価されやすいというトレードオフの問題が生じてしまう. このトレードオフを解消するためには, 学習者の能力変化の度合いと, ある時点での能力値が継続する時間 (課題数) を考慮したモデルが必要である.

本研究では, 学習者の能力が学習過程において変化するプロセスを項目反応理論に組み込んだ新しいモデルを提案し, 学習者のヒント提示後のパフォーマンスの予測精度を向上させることを目的とする. 具体的には, 学習者の能力が学習過程において隠れマルコフ過程に従うと仮定した新しい項目反応モデルを提案する. このモデルでは, ある時点での能力値が影響する時間

(課題数) を表すウィンドウサイズと学習者の能力の変動の程度を反映する変動パラメータをもち, これらの最適値がデータから推定されるために, トレードオフの問題を解決し, 学習者の真の能力変化を反映できると期待される. 具体的には,

- (1) 学習者の能力が隠れマルコフ過程に従って変化する新しい項目反応モデルを提案する.
- (2) 提案モデルについて MCMC (Markov chain Monte Carlo) 法によるパラメータ推定法を提案する.
- (3) 学習者のパフォーマンスの予測精度を最適にするウィンドウサイズの推定法を提案する.

本論文では, 高次の問題解決の例として Ueno and Miyazawa [10] で扱われたプログラミング学習におけるトレース問題を扱う. トレース問題はプログラムを解説する上で必要なプロセスであり, 国家試験の情報処理技術者試験に出題されるなど, 教育現場でも多く用いられている [11]. トレース問題では, 課題を解くために必要な変数の意味を理解し, 変数の値の変化を正確に把握した上でプログラム全体の機能を理解しなければならぬ. プログラムの文法を暗記しているだけでは獲得し難いスキルを必要とし, 初学者には自力で解けない高次の問題解決能力を求めめるため, 足場がけによる学習が有効であると考えられる. 実データから本提案モデルの有効性を示す.

## 2. 項目反応理論

効果的な足場がけを行うためには, 学習者の現時点での能力値とヒントを与えた後の学習者のパフォーマンスの予測を正確に行う必要がある. このための能力評価をダイナミックアセスメントと呼び, ダイナミックアセスメントの精度の高さが効果的な足場がけを実現する. 本研究では, この精度向上のための手法開発が主な提案となる. この目標のために, 本研究では, ヒントを与えた後の学習者のパフォーマンスを予測する項目反応理論 [12], [13] を用いる. 項目反応理論はテスト理論の一つで, 近年コンピュータテストの普及に伴って, 様々な分野で使用されている実践的な数理モデルである. 項目反応理論の利点には以下が挙げられる.

- (1) 推定精度の低い異質項目の影響を最小限に抑えて能力推定を行うことができる.
- (2) 異なる項目への学習者の反応を同一尺度上で評価できる.
- (3) 過去の反応データに基づいて, 課題への正答

確率を予測できる。

ここでは、項目反応理論の中でも一般的に多く用いられる 2 母数ロジスティックモデルについて説明する。2 母数ロジスティックモデルでは、課題  $i$  に対する学習者の反応データが以下の変数  $u_i$  で表される。

$$u_i = \begin{cases} 1: & \text{学習者が課題 } i \text{ に正答} \\ 0: & \text{上記以外} \end{cases}$$

また、能力値  $\theta_j$  の学習者  $j$  が課題  $i$  に正答する確率を次式で表す。

$$p(u_i = 1 | \theta_j) = \frac{1}{1 + \exp(-1.7a_i(\theta_j - b_i))} \quad (1)$$

ここで、 $a_i$  は課題  $i$  の識別力パラメータ、 $b_i$  は課題  $i$  の難易度パラメータ、 $\theta_j$  は学習者  $j$  の能力パラメータを表す。項目パラメータ  $a_i$ 、 $b_i$  は学習データから事前に推定した値を用いる。

### 3. ダイナミックアセスメント・システム

古典的なダイナミックアセスメントは、学習者が誤答した際に段階的にヒントを提示することによってその学習過程を評価する [7]。本研究では、Ueno and Miyazawa [10] が開発した、プログラミング学習におけるトレース問題について段階的ヒントを用いたダイナミックアセスメント・システムを用いる。

システムでは最初にプログラミングの基礎知識について学習し、その後、プログラミングのコードを読み、変数の最後の値を回答する課題を数問学習する。課題では、学習者が誤答した際には、図 1 のようにヒントとしてプログラミングの文法に関する説明やコードの意味などが段階的に提示され、学習者が課題を達成す

るまでヒントをより具体的な内容にしていく。学習者が課題に正答した場合や、最後のヒントを提示しても正答しなかった場合には、課題の解説と解答がフィードバックとして与えられる。課題の解答とその解説をフィードバックすることで、正答者は解答に至るまでの思考過程を再確認し、誤答者は解説を読むことで間違えていた箇所を確認することができる。Ueno and Miyazawa [10] では、本システムを用いて予測正答確率が 0.5 に近づくようなヒントを提示して正答できた場合に能力値が最も向上すると報告している。また、ヒントなしで正解した場合、ヒントありで誤答した場合でも、前述のフィードバックにより能力値が向上することが報告されている。本論文では、本システムを用いた学習過程において学習者の能力値が単調に増加すると仮定する。

本システムで得られるデータのデータ発生モデルを提案することが本論の主な目的である。ここでは、本システムで得られるデータを整理しておく。このシステムでは、各課題  $i$  に対して  $K - 1$  個の段階的ヒント  $\{k\}$ , ( $k = 1, \dots, K - 1$ ) が用意されている。初めはヒントを表示しない状態で学習者  $j$  に課題  $i$  を提示する。学習者が課題  $i$  に誤答した場合はヒント  $k = 1$  を提示し、更に誤答するごとにヒント  $k = K - 1$  までを順次提示する。正答するか、最後のヒントが提示されても誤答した場合は、次の課題  $i + 1$  を提示する。課題数  $I$  に達するまでこの操作を繰り返す。学習者数を  $J$ 、課題数を  $I$  とすると、学習者  $j$  が課題  $i$  にヒント  $k$  を表示した段階で正答した反応データは次のように得る。

$$x_{ji} = \begin{cases} k: & \text{ヒント } k \text{ を与えられたときに正答} \\ K: & \text{全てのヒントを提示しても誤答} \\ 0: & \text{ヒントなしで正答} \end{cases}$$

$$X = \{x_{ji}\}, (j = 1, \dots, J, i = 1, \dots, I)$$

適切な足場がけを行うためには、学習者の能力を正確に把握し、各ヒントを提供した後の学習者のパフォーマンスを予測しなければならない。これまで、学習者の能力とヒントごとの学習者のパフォーマンスを予測するための項目反応モデルとして段階反応モデルが提案されてきた [10], [14]。次章でダイナミックアセスメントのための項目反応理論である段階反応モデルについて述べる。

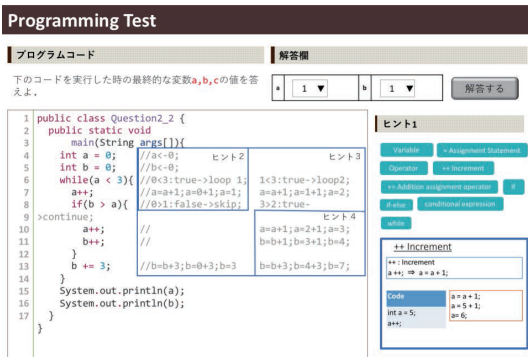


図 1 段階的ヒントの例  
Fig. 1 Example of graded hints.

#### 4. ダイナミックアセスメントのための項目反応理論

適切な足場がけを行うためには、学習者の能力を正確に把握し、各ヒントを提供した後の学習者のパフォーマンスを予測しなければならない。これまで、学習者の能力とヒントごとの学習者のパフォーマンスを予測するための項目反応モデルとして段階反応モデルが提案されてきた [10], [14]。以降は、この段階反応モデルを IRT (Item Response Theory) と略記する。

本章ではダイナミックアセスメントのための IRT モデルについて説明する。IRT モデルでは学習者  $j$  が課題  $i$  に対してヒント  $k$  ( $k = 1, \dots, K-1$ ) で正答する確率  $P_{ijk}$  を次式で与える。

$$P_{ijk} = P_{ijk-1}^* - P_{ijk}^* \quad (2)$$

$$P_{ijk}^* = \frac{1}{1 + \exp(-a_i(\theta_j - b_{ik}))} \quad (3)$$

ただし、 $P_{ij0}^* = 1$ ,  $P_{ijK}^* = 0$  である。ここで、 $a_i$  は課題  $i$  の識別力を表すパラメータ、 $b_{ik}$  は課題  $i$  でヒント  $k$  が提示されたときの難易度を表すヒントパラメータ、 $\theta_j$  は学習者  $j$  の能力値パラメータを表す。ただし、 $b_{j1} > \dots > b_{jk} > \dots > b_{jK-1}$  である。図 2 に、 $K = 5$ ,  $a_i = 1.0$ ,  $b_{i1} = 3.0$ ,  $b_{i2} = 1.0$ ,  $b_{i3} = -1.0$ ,  $b_{i4} = -3.0$  とした 4 つのヒントを有する課題に対する項目反応関数の例を示す。図 2 は、横軸は学習者の能力を示し、縦軸は  $k$  番目のヒントが提示されたときに学習者  $j$  が課題  $i$  に正答する確率  $P_{ijk}$  を示す。図 2 より、ヒントなし ( $k=0$ ) の場合には、能力の低い学習者はほとんど正答せず、能力の高い学習者の正答確率が高くなっている。また、ヒント数が増えるごとに、能力の低い学習者の正答確率が上昇していくことがわかる。

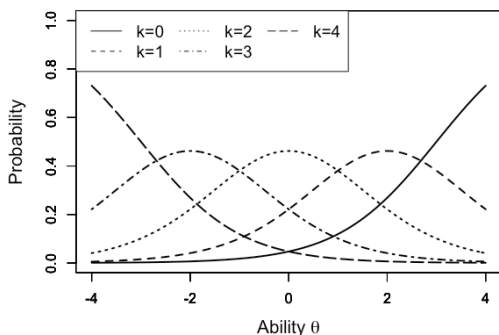


図 2 段階反応モデルの例  
Fig. 2 Graded response model for hints.

更に、Ueno and Miyazawa [10] は、IRT モデルを用いることで学習者にヒントを与えた後の課題への正答確率を予測し、適応的に支援を行う足場がけシステムを開発した。これらの研究からは、ヒントは学習者に多すぎても、少なすぎても学習効果が減少してしまい、ちょうど予測正答確率が 0.5 になるようにヒントを出す支援が最も学習効率が良いことが報告されている。以上より、ヒント提示後の学習者の正答確率を精度高く予測することが効果的な学習のために有効であることがわかる。しかし、Ueno and Miyazawa [10] のシステムで用いている IRT モデルでは、学習者の真の能力値は固定されており、能力の変化の度合いやある時点での能力値が継続する時間が考慮されていない。

信頼性の高い能力推定を行うためには、学習過程で学習者の真の能力値が変化することを考慮し、推定に用いるデータがある時点以前で忘却させる必要がある。しかし、データを忘却させることで能力推定に用いられるデータ数が少なくなり、過学習が起こり、能力値が過大評価または過小評価されやすくなるというトレードオフの問題が生じる。このトレードオフを解消するためには、学習者の能力変化の度合いと、ある時点での能力値が継続する時間を考慮したモデルが必要である。

本研究では、このトレードオフを解消するために、学習者の潜在変数である能力値の時系列変化が隠れマルコフ過程に従うと仮定し、ある時点での能力値が影響する時間 (課題数) を表すウィンドウサイズと学習者の能力の変動の程度を反映する変動パラメータを導入した新たな項目反応モデルを提案する。このモデルによってトレードオフの問題を解消し、過学習を避け、推定精度の向上が期待できる。

#### 5. 隠れマルコフ IRT モデル

本章では、ダイナミックアセスメントのための新しい IRT モデルを提案する。段階反応モデルにおいて、従来では固定されていた学習者  $j$  の能力値  $\theta_j$  を時系列で変化させ、ある時点  $t$  の能力値  $\theta_{jt}$  が一つ前の時点  $t-1$  での能力値  $\theta_{jt-1}$  に依存する隠れマルコフモデルを IRT に組み込んだモデルを提案する。通常、隠れマルコフモデルの隠れ変数は離散値で扱われるが、提案モデルでは能力値を隠れ変数とするため、連続値で扱う。

従来の IRT モデルと隠れマルコフ IRT モデルのグラフィカルモデルを図 3, 図 4 に示す。前述したとおり、従来の IRT モデルは学習過程が一つの能力値  $\theta_j$  に依存する。一方、隠れマルコフ IRT モデルは、学習



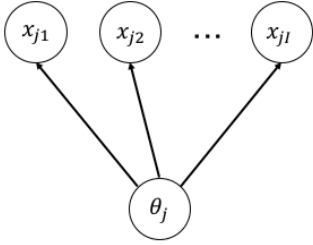


図 3 従来の IRT モデル  
Fig. 3 Conventional IRT model.

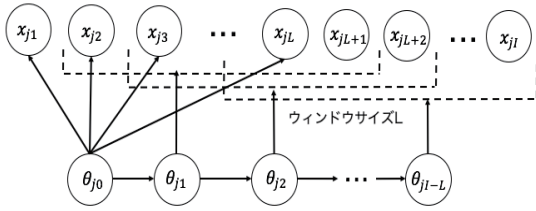


図 4 隠れマルコフ IRT モデル  
Fig. 4 Hidden Markov IRT model.

過程 (課題) が進むごとに、学習者の能力値  $\theta_{jt}$  が直前の  $\theta_{jt-1}$  に依存して確率的に変化していくモデルである。このとき、能力値  $\theta_{jt}$  の変動パラメータ  $\delta$  を設定することで、 $\theta_j$  の変動を制限する。提案モデルでは、学習過程において学習者の能力値が影響を及ぼしていると考えられる課題数 (ウィンドウサイズ) を  $L$  と設定する。

能力値  $\theta_{jt}(t = 1, \dots, I - L)$  の変動モデルには、音声認識や画像認識の分野でパラメータ推定の手法に用いられるスライディングウィンドウ方式 [15], [16] を用いる。スライディングウィンドウは、ある小領域を設定し、一定の幅でずらしながら隠れ変数が影響する顕在変数領域を決定する方法である。本モデルでは、課題  $i = L$  以降の能力値推定において、推定に用いる学習課題を 1 題ずつずらして行うことで、能力値の推移を考慮する。この方法によって、学習者が取り組んだ直近の  $L$  個の課題以前の学習データを忘却した能力推定が可能となる。

学習過程における  $\{t\}, (0, \dots, I - L)$  は、

$$\begin{cases} t = 0: & i = 1, \dots, L \text{ のとき} \\ t = 1: & i = 2, \dots, L + 1 \text{ のとき} \\ \vdots & \vdots \\ t = I - L: & i = I - L - 1, \dots, I \text{ のとき} \end{cases} \quad (4)$$

とする。提案モデルは、ウィンドウサイズ  $L$  が小さい場合は能力値  $\theta_{jt}$  が短期間の学習過程にのみ影響する (能力値推定において過去の学習過程のデータを多く忘却する) モデルとなり、 $L$  が大きい場合は一つの  $\theta_{jt}$  が長期間の学習過程に影響する (能力値推定において過去の学習過程のデータをあまり忘却しない) モデルとなる。

提案モデルでは時点  $t$  において学習者  $j$  が課題  $i$  にヒント  $k$  で正答する確率  $P_{ijtk}$  を次式で表す。

$$P_{ijtk} = P_{ijtk}^* - P_{ijtk-1}^* \quad (5)$$

$$P_{ijtk}^* = \frac{1}{(1 + \exp(-a_i(\theta_{jt} - b_{ik})))} \quad (6)$$

$a_i$  は課題  $i$  の識別力パラメータ、 $b_{ik}$  は課題  $i$  でヒント  $k$  が提示されたときの難易度を表すヒントパラメータ、 $\theta_{jt}$  は時点  $t$  での学習者  $j$  の能力パラメータを表す。ただし、

$$\theta_{jt} \sim N(\theta_{jt-1}, \delta) \quad (7)$$

$$\theta_{j0} \sim N(0, 1) \quad (8)$$

ここで、 $N(\mu, \sigma)$  は平均  $\mu$ 、標準偏差  $\sigma$  の正規分布を表す。式 (7) から  $\delta$  は時間経過による能力の変動の大きさを表すパラメータとみなせる。提案モデルの目的は、ダイナミックアセスメントにおいて、学習者の学習過程から学習者の能力値と課題へのパフォーマンスを予測することである。提案モデルのウィンドウサイズ  $L$  と能力値の変動パラメータ  $\delta$  は、能力推定に利用する学習データ数と能力推定値のトレードオフの問題を解決するための重要な役割をもつ。

提案モデルは、ウィンドウサイズ  $L$  を全課題数に一致させた場合には、能力値が過去のデータを忘却せず時系列変化しないモデルとなるため、従来の IRT モデルと同じモデルを表現することができる。

効果的な足場かけを行うためには、学習に合わせて最適なウィンドウサイズ  $L$  と変動パラメータ  $\delta$  の組み合わせを求める必要がある。この二つのパラメータを変化させることで、提案モデルは多様な学習過程に柔軟に対応させることができる。ウィンドウサイズ  $L$  と変動パラメータ  $\delta$  の関係は以下のとおりである。

(1)  $L$  と  $\delta$  が共に小さい

$\theta_{jt}$  が影響する課題数が少なく、能力の変動もほぼ起こらないため、それまでの学習過程に関係なく、 $\theta_{jt}$  がほとんど変化しないモデル。

(2)  $L$  が小さく  $\delta$  が大きい

直前の学習過程にのみ影響され、能力値の変動幅が大きいいため、 $\theta_{jt}$  の急激な変動が起こるモデル。

(3)  $L$  と  $\delta$  が共に大きい

それまでの学習過程に強く影響を受け、 $\theta_{jt}$  が大きく変動するモデル。

(4)  $L$  が大きく  $\delta$  が小さい

それまでの学習過程の影響を受けるが、 $\theta_{jt}$  の急激な変動を抑制するモデル。

次章で、本モデルのパラメータ推定法について述べる。

## 6. 項目パラメータ推定

従来の段階反応モデルのパラメータ推定には、一般に、ニュートンラフソン法や EM アルゴリズムを用いた周辺最ゆう推定 (Marginal Maximum Likelihood: MML) や最大事後確率推定 (Maximum A Posteriori: MAP) が用いられてきた。また、近年では、マルコフ連鎖モンテカルロ (MCMC) 法を用いた期待事後確率推定 (Expected A Posteriori: EAP) も一般的になりつつある。ニュートンラフソン法を用いた MAP 推定や MML 推定は、2 母数ロジスティックモデルや段階反応モデルなどの単純なモデルを用いる場合や、大量のデータが得られている場合には高速に安定したパラメータ推定が可能であるが、複雑なモデルを扱う場合には推定精度が低下する。MCMC 法は事後分布からのランダムサンプルを用いてパラメータを推定する手法であり、計算コストは高いが、本研究のようにモデルが複雑な場合やデータ数が少ない場合にも高精度なパラメータ推定を実現できる [18]。

ここで、各パラメータの集合をそれぞれ  $\theta = \{\theta_{10}, \dots, \theta_{JL-L}\}$ ,  $\mathbf{a} = \{a_1, \dots, a_I\}$ ,  $\mathbf{b} = \{b_{11}, \dots, b_{IK-1}\}$ ,  $\theta_{jt}$  と  $a_i$  の事前分布をそれぞれ  $g(\theta_{jt})$ ,  $g(a_i)$  とし、 $\mathbf{b}_i = \{b_{i1}, \dots, b_{ik-1}\}$  の事前分布を  $g(\mathbf{b}_i)$  を表す。

このとき、反応データ  $\mathbf{X}$  を所与としたパラメータの事後分布は以下のように表せる。

$$\begin{aligned} p(\theta, \mathbf{a}, \mathbf{b} | \mathbf{X}) & \propto L(\mathbf{X} | \theta, \mathbf{a}, \mathbf{b})g(\mathbf{a})g(\mathbf{b})g(\theta) \\ & = \left[ \prod_{t=0}^{I-L} \prod_{i=t+1}^{L+t+1} \prod_{k=1}^K (P_{ijtk})^{z_{ijk}} \right] \quad (9) \\ & \quad \left[ \prod_{i=1}^I (g(a_i) \cdot g(\mathbf{b}_i)) \right] \left[ \prod_{t=0}^{I-L} \prod_{j=1}^J g(\theta_{jt}) \right] \end{aligned}$$

ここで、

$$z_{ijk} = \begin{cases} 1: & x_{ji} = k \\ 0: & \text{上記以外} \end{cases}$$

MCMC の手法のうち、ブロック化ギブス・サンプリング法とメトロポリスヘイスティングス法を組み合わせた手法 [17], [18] でパラメータ推定を行う。以下に手順を示す。

(1) 初めに、各パラメータの初期値を事前分布からランダムにサンプリングする。本研究では、各パラメータの事前分布はそれぞれ次のように設定する。

$$\log a_i \sim N(0.0, 0.2)$$

$$\theta_{j0} \sim N(0.0, 1.0)$$

$$\theta_{jt} \sim N(\theta_{jt-1}, \delta)$$

$$\mathbf{b}_i \sim MN(\mu, \Sigma)$$

$$\mu = \{-2.0, -1.0, 0.0, 1.0, 2.0\}$$

$$\Sigma = \text{diag}[0.16, 0.16, \dots, 0.16]$$

(2)  $\theta_j = \{\theta_{j0}, \dots, \theta_{jL-L}\}$  を現在のパラメータ値  $\theta_j'$  に依存する提案分布  $q(\theta_j | \theta_j')$  に従ってサンプリングし、以下の採択率に基づいて採択する。

$$\begin{aligned} \alpha(\theta_j | \theta_j') & = \min \left( \frac{L(\mathbf{X}_j | \theta_j, \mathbf{a}', \mathbf{b}') \prod_{t=0}^{I-L} g(\theta_{jt})}{L(\mathbf{X}_j | \theta_j', \mathbf{a}', \mathbf{b}') \prod_{t=0}^{I-L} g(\theta_{jt}')}, 1 \right) \quad (10) \end{aligned}$$

提案分布には  $N(\theta_j', \sigma \mathbf{1}_{I-L})$  を用いる。ここで、 $\mathbf{1}_n$  は  $n \times n$  の単位行列を表す。本研究では、 $\sigma = 0.01$  と

---

### Algorithm 1 MCMC algorithm

---

**Given** maximum chain length  $S$ , burn-in  $B$ , interbal  $E$   
**Initialize** MCMC sample  $A \leftarrow \phi$   
**Initialize**  $\theta^0, \mathbf{a}^0, \mathbf{b}^0$   
1: **for**  $s = 1$  to  $S$  **do**  
2:   **for**  $j \in \{1 \dots J\}$  **do**  
3:     Sample  $\theta_j^s \sim N(\theta_j^{s-1}, \sigma \mathbf{1}_{I-L})$   
4:     Accept  $\theta_j^s$  with the probability  $\alpha(\theta_j^s | \theta_j^{s-1})$   
5:   **end for**  
6:   **for**  $i \in \{1 \dots I\}$  **do**  
7:     Sample  $a_i^s \sim N(a_i^{s-1}, \sigma \mathbf{1})$   
8:     Accept  $a_i^s$  with the probability  $\alpha(a_i^s | a_i^{s-1})$   
9:     Sample  $\mathbf{b}_i^s \sim N(\mathbf{b}_i^{s-1}, \sigma \mathbf{1}_{K-1})$   
10:     Accept  $\mathbf{b}_i^s$  with the probability  $\alpha(\mathbf{b}_i^s | \mathbf{b}_i^{s-1})$   
11:   **end for**  
12:   **if**  $s \geq B$  and  $s\%E = 0$  then **then**  
13:      $A \leftarrow (\theta^s, \mathbf{a}^s, \mathbf{b}^s)$   
14:   **end if**  
15: **end for**  
16: **return** average value of  $A$

---

表 1 各課題におけるヒント数とヒントなしでの正答率  
Table 1 Number of hints and percentage correct for each task without hints.

課題	1	2	3	4	5	6
ヒント数	8	8	8	9	10	11
正答割合 (%)	60.0	66.7	65.3	46.7	50.6	46.7
課題	7	8	9	10	11	12
ヒント数	9	8	13	12	12	13
正答割合 (%)	54.7	54.7	50.6	53.7	48.0	80.0
課題	13	14	15	16	17	18
ヒント数	7	10	11	6	9	8
正答割合 (%)	81.3	86.7	49.3	94.7	80.0	49.3

する。

(3) パラメータ  $a_i$  と  $b_i$  についても上記と同様にサンプリングを行う。

(4) 初期値の影響をなくすために、burn-in で設定した回数より前のサンプルは破棄する。また、自己相関を考慮し、得られたサンプルの thinning を行い、そのサンプル列の期待値を推定値とする。本研究では burn-in を 20,000 回として、20,000~40,000 回のうちから 1,000 回の間隔でサンプルを取得し、その平均値を EAP 推定値とした。提案モデルの MCMC アルゴリズムの擬似コードを Algorithm 1 に示す。

## 7. 評価実験

### 7.1 データ

評価実験では、プログラミング初学者の大学生 75 人を対象にプログラミング学習におけるトレース問題 18 課題について、3. で示したシステムを用いて学習データを収集した。学習者は「変数の四則演算」「条件分岐 while ループ」「for ループ」「配列」「関数・メソッド呼び出し」の文法について学習し、各領域を学習した後、対応するトレース問題に回答する。ただし、「変数の四則演算」「条件分岐 while ループ」「for ループ」は各 4 題、「配列」「関数・メソッド呼び出し」では 3 題が出題される。出題例として各領域から 1 題ずつの課題とヒントの内容を付録に示した。また、課題ごとのヒント数と各課題におけるヒントなしでの正答率を表 1 に示す。

初めのヒントは図 1 の画面右側のように、プログラミングの基礎的な用語についての説明を提示する。ヒント提示後も誤答した場合には、画面左側のように、プログラムの各行の操作について、説明を順次提示していく。課題  $i$  のヒント数が  $K$  のとき、学習者  $j$  がヒントなしで正答した場合には学習者  $j$  の学習データを  $x_{ji} = 0$  とし、ヒント  $k$  が提示された後に正答した

場合は  $x_{ji} = k$ 、最後のヒントが提示されても誤答した場合は  $x_{ji} = K + 1$  とする。

### 7.2 学習データに最適なウィンドウサイズと変動パラメータの決定

5. で述べたとおり、提案モデルは、能力値  $\theta_{jt}$  が学習に影響する課題数 (ウィンドウサイズ)  $L$  の値によって異なるモデルとなる。また、モデルごとに能力の変動パラメータ  $\delta$  の最適値も変化するため、最適な  $L$  と  $\delta$  の組み合わせを求める必要がある。提案モデルでは、トレーニングデータ数がウィンドウサイズによって変動するため、周辺ゆう度や AIC, BIC などの従来のモデル選択基準では、これらの最適値を求めることができない。そこで、本研究では、以下の手順で算出される予測精度を最大にする  $L$  と  $\delta$  の組み合わせを最適値とする。

(1)  $L$  と  $\delta$  を所与として、提案モデルの識別力パラメータとヒントパラメータを MCMC アルゴリズムで推定する。

(2) 提案モデルを用いて、学習者  $j$  が課題  $i \in \{2, \dots, 18\}$  においてヒント  $k \in \{0, 1, \dots, K\}$  で正答する確率  $P_{ijtk}$  を求め、 $P_{ijtk}$  が最大となるヒント  $k$  を予測利用ヒント数  $\hat{x}_{ji}$  とする。

$$\hat{x}_{ji} = \arg \max_{k \in \{0, 1, \dots, K\}} P_{ijtk} \quad (11)$$

ただし、 $P_{ijtk}$  の計算に利用する各学習者の能力  $\hat{\theta}_{jt}$  は、課題  $i - 1$  以前のデータ  $\mathbf{x}_j^{(i-1)} = \{x_{j1}, \dots, x_{j,i-1}\}$  を用いて以下の EAP 推定法で求める。

$$\begin{aligned} \hat{\theta}_{jt} &= E[\theta_{jt} | \mathbf{x}_j^{(i-1)}] \\ &= \frac{\int_{-\infty}^{+\infty} \theta_{jt} g(\theta_{jt}) L(\mathbf{x}_j^{(i-1)} | \theta_{jt}) d\theta_{jt}}{\int_{-\infty}^{+\infty} g(\theta_{jt}) L(\mathbf{x}_j^{(i-1)} | \theta_{jt}) d\theta_{jt}} \end{aligned} \quad (12)$$

ここで、

(a)  $i - 1 \leq L$  のとき

$$\theta_{j0} \sim N(0.0, 1.0) \quad (13)$$

$$L(\mathbf{x}_j^{(i-1)} | \theta_{jt}) = \prod_{i'=1}^{i-1} \prod_{k=1}^K (P_{i'jtk})^{z_{i'jk}} \quad (14)$$

(b)  $i - 1 > L$  のとき

$$\theta_{jt} \sim N(\theta_{jt-1}, \delta) \quad (15)$$

$$L(\mathbf{x}_j^{(i-1)} | \theta_{jt}) = \prod_{i'=t+1}^{i-1} \prod_{k=1}^K (P_{i'jtk})^{z_{i'jk}} \quad (16)$$



表 2 予測利用ヒント数の予測精度  
Table 2 Prediction accuracies of the predictive number of hints.

delta	ウィンドウサイズ $L$								
	1	2	3	4	5	6	7	8	9
0.1	52.1%	63.4%	65.6%	65.1%	64.3%	64.0%	63.4%	63.2%	63.3%
0.2	54.6%	64.2%	65.4%	64.7%	64.2%	63.9%	63.3%	62.8%	62.8%
0.3	54.7%	63.1%	65.3%	64.4%	63.5%	63.5%	63.0%	63.1%	62.8%
0.4	57.1%	63.1%	65.0%	64.3%	63.3%	63.7%	63.0%	62.8%	62.8%
0.5	55.6%	64.0%	65.0%	64.2%	63.2%	63.1%	63.0%	62.3%	62.2%
delta	10	11	12	13	14	15	16	17	18 (従来の IRT)
0.1	62.9%	62.3%	62.3%	62.0%	61.9%	61.2%	61.3%	60.8%	60.95%
0.2	62.5%	61.9%	62.3%	61.8%	61.7%	61.0%	61.1%	61.2%	60.95%
0.3	62.4%	62.1%	61.9%	61.8%	61.5%	61.0%	61.0%	60.7%	60.95%
0.4	62.2%	61.7%	61.9%	61.8%	61.1%	61.3%	61.1%	60.8%	60.95%
0.5	62.3%	61.9%	61.8%	61.8%	61.4%	61.2%	61.0%	60.6%	60.95%

表 3 課題ごとの予測利用ヒント数の予測精度  
Table 3 Prediction accuracies of the predictive number of hints for each task.

	課題 2	課題 3	課題 4	課題 5	課題 6	課題 7	課題 8	課題 9	課題 10
提案モデル ( $L = 3, \delta = 0.1$ )	52.7%	63.9%	46.9%	50.9%	68.8%	55.7%	62.7%	48.3%	57.2%
従来の IRT モデル	34.2%	60.3%	46.3%	45.4%	64.8%	52.8%	52.7%	46.7%	56.0%
DA	38.7%	30.7%	9.3%	1.3%	1.3%	8.0%	8.0%	0.0%	0.0%
	課題 11	課題 12	課題 13	課題 14	課題 15	課題 16	課題 17	課題 18	平均
提案モデル ( $L = 3, \delta = 0.1$ )	49.3%	79.7%	83.5%	87.6%	50.5%	96.0%	91.6%	69.2%	65.6%
従来の IRT モデル	40.2%	67.0%	83.0%	87.6%	49.7%	85.3%	90.7%	69.0%	60.3%
DA	0.0%	0.0%	1.3%	0.0%	0.0%	0.0%	2.7%	1.3%	5.7%

実際には、式中の積分は  $-2.5 < \theta_{jt} < 2.5$  での 100 点の区分求積法を用いて近似値を求める。

(3) 学習者  $j$  の課題  $i$  における実際のヒント利用数  $x_{ji}$  と予測利用ヒント数  $\hat{x}_{ji}$  を用いて、各課題  $i$  における一致率  $c_i$  を次式で求める。

$$c_i = \frac{1}{J} \sum_{j=1}^J \psi(\hat{x}_{ji}, x_{ji}) \quad (17)$$

ここで、 $\psi(\hat{x}_{ji}, x_{ji})$  は  $\hat{x}_{ji}$  と  $x_{ji}$  が一致するときに 1、そうでないときに 0 をとる関数とする。

(4) 手順 (3) で求めた一致率を全ての課題について平均し、提案モデルの予測精度  $c$  として次式を求める。

$$c = \frac{1}{I-1} \sum_{i=2}^I c_i \quad (18)$$

表 2 に  $L \in \{1, 2, \dots, 17\}$  と  $\delta \in \{0.1, 0.2, 0.3, 0.5\}$  における予測精度  $c$  を示す。ここで、ウィンドウサイズ  $L = 18$  (全課題数) の提案モデルは、従来の IRT モデルと一致することに注意されたい。本実験では、 $\delta > 0.5$  の結果を示していないが、 $\delta$  が 0.5 より大きい場合は、全てのケースにおいて  $\delta = 0.5$  より精度が低下することを確認している。

表 2 より、予測精度  $c$  が最大となるのは、ウィンドウサイズを  $L = 3$ 、能力値の変動幅を  $\delta = 0.1$  とした提案モデルであり、従来の IRT モデルより大幅に予測精度が改善していることがわかる。

### 7.3 課題ごとの利用ヒント数の予測精度

本節では、最適な  $L$  と  $\delta$  を用いた提案モデル ( $L = 3, \delta = 0.1$ ) と従来の IRT モデル、従来のダイナミックアセスメント手法 [4] について、前節の予測精度算出手順 (3) で得られる課題ごとの利用ヒント数の予測精度  $c_i$  を分析する。ただし、従来のダイナミックアセスメント手法では、学習者がこれまでに解いた課題で利用したヒント数の平均を次の課題での予測利用ヒント数として、手順 (3) で予測精度を求める。

実験結果を表 3 に示す。表 3 より、提案モデルは全ての課題において従来の IRT モデルより高い予測精度を示したことがわかる。また、従来のダイナミックアセスメントの手法 (DA) では、提案モデル、従来の IRT モデルに比べて著しく予測精度が低いことがわかる。

ここで、提案モデルと従来の IRT モデルのパラメータ推定値の特徴を分析するために、提案モデル ( $L = 3, \delta = 0.1$ ) と従来の IRT モデルにおけるパラメータ推定値を表 4 に示す。表 4 より、提案モデルで

表 4 課題の識別力パラメータ  $a_i$  とヒントパラメータ  $b_{ik}$   
Table 4 Estimated characteristic parameters of hints.

従来の IRT モデル															
課題 $i$	$a_i$	$b_{i0}$	$b_{i1}$	$b_{i2}$	$b_{i3}$	$b_{i4}$	$b_{i5}$	$b_{i6}$	$b_{i7}$	$b_{i8}$	$b_{i9}$	$b_{i10}$	$b_{i11}$	$b_{i12}$	$b_{i13}$
1	1.20	1.83	1.10	0.58	0.37	-0.12	-0.45	-0.95	-1.36	-2.46					
2	1.06	2.02	1.46	0.71	0.37	0.06	-0.56	-1.28	-1.51	-2.21					
3	1.18	1.68	1.02	0.45	0.31	0.14	-0.38	-1.00	-1.22	-2.24					
4	0.97	2.25	1.34	0.68	0.49	0.16	-0.22	-0.69	-1.13	-1.34	-2.10				
5	1.03	2.08	1.39	1.10	0.76	0.47	-0.07	-0.18	-0.52	-1.01	-1.51	-2.30			
6	1.31	1.69	1.06	0.83	0.56	0.34	0.04	-0.15	-0.54	-0.80	-1.00	-1.82	-2.16		
7	1.00	1.86	1.49	0.96	0.63	0.54	-0.24	-0.57	-1.25	-1.66	-2.23				
8	1.13	1.75	0.93	0.64	0.39	0.08	-0.24	-0.60	-1.60	-2.38					
9	1.16	2.16	1.40	1.09	0.90	0.66	0.37	0.28	-0.01	-0.43	-0.93	-1.11	-1.47	-1.72	-2.31
10	0.91	1.79	1.30	1.14	0.83	0.58	0.46	0.07	-0.18	-0.30	-0.74	-1.27	-1.79	-2.29	
11	0.84	2.14	1.65	1.06	0.81	0.52	0.30	0.19	-0.03	-0.24	-1.02	-1.40	-1.71	-2.22	
12	1.40	1.42	0.92	0.84	0.74	0.60	0.31	0.12	-0.27	-0.47	-0.69	-0.88	-1.63	-1.89	-2.26
13	1.36	1.18	0.72	0.48	0.14	-0.29	-0.81	-1.46	-1.93						
14	1.57	1.06	0.74	0.56	0.42	0.23	-0.03	-0.26	-0.74	-1.29	-1.56	-2.14			
15	1.19	2.09	0.87	0.73	1.02	0.55	0.54	0.12	-0.22	-0.47	-0.89	-1.50	-2.29		
16	1.51	0.71	0.44	0.24	-0.09	-0.84	-1.45	-2.07	0.00						
17	1.49	0.94	0.76	0.60	0.34	0.14	-0.34	-0.62	-0.96	-1.65	-2.20				
18	0.99	1.50	1.14	0.79	0.35	-0.18	-0.33	-0.79	-1.26	-2.49					
提案モデル ( $L = 3, \delta = 0.1$ )															
課題 $i$	$a_i$	$b_{i0}$	$b_{i1}$	$b_{i2}$	$b_{i3}$	$b_{i4}$	$b_{i5}$	$b_{i6}$	$b_{i7}$	$b_{i8}$	$b_{i9}$	$b_{i10}$	$b_{i11}$	$b_{i12}$	$b_{i13}$
1	1.14	1.14	0.58	0.21	0.05	-0.33	-0.58	-1.04	-1.39	-2.47					
2	1.29	0.97	0.56	-0.02	-0.25	-0.46	-0.93	-1.52	-1.65	-2.31					
3	1.73	0.45	-0.04	-0.44	-0.46	-0.50	-0.85	-1.30	-1.36	-2.56					
4	1.26	1.13	0.41	-0.08	-0.17	-0.38	-0.64	-1.00	-1.28	-1.37	-1.90				
5	1.31	1.00	0.50	0.32	0.11	-0.06	-0.43	-0.47	-0.71	-1.08	-1.55	-2.35			
6	1.84	0.46	0.01	-0.12	-0.29	-0.42	-0.57	-0.61	-0.85	-0.92	-1.00	-1.89	-2.06		
7	1.28	0.80	0.57	0.17	-0.05	-0.08	-0.70	-0.93	-1.52	-1.82	-2.25				
8	1.47	0.57	-0.05	-0.21	-0.31	-0.45	-0.60	-0.76	-1.65	-2.39					
9	1.50	1.01	0.44	0.22	0.10	-0.05	-0.23	-0.26	-0.45	-0.79	-1.26	-1.33	-1.66	-1.76	-2.33
10	1.16	0.76	0.43	0.35	0.18	0.06	0.02	-0.20	-0.36	-0.39	-0.74	-1.26	-1.78	-2.31	
11	1.00	1.24	0.86	0.38	0.22	0.03	-0.09	-0.12	-0.26	-0.40	-1.19	-1.51	-1.74	-2.24	
12	2.11	0.22	-0.16	-0.18	-0.21	-0.24	-0.39	-0.43	-0.65	-0.72	-0.79	-0.87	-1.74	-1.88	-2.07
13	2.12	0.03	-0.30	-0.41	-0.57	-0.77	-1.08	-1.54	-1.69						
14	2.78	-0.06	-0.28	-0.38	-0.42	-0.47	-0.53	-0.59	-0.88	-1.29	-1.45	-1.96			
15	1.60	1.02	0.39	0.13	0.30	0.09	-0.15	-0.45	-0.66	-0.77	-1.07	-1.57	-2.23		
16	2.59	-0.43	-0.65	-0.71	-0.80	-1.31	-1.52	-1.75							
17	2.22	-0.06	-0.18	-0.27	-0.40	-0.48	-0.79	-0.94	-1.16	-1.87	-2.31				
18	1.01	0.96	0.69	0.43	0.09	-0.32	-0.46	-0.87	-1.31	-2.52					

は従来の IRT モデルよりヒントパラメータ  $b_{ik}$  が低く推定される傾向がある。この理由について、次節で能力推定値の差から分析する。

#### 7.4 能力値 $\theta$ の推定

前節では、提案モデルが従来の IRT モデルに比較して学習者のパフォーマンスを正確に予測することを示した。また、推定されるモデルパラメータの値の差異を示した。本節では、提案モデルと従来の IRT モデルで推定される能力推定値の差異について考察する。

ここでは、従来の IRT モデルと  $\delta = 0.1$  に固定した場合の提案モデル ( $L = 15, L = 10, L = 6, L = 3$ ) を用いて、各課題 ( $i = \{1, \dots, 18\}$ ) における各学習者の能力推定値を式 (12) で求め、課題ごとに能力推定値

の平均を算出した。結果を図 5 に示す。図 5 から、従来の IRT モデルは学習過程において  $\theta_{jt}$  が大きく変動していることがわかる。一方、提案モデルでは  $\theta_{jt}$  の変動が小さく、推定能力値が徐々に変化する。従来の IRT モデルは能力推定値が真の能力値より過大評価されるのに対し、提案モデルでは能力値の変動パラメータを最適化して過学習を避けながら能力値の変動を抑えている。前節で示した従来の IRT モデルにおけるヒントパラメータ  $b_{ik}$  の過大推定は、能力値の過大推定が要因であると考えられる。

#### 7.5 予測利用ヒント数の誤差分析

7.3 では提案モデル ( $L = 3, \delta = 0.1$ ) が従来の IRT モデルより正確に学習者のパフォーマンスを予測する

表 5 予測利用ヒント数の過大予測率と過小予測率  
Table 5 Over estimation rates and under estimation rates of the predictive number of hints.

		課題 2	課題 3	課題 4	課題 5	課題 6	課題 7	課題 8	課題 9	課題 10
提案モデル	extra	4.1%	6.5%	9.1%	2.7%	2.9%	3.2%	2.9%	6.3%	2.9%
	missing	43.2%	29.6%	44.0%	46.4%	28.3%	41.1%	34.4%	45.5%	39.9%
従来の IRT モデル	extra	29.3%	15.1%	8.7%	13.7%	14.1%	8.3%	18.3%	15.5%	7.5%
	missing	30.1%	25.1%	45.9%	41.3%	19.7%	39.3%	29.2%	37.6%	37.9%
		課題 11	課題 12	課題 13	課題 14	課題 15	課題 16	課題 17	課題 18	平均
提案モデル	extra	3.5%	2.9%	0.1%	0.5%	4.8%	1.1%	1.9%	3.1%	3.4%
	missing	47.2%	17.3%	16.4%	11.9%	44.7%	2.9%	6.5%	27.7%	31.0%
従来の IRT モデル	extra	21.9%	20.7%	3.3%	4.0%	3.5%	12.9%	2.7%	0.4%	11.7%
	missing	37.9%	11.9%	13.7%	8.4%	46.8%	2.1%	6.7%	30.5%	27.3%

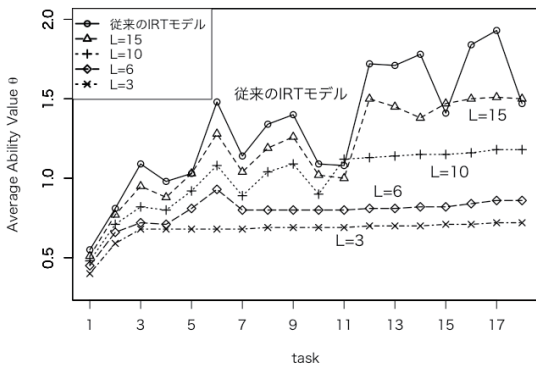


図 5 学習者の能力値の推移

Fig. 5 Transition of learners' average estimated abilities.

ことを示した。ここでは、提案モデル ( $L = 3, \delta = 0.1$ ) と従来の IRT モデルによる予測利用ヒント数の誤差を分析するために、各モデルによる予測利用ヒント数が実際の利用ヒント数より多かった学習者の割合と少なかった学習者の割合を求めた。

結果を表 5 に示す。「extra」は各課題で予測利用ヒント数が実際の利用ヒント数より多かった割合を表し、「missing」は予測利用ヒント数が実際の利用ヒント数より少なかった割合を表す。表 5 より、提案モデル ( $L = 3, \delta = 0.1$ ) は extra が比較的小さく、missing が大きいことから、必要なヒント数を過小評価する傾向にあることがわかる。一方、従来の IRT モデルでは extra が大きく、必要なヒント数を過大に予測する傾向が読み取れる。従来の IRT モデルでは能力値の変動を考慮せずにパラメータ推定を行っているために、実際の能力値の変動に対応したヒントパラメータが推定されておらず、誤差を増加させていると考えられる。Ueno and Miyazawa [10] は支援が過剰な場合と過小な場合とでは、過小な方が学習効率（事前テストと事後

テストの差異）が良いことを報告している。これより、予測利用ヒント数の推定精度が高く、ヒント数を過小に推定する傾向がある提案モデル ( $L = 3, \delta = 0.1$ ) では、従来の IRT に比べて高い学習効率が期待できる。

## 8. むすび

ダイナミックアセスメントは、過去の学習履歴から足場がけによる学習者のパフォーマンスを予測することであるといえる。本論文では、学習履歴データより統計的機械学習を用いてヒント提示後の学習者の正答確率を正確に予測することを目的とした。具体的には、学習者の能力が学習過程において変化していくプロセスを項目反応理論に組み込み、学習者の能力が隠れマルコフ過程に従って変動すると仮定した、新しい隠れマルコフ IRT モデルを提案した。隠れマルコフ IRT モデルでは、従来の IRT モデルにおいて、学習過程によって学習者の能力値が過大評価または過小評価されることを避けるために、学習過程における能力値の変動幅を反映する変動パラメータ  $\delta$  と、学習過程においてある時点での学習者の能力値が継続すると考えられる課題数（ウィンドウサイズ） $L$  をパラメータとして新たに導入した。提案モデルは、ウィンドウサイズ  $L$  と変動パラメータ  $\delta$  をデータから最適化することによって、能力推定に利用する学習データ数と能力値変化の推定精度のトレードオフの問題を解決し、さまざまな学習過程を表現することが可能となる。本論文では、提案モデルを用いた実験によって以下を示した。

(1) 提案モデルが従来の IRT モデルのパフォーマンスの予測精度を大きく改善することを示した。

(2) 実験データに提案モデルを適用することにより、従来の IRT モデルでの能力推定値の過剰評価を避けることが可能となる。

(3) 本論文では、提案されたダイナミックアセス

メントにより支援後の学習者のパフォーマンス予測を向上させた。

実際の足場がけに適用し、学習効率の評価を今後の課題としたい。足場がけは暗記した知識のみを問う課題ではなく、暗記だけでは得られないような、手順が明示化されていない問題解決に適している。そのため、プログラミングのトレース問題に限らず、数学や作文など複雑な問題解決に応用することができ [19], [20]。ただし、Ueno and Miyazawa [10] で開発されたシステムでは、論述問題など正答が一意に決定しないオープンエンドの課題には適応できない。近年、オープンエンドの課題に対してピアアセスメントにより学習者評価を行う項目反応理論が実用化されている [18], [21]。本研究にこのような手法を取り入れることにより、オープンエンドの課題に対応するシステムに改良することも今後の課題としたい。

**謝辞** 本論文の作成にあたり、統計数理研究所の持橋大地先生には大変有益なご助言を頂きました。心より感謝いたします。

## 文 献

- [1] L.S. Vygotsky, *Thought and Language*, Harvard University Press, 1962.
- [2] L.S. Vygotsky, *Mind in Society*, Harvard University Press, 1978.
- [3] J. Bruner, *The Culture of Education*, Harvard University Press, 1996.
- [4] D. Wood, J.S. Bruner, and G. Ross, "The role of tutoring in problem solving," *J. Child Psychiatry and Psychology, and Allied Disciplines*, pp.89–100, 1976.
- [5] A. Collins, "JS & Newman, SE(1989). Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics," Resnick, LB *Knowing, learning and instruction*, pp.453–494, 1989.
- [6] A. Brown and R. Ferrara, "Diagnosing zones of proximal development," *Culture, communication, and cognition: Vygotskian perspectives*, J. Wertsch, ed., pp.273–305, Cambridge, England, Cambridge University Press, 1985.
- [7] D. Wood, "Scaffolding contingent tutoring and computer-supported learning," *Int. J. Artificial Intelligence in Education*, pp.280–292, 2001.
- [8] 植野真臣, 松尾淳哉, "項目反応理論を用いて適応的ヒントを提示する足場かけシステム," *信学論 (D)*, vol.J98-D, no.1, pp.17–29, Jan. 2015.
- [9] M. Ueno and Y. Miyazawa, "Probability based scaffolding system with fading," *Artificial Intelligence in Education - 17th International Conference, AIED 2015*, pp.492–503, 2015.
- [10] M. Ueno and Y. Miyazawa, "IRT-based adaptive hints to scaffold learning in programming," *IEEE Trans. Learning Technologies*, vol.14, no.8, Aug. 2017.
- [11] 情報処理推進機構 (IPA), "情報処理技術者試験 情報処理安全確保支援士試験 出題範囲," 情報処理推進機構 (IPA), [https://www.jitec.ipa.go.jp/1-13download/haniver4\\_0.pdf](https://www.jitec.ipa.go.jp/1-13download/haniver4_0.pdf), 参照 Aug. 2018.
- [12] F.M. Lord and M.R. Novick, *Statistical Theories of Mental Test Scores*, Addison-Wesley, 1968.
- [13] F.B. Baker, and S. Kim, *Item Response Theory: Parameter Estimation Techniques*, Second Ed., Marcel Dekker, 2004.
- [14] F. Samejima, "Estimation of latent ability using a response pattern of graded scores," *Psychometrika Monography*, no.17, pp.1–100, 1969.
- [15] S. Impedovo, A. Ferrante, and R. Modugno, "HMM Based Handwritten Word Recognition System by Using Singularities," *10th International Conference on Document Analysis and Recognition, ICDAR'09*, pp.783–787, 2009.
- [16] J. Ortiz, A.G. Olaya, and D. Borrajo, "A dynamic sliding window approach for activity recognition," *Proc. 19th International Conference on User Modeling, Adaption, and Personalization (UMAP'11)*, pp.219–230, 2011.
- [17] R.J. Patz and B.W. Junker, "Applications and extensions of MCMC in IRT: Multiple item types, missing data, and rated responses," *J. Educational and Behavioral Statistics*, vol.24, no.4, pp.342–366, 1999.
- [18] M. Uto and M. Ueno, "Item response theory for peer assessment," *IEEE Trans. Learning Technologies*, vol.9, no.2, pp.157–170, 2016.
- [19] S. Sei and Y. Miyazawa, "Application of the graded response model of item response theory to computerized dynamic assessment in L2 English education," *J. Information and Systems in Education*, vol.16, no.1, pp.18–25, 2017.
- [20] 榎本 命, 宮澤芳光, 宮寺庸造, 森本康彦, "項目反応理論と穴あきワークシートを用いた適応的プログラミング学習支援システム," *教育システム情報学会誌*, vol.35, no.2, pp.175–191, 2018.
- [21] M. Uto and M. Ueno, "Item response theory without restriction of equal interval scale for rater's score," *International Conference on Artificial Intelligence in Education*, pp.363–368, London, 2018.

## 付 録

ここでは被験者に提示した課題、及び提示したヒントの例を「変数の四則演算」「条件分岐 while ループ」「for ループ」「配列」「関数・メソッド呼び出し」の各領域から 1 題ずつ掲載する。

### 変数の四則演算

以下の課題コード実行時の、変数 a, b, c, d の最終的な値を答えよ

```

1 public class variable{
2     public static void main{
3         int a = 2;
4         int b = 5;
5         int c = 1;
6         int d = 1;
7         a ++;
8         b -= 1;
9         c += a;
10        d += a*2;
11        System.out.println(a);
12        System.out.println(b);
13        System.out.println(c);
14        System.out.println(d);
15    }
16 }

```

ヒント 1：事前に学習したプログラミング文法の再提示

ヒント 2：課題コード 1 行目から 6 行目まで変数 a, b, c, d の値を表示

ヒント 3：課題コード 1 行目から 7 行目までの計算方法を表示

ヒント 4：課題コード 1 行目から 8 行目までの計算方法を表示

ヒント 5：課題コード 1 行目から 9 行目までの計算方法を表示

ヒント 6：課題コード 1 行目から 10 行目までの計算方法を表示

ヒント 7：課題コード 1 行目から 8 行目まで変数 a, b, c, d の値を表示

ヒント 8：課題コード 1 行目から 10 行目まで変数 a, b, c, d の値を表示

#### 条件分岐 while ループ

以下の課題コード実行時の、変数 a, b, c, d の最終的な値を答えよ

```

1 public class IfWhile{
2     public static void main{
3         int a = 1;
4         int b = 8;
5         int c = 1;
6         int d = 3;
7         While(a<3){

```

```

8         c += a;
9         a++;
10    }
11    While(b>6){
12        d += a;
13        b--;
14    }
15    System.out.println(c);
16    System.out.println(d);
17 }
18 }

```

ヒント 1：事前に学習したプログラミング文法の再提示

ヒント 2：課題コード 1 行目から 7 行目まで変数 a, b, c, d の値を表示

ヒント 3：課題コード 1 行目から *while(a < 3)* ループ 1 週目の 9 行目まで変数 a, b, c, d の値を表示

ヒント 4：課題コード 1 行目から *while(a < 3)* ループ 2 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 5：課題コード 1 行目から *while(a < 3)* ループ 2 週目の 9 行目まで変数 a, b, c, d の値を表示

ヒント 6：課題コード 1 行目から *while(a < 3)* ループ 3 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 7：課題コード 1 行目から 11 行目まで変数 a, b, c, d の値を表示

ヒント 8：課題コード 1 行目から *while(b > 6)* ループ 1 週目の 13 行目まで変数 a, b, c, d の値を表示

ヒント 9：課題コード 1 行目から *while(b > 6)* ループ 2 週目の 11 行目まで変数 a, b, c, d の値を表示

ヒント 10：課題コード 1 行目から *while(b > 6)* ループ 2 週目の 13 行目まで変数 a, b, c, d の値を表示

#### for ループ

以下の課題コード実行時の、変数 a, b, c, d の最終的な値を答えよ

```

1 public class ForRoop{
2     public static void main{
3         int a = 3;
4         int i;
5         for(i=1;i<6;i++){
6             if(i/2==1){
7                 a *= 2;
8             }else if(a>10){

```



```

9      break;
10   }
11 }
12   System.out.println(a);
13 }
14}

```

ヒント 1：事前に学習したプログラミング文法の再提示

ヒント 2：課題コード 1 行目から 5 行目まで変数 a, b, c, d の値を表示

ヒント 3：課題コード 1 行目から *for* ループ 1 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 4：課題コード 1 行目から *for* ループ 1 週目の 11 行目まで変数 a, b, c, d の値を表示

ヒント 5：課題コード 1 行目から *for* ループ 2 週目の 5 行目まで変数 a, b, c, d の値を表示

ヒント 6：課題コード 1 行目から *for* ループ 2 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 7：課題コード 1 行目から *for* ループ 2 週目の 11 行目まで変数 a, b, c, d の値を表示

ヒント 7：課題コード 1 行目から *for* ループ 3 週目の 5 行目まで変数 a, b, c, d の値を表示

ヒント 8：課題コード 1 行目から *for* ループ 3 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 9：課題コード 1 行目から *for* ループ 3 週目の 11 行目まで変数 a, b, c, d の値を表示

ヒント 10：課題コード 1 行目から *for* ループ 4 週目の 5 行目まで変数 a, b, c, d の値を表示

ヒント 11：課題コード 1 行目から *for* ループ 4 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 12：課題コード 1 行目から *for* ループ 4 週目の 8 行目まで変数 a, b, c, d の値を表示

### 配列

以下の課題コード実行時の、変数 a, b, c, d の最終的な値を答えよ

```

1 public class Array{
2     public static void main{
3         int []a = {3,5,7};
4         for(int i=0;i<=2;i++){
5             if(a[i]>5){
6                 a[i] = 2;
7             }else if(a[i]<4) {

```

```

8         a[i] = 0;
9     }else{
10        a[i] = 1;
11    }
12 }
13   System.out.println(a[0]); //a[0] = 0
14   System.out.println(a[1]); //a[1] = 1
15   System.out.println(a[2]); //a[2] = 2
16 }
17 }

```

ヒント 1：事前に学習したプログラミング文法の再提示

ヒント 2：課題コード 1 行目から 4 行目まで変数 a, b, c, d の値を表示

ヒント 3：課題コード 1 行目から 7 行目まで変数 a, b, c, d の値を表示

ヒント 4：課題コード 1 行目から 9 行目まで変数 a, b, c, d の値を表示

ヒント 5：課題コード 1 行目から 12 行目まで変数 a, b, c, d の値を表示

ヒント 6：課題コード 1 行目から *for* ループ 2 週目の 5 行目まで変数 a, b, c, d の値を表示

ヒント 7：課題コード 1 行目から *for* ループ 2 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 8：課題コード 1 行目から *for* ループ 2 週目の 9 行目まで変数 a, b, c, d の値を表示

ヒント 9：課題コード 1 行目から *for* ループ 2 週目の 12 行目まで変数 a, b, c, d の値を表示

ヒント 10：課題コード 1 行目から *for* ループ 3 週目の 5 行目まで変数 a, b, c, d の値を表示

ヒント 11：課題コード 1 行目から *for* ループ 2 週目の 7 行目まで変数 a, b, c, d の値を表示

ヒント 12：課題コード 1 行目から *for* ループ 2 週目の 11 行目まで変数 a, b, c, d の値を表示

ヒント 13：課題コード 1 行目から *for* ループ 2 週目の 12 行目まで変数 a, b, c, d の値を表示

### 関数・メソッド呼び出し

以下の課題コード実行時の、変数 a, b, c, d の最終的な値を答えよ

```

1 public class variable{
2     static int a = 10;
3     static int b = 20;

```

```

4  static int c = 30;
5
6  public static void main(String[] args){
7  int m;
8  m = average();
9  System.out.println( m );
10 }
11 public static void average(){
12 int d;
13 d = ( a + b + c ) / 3;
14 return d;
15 }
16 }

```

ヒント 1: 事前に学習したプログラミング文法の再提示

ヒント 2: 課題コード 1 行目から 8 行目まで変数 a, b, c, d の値を表示

ヒント 3: 課題コード 1 行目から 11 行目までの操作の説明を表示

ヒント 4: 課題コード 1 行目から 11 行目までの操作の説明を追加

ヒント 5: 課題コード 1 行目から 13 行目までの操作の説明を表示

ヒント 6: 課題コード 1 行目から 8 行目まで変数 a, b, c, d の値を表示

ヒント 7: 課題コード 1 行目から 14 行目まで変数 a, b, c, d の値を表示

(平成 30 年 3 月 30 日受付, 8 月 25 日再受付,  
10 月 18 日早期公開)



宇都 雅輝 (正員)

2013 年電気通信大学大学院情報システム学研究科博士後期課程修了。博士(工学)。長岡技術科学大学を経て、2015 年より電気通信大学助教に就任、現在に至る。e テスティング, e ラーニング, 人工知能, ベイズ統計, 自然言語処理などの研究に従事。



植野 真臣 (正員)

1992 年神戸大学大学院教育学研究科修了, 1994 年東京工業大学大学院総合理工学研究科修了。博士(工学)。東京工業大学, 千葉大学, 長岡技術科学大学を経て 2006 年より電気通信大学助教授, 2013 年より教授, 現在に至る。



堤 瑛美子

2018 年電気通信大学情報理工学部情報・通信工学科卒。同年、電気通信大学大学院情報理工学研究科情報・ネットワーク工学専攻入学、現在に至る。アダプティブ・ラーニングの研究・開発に従事。