論　文

Knowledge Tracing のための Sliding Window 隠れマルコフ IRT

堤　瑛美子[†a)]　　木下　　涼[†b)]　　植野　真臣[†c)]

Sliding Window Hidden Markov Item Response Theory for Knowledge Tracing

Emiko TSUTSUMI[†a)], Ryo KINOSHITA[†b)], and Maomi UENO[†c)]

あらまし　近年，人工知能分野では教育ビッグデータを分析することにより，学習過程における学習者の能力値や知識状態を把握し，課題への反応予測を行う Knowledge Tracing が注目されている．学習者の能力変化を正確に把握するためには，能力値推定の際に学習者の学習履歴データを適切に忘却する必要がある．しかし，既存手法における忘却パラメータは予測精度を最大にするための最適化が困難であった．本論文では Knowledge Tracing のために Sliding Window 方式によって過去の学習データを忘却する Sliding Window 隠れマルコフ IRT を提案する．提案手法は学習データの忘却度を決定する離散値の忘却パラメータ Window size をもち，交差検証を用いて容易に最適化することができる．評価実験では提案手法と BKT，IRT，DKT 関連の既存手法の予測精度比較を行い，提案手法の有効性を示す．

キーワード　Bayesian Knowledge Tracing，Deep Knowledge Tracing，Item Response Theory，教育ビッグデータ，パフォーマンス予測

1. まえがき

近年，コンピュータやタブレット端末の普及に伴なってオンライン学習システムを用いた学習が広まり，大量の学習履歴データ（学習者の課題への反応データ）を容易に入手できるようになった．人工知能分野では学習データを分析することにより，学習過程における学習者の能力値や知識状態を把握し，未知の課題への反応を予測することが課題となっている．教師側は学習者の能力値や知識状態を把握することで学習者の未習熟の課題を同定し，個人の成長に最適な指導を行うことが可能となる．学習過程における過去の学習データから現在の能力値や知識状態を推定する手法は確率的アプローチ ([1]〜[12]) とディープラーニングアプローチ ([13]〜[15]) に大別できる．確率的アプローチでは Bayesian Knowledge Tracing (BKT) [2] と項目反応理論 (IRT : Item Response Theory) [1] が知られている．BKT は学習過程における学習者の知識状態の変化を隠れマルコフモデルで表現した数理モデルである．学習者が課題解決に必要なスキルをどの程度習得しているかを推定し，課題への正答確率を予測する．IRT は過去の学習データをもとに学習者の知識状態を表す連続隠れ変数を推定し，未知の課題への正答確率の予測を行う数理モデルである．しかし，BKT では学習者の知識状態が 2 値のみの隠れ変数で表現されており，IRT では学習過程における継時的な能力値の変化が考慮されていない．そのため，BKT，IRT は学習過程での能力値や知識状態の変化を柔軟に表現することができないという問題がある．

近年，学習過程での学習者の能力値を隠れマルコフ過程に従って時系列変化させた隠れマルコフ IRT(HMIRT : Hidden Markov IRT) が複数提案されている [6]〜[12]．HMIRT は知識状態を連続値で表現した BKT の一般化，能力値の時系列変化を考慮した IRT の一般化手法であると解釈できる．

一方，ディープラーニングアプローチとしては Deep Knowledge Tracing (DKT) が知られている [13]．DKT は過去の学習データと特徴量を利用し，Long short-term memory (LSTM) を用いて学習者の知識状態を推定し，未知の課題への反応を予測する．

† 電気通信大学大学院情報理工学研究科，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: kinoshita@ai.lab.uec.ac.jp
c) E-mail: ueno@ai.is.uec.ac.jp
DOI:10.14923/transinfj.2020JDP7026

近年の研究では，これらの確率的アプローチとディープラーニングアプローチの予測精度の優位性が議論されており，各アプローチの予測精度を比較した実験が行われている [11], [14], [15]．Wilson ら [8], [11] は確率的アプローチ手法として IRT，難易度パラメータに事前分布を付与した Hierarchical IRT (HIRT)，過去の学習データを忘却する機能をもつ新たな HMIRT (TIRT : Temporal IRT) と DKT との比較実験を行い，三つの確率的アプローチ手法が DKT の精度を上回ることを示した．彼らは DKT にはチューニングパラメータが多く，その設定が難しいと指摘している．また，DKT で推定されるパラメータは解釈可能性が低いことも問題として挙げている．

近年，DKT の予測精度を向上させるために，各スキルの習得状態を保存する Memory Network を用いた Dynamic Key-Value Memory Network (DKVMN) が提案されている [14]．更に，DKT や DKVMN のパラメータの解釈可能性を向上させた手法として，DKVMN と IRT を組み合わせた Deep-IRT [15] が提案されている．Deep-IRT は高い予測精度とパラメータの解釈可能性を両立していることが報告されており，注目を集めている．

反応予測を最適化するためには，学習過程における学習者の能力値変化を正確に把握する必要がある．このためには，能力値推定の際に学習履歴データを適切に忘却しなければならない．なぜならば，学習者の能力が学習や忘却によって変化する場合，古い学習履歴データは現在の能力を正しく反映していない場合があり，どの時点のデータを能力値推定に用いるか（どの時点以前のデータを忘却するか）によって，反応予測精度が変化してしまうからである．そのため，反応予測精度を最大化するようにどの時点以前のデータを忘却するかを決めなければならない．

しかし，忘却パラメータをもつ既存手法では予測精度を最大化するように学習データの忘却を行うことは難しい．例えば，TIRT は連続値の忘却パラメータをもつが，一般に連続値のチューニングパラメータは探索範囲が広すぎるために最適化するのが難しい．また，LSTM などのディープラーニング手法は「forget gate」と呼ばれる忘却パラメータをもつ．forget gate は現在の能力値が過去の学習データにどれだけ依存するかを決めるパラメータであるが，データへのフィッティングを最適化するように推定されており，交差検証などを用いた予測の最適化は行われていないため訓練データに対して過学習を起こす可能性がある [14]．DKT では過学習を防ぐためにドロップアウトなどの工夫がされているが，これらを用いても忘却パラメータの予測最適化は理論的には保証されていない．DKVMN や Deep-IRT における忘却パラメータは TIRT と同様にパラメータが連続値かつ探索空間が膨大なため，最適化が難しい．

これらの問題を解決するために，本論文では Knowledge Tracing のための Sliding Window 隠れマルコフ IRT (SHMIRT: Sliding HMIRT) を提案する．Sliding Window 方式は画像処理や音声処理，通信工学などの分野で用いられる手法であり，指定した長さの Window が学習者の学習履歴データ上を移動し，Window 内の学習履歴データのみを用いて能力値推定を行う [16]〜[19]．つまり，Window 外の学習履歴データは忘却する．Window size は過去の学習履歴データの忘却度を決定するパラメータであり，予測が最大になるように最適化する必要がある．本論文で提案されている Window size は簡単な離散値で表現されるため，連続値の忘却パラメータより探索範囲が狭く，交差検証などを用いて容易に最適化することができる．このため，パラメータの過学習を防ぐことができることも特徴である．

本研究では，これまで学習過程での学習者の能力値推定に用いられてきた手法 (Logistic Hidden Markov Model [5], IRT [1], HIRT [11], TIRT [8], DKT [13], DKVMN [14], Deep-IRT [15]) と提案手法との予測精度比較を行う．学習期間・学習者数・課題数の異なる様々な学習データを用いて実験を行い，提案手法の有効性を示す．

ただし，堤ら [20] でも Sliding Window 隠れマルコフ IRT を提案しているが，適応的なヒントを提示するための段階ヒントモデルであり，Knowledge Tracing のための本論の目的とは異なる．

2. 学習者の能力値（知識状態）推定モデル

本章では Knowledge Tracing に関する能力値や知識状態の推定モデルを紹介する．本論文では学習履歴データにおける学習者数を I，課題数を M と表し，学習者 i の課題 m に対する反応データ x_{im} を以下で表す．

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

$$X = \{x_{im}\}, (i = 1, \cdots, I, m = 1, \cdots, M)$$

2.1 Bayesian Knowledge Tracing (BKT)

BKT は学習過程での学習者の知識状態の変化を隠れマルコフモデルで表現した数理モデルであり，学習者の過去の学習履歴データから課題解決に必要なスキル（例えば算数であれば「加算，減算，乗算，除算」など）への知識状態を推定する [2]．学習者の未習熟なスキルを同定することで次に学習者が取り組むべき課題を予測することが可能になる [21], [22]．BKT では学習者が課題に対するスキルを習得しているかしていないかを 2 値の隠れ変数で表現する．近年では，学習者や課題の個々の特性を考慮した BKT の拡張手法が提案されている [3]〜[5]．

最も新しい研究として，Pelánek ら [5] は BKT における知識状態の変化をより詳細に把握するため，知識状態を段階的な離散値に拡張し，予測反応確率がロジスティック関数に従う Logistic Hidden Markov Model(LHMM) を開発した．LHMM は学習者と課題の相関を表す識別力や学習の難易度をパラメータとして組み込んでおり，課題の特性を考慮した推定が可能となっている．学習者 i の知識状態が $s \in \{1, \cdots, S\}$ であるとき，課題 m に正答する確率を以下で表す．

$$p(x_{im} = 1 | Z_{im} = s) = \frac{1}{1 + \exp(-a(s/(S-1) - b)} \quad (1)$$

ここで，S は知識状態の段階数，a は識別力パラメータ，b は難易度パラメータを表し，Z_{im} は学習者 i の課題 m に対する知識状態が値 s をとる確率変数を示す．LHMM では知識状態の低下と 2 段階以上の遷移は起こらないと仮定されており，知識状態の遷移が制限されていることからモデルとしての柔軟性に欠けるという問題がある．

2.2 Item Response Theory (IRT)

学習支援システムの枠組みでは，IRT は過去の学習データをもとに課題項目の特性パラメータを推定し，学習者の課題への反応データより能力値を推定し，未知の課題への反応を予測する [11], [20], [23], [24]．IRT は学習者の能力値を連続隠れ変数で推定するため，BKT で推定される 2 値の離散値の知識状態隠れ変数より柔軟な解釈が可能となる．ここでは，IRT 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，能力値 θ_i の学習者 i が課題 m に正答する確率を次式で表す．

$$p(x_{im} = 1 \mid \theta_i) = \frac{1}{1 + \exp(-a_m(\theta_i - b_m))} \quad (2)$$

ここで，a_m は課題 m の識別力パラメータ，b_m は課題 m の難易度パラメータ，θ_i は学習者 i の能力値隠れ変数を表す．項目パラメータ a_m，b_m は学習データから事前に推定した値を用いる．標準的な IRT では学習過程での学習者の能力値は固定値であるため，学習者の能力値変化は反映されていない．

2.3 ディープラーニング手法

本章では，ディープラーニングを用いた学習者の反応予測手法 (DKT,DKVMN,Deep-IRT) を紹介する．BKT は各スキルのデータを別々に入力することしかできなかったが，DKT は全てのスキルのデータを同時に入力できる [13]．また，DKT は学習者の知識状態を Long short-term memory (LSTM) の高次元変数として推定し，予測精度が BKT を上回ることが報告されている [13]．しかし，DKT の推定値は各スキルに対応しておらず，解釈可能性が低いという問題がある．

近年，DKT の予測精度を向上させるために，各スキルの習得状態を保存する Memory Network を用いた Dynamic Key-Value Memory Network (DKVMN) が提案されている [14]．DKVMN は高い予測精度を示すことが知られているが，DKT と同様にパラメータの解釈可能性が低いという問題があった．そこで，DKT や DKVMN のパラメータの解釈可能性を向上させた手法として，DKVMN と IRT を組み合わせた Deep-IRT [15] が提案されている．Deep-IRT は高い予測精度とパラメータの解釈可能性を両立していることが報告されており，注目を集めている．また，国内でも同時期に Deep Learning と IRT を組み合わせた Item Deep Response Theory が提案されている [25]．ただし，本手法はテスト理論として開発されており，能力値の時系列的変化には対応していない．

ディープラーニング手法が高い予測精度を示す理由の一つに，現在の能力値が過去の学習データにどれだけ依存するかを決める忘却パラメータ「forget gate」をもつことが挙げられる．しかし，DKT ではデータへのフィッティングを最適化するように推定されており，交差検証などを用いた予測の最適化をしていないため訓練データに対して過学習を起こす可能性が指摘されている [14]．DKT では過学習を防ぐためにドロップ

アウトなどの工夫がされているが，これらを用いても忘却パラメータの予測最適化は理論的には保証されない．DKVMN や Deep-IRT における忘却パラメータは連続値かつ探索空間が膨大であるため最適化が難しい．

2.4 時系列 IRT モデル

一方，近年，確率的アプローチでは，学習過程での学習者の能力値変化を把握するために，学習者の能力値変数を隠れマルコフモデルに従って変化させる隠れマルコフ IRT(HMIRT : Hidden Markov IRT) が提案されている [6]〜[12]．これらのモデルは離散値の知識状態を連続値の隠れ変数で表現した BKT 手法の一般化モデルともいえる．

Wilson ら [8], [11] が提案した Temporal IRT (TIRT) は，学習データの忘却パラメータをもつ HMIRT モデルであり，学習の経過時間によって能力値推定に用いる学習データを徐々に忘却させる手法である．TIRT では時点 t での課題への予測正答確率を以下のようにモデル化している．

$$p(x_{im} = 1 \mid \theta_{it}) = \frac{1}{1 + \exp\left(-\tilde{a}_{\Delta_t}(\theta_{it} - b_m)\right)} \tag{3}$$

$$\tilde{a}_{\Delta_t} = \frac{a_m}{\sqrt{1 + \sigma a_m^2 \Delta_t}} \tag{4}$$

ただし，

$$\theta_{it} \sim N(\theta_{it-1}, \sigma),\ \theta_{i0} \sim N(0, 1) \tag{5}$$

θ_{it} は時点 t での学習者 i の能力値を表す．Δ_t は課題に取り組んだ時点からの経過時間を表し，σ は学習データの忘却度を決定する忘却パラメータである．σ は連続値のパラメータであり，$\sigma > 0$ のとき，経過時間が増加するごとに識別力 $\tilde{a}_{\Delta_t}$ が小さくなり徐々に過去の学習データを忘却する．$\sigma = 0$ の場合には学習データを忘却しない一般的な IRT と一致する．Wilson ら [11] では，TIRT の忘却パラメータを学習データに最適化することにより，DKT より高い予測精度が得られることが示されている．しかし，一般に連続値のチューニングパラメータは探索範囲が広すぎるために最適化するのが難しいという問題がある．

3. Sliding Window 隠れマルコフ IRT (Sliding HMIRT)

前章では学習過程での学習者の能力値変化をモデル化した BKT，HMIRT 手法や学習データの忘却パラメータをもつディープラーニング手法，TIRT を紹介した．学習者の課題への反応を正確に予測するためには，過去の学習データを適切に忘却し，学習過程での能力値変化をより正確に把握する必要がある．しかし，ディープラーニング手法や TIRT における忘却パラメータは予測精度についての最適化が困難であった．

これらの問題を解決するために，本論文では Knowledge Tracing のための Sliding Window 隠れマルコフ IRT (SHMIRT: Sliding HMIRT) を提案する．SHMIRT では Sliding Window 方式により学習者の能力値推定を行う．Sliding Window 方式は画像処理や音声処理，通信工学などの分野で用いられる手法であり，指定した長さの Window が学習者の学習履歴データ上を移動し，Window 内の学習データのみを用いて能力値を推定する [16]〜[19]．すなわち，Sliding Window の長さを示す Window size は過去の学習履歴データの忘却度を決定するパラメータであり，Window 外の過去の学習履歴データを忘却する．

Window size は予測精度が最大になるように最適化する必要があるが，本論文で用いる Window size $L = \{1, 2, \cdots, M\}$ は少数の離散値で表現されるため，連続値の忘却パラメータより探索範囲が狭く，交差検証などを用いて容易に最適化することができる．また，提案手法は能力値の変動幅を制限する分散パラメータ σ をもち，このパラメータの機能によって過学習を防ぐことができることも特徴である．

SHMIRT は課題 $m = L$ 以降の能力値推定において，図 1 のように推定に用いる学習課題を 1 題ずつずらすことで能力値の変化を反映する．この手法は Window の重複をなくすことで学習過程を複数期間に分割して能力値推定を行う Martin らの手法 [9] の一般化でもある．また，前述のように，堤ら [20] は Sliding Window 隠れマルコフ IRT を提案しているが，適応的なヒントを提示するためのヒントへの多段階モデルであり，

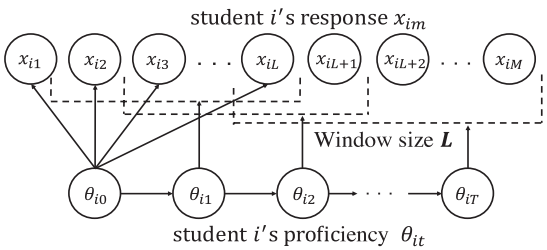

図 1　Sliding HMIRT

Knowledge Tracing には用いることはできない．

SHMIRT では時点 t において学習者 i が課題 m に正答する確率 P_{imt} を次式で表す．

$$P_{imt} = \frac{1}{(1 + \exp(-a_m(\theta_{it} - b_m))} \tag{6}$$

a_m は課題 m の識別力パラメータ，b_m は課題 m の難易度パラメータ，θ_{it} は時点 t での学習者 i の能力値を表す．ただし，

$$\theta_{it} \sim N(\theta_{it-1}, \sigma) \tag{7}$$

$$\theta_{i0} \sim N(0, 1) \tag{8}$$

σ は能力値の変動を制限する分散パラメータである．

SHMIRT は Window size L と分散パラメータ σ の組合せを変化させることで，多様な学習過程を表現することができる．Window size L と分散パラメータ σ の関係は以下のとおりである．

（1） L と σ が共に小さい

L が小さいために θ_{it} が影響する課題数が少なく，σ が小さいため能力値の変動もほぼ起こらない．このため，それまでの学習過程に関係なく θ_{it} がほとんど変化しないモデル．

（2） L が小さく σ が大きい

L が小さいために直前の学習過程にのみ影響され，ゆう度に対して事前分布の影響が大きくなり，σ が大きいため θ_{it} の急激な上下変動が起こるモデル．

（3） L と σ が共に大きい

L が大きいためにそれまでの学習過程に強く影響を受け，ゆう度が事前分布に対して相対的に大きくなり，σ が大きいため θ_{it} が学習過程全体で上方向（若しくは下方向）に大きく変動するモデル．

（4） L が大きく σ が小さい

L が大きいためにそれまでの学習過程の影響を強く受けるが，σ が小さいため θ_{it} の急激な変動を抑制するモデル．

実際には，交差検証などを用いて学習データごとに学習者の予測を最大にする組合せを最適値として決定する．次章で，本モデルのパラメータ推定法について述べる．

4. パラメータ推定法

本研究ではパラメータ推定法として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) を用いる [26], [27]．堤ら [20] ではデータに最適な Window size L と能力値の分散パラメータ σ を貪欲法を用いて求めていたが，学習データが膨大になるほど，最適な組合せを求めるのに膨大な時間を要するという問題があった．本研究では，推定時間を短縮し，大規模学習データに適応させるために σ を MCMC アルゴリズムで推定する．また，堤ら [20] ではできなかった学習者ごとに課題の出題順序が異なる場合にもパラメータ推定を可能とするため，学習者 j が n 番目に解いた課題番号を表す順序データ $\boldsymbol{J_i} = \{J_{i1}, \cdots, J_{ik}, \cdots, J_{in}\}$ を設定する．

ここで，各パラメータの集合をそれぞれ $\boldsymbol{\theta} = \{\theta_{10}, \cdots, \theta_{IT}\}, \boldsymbol{a_m} = \{a_{m=1}, \cdots, a_{m=M}\}, \boldsymbol{b_m} = \{b_{m=1}, \cdots, b_{m=M}\}$，各事前分布をそれぞれ $g(\theta_{it}|\sigma)$, $g(a_m)$, $g(b_m)$, $g(\sigma)$ と表す．このとき，反応データ $\boldsymbol{X}$，順序データ $\boldsymbol{J_i}$ 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．

$$\begin{aligned} &p(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b} \mid \boldsymbol{X}) \\ &\propto L(\boldsymbol{X} \mid \boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}) g(\boldsymbol{a}) g(\boldsymbol{b}) g(\boldsymbol{\theta}|\sigma) g(\sigma) \\ &= \left[\prod_{t=0}^{T} \prod_{n=t+1}^{L+t+1} (P_{iJ_{in}t})^{x_{iJ_{in}}} (1 - P_{iJ_{in}t})^{1 - x_{iJ_{in}}}\right] \\ &\quad \left[\prod_{m=1}^{M} g(a_m) \cdot g(b_m)\right] \left[\prod_{i=1}^{I} \prod_{t=0}^{T} g(\theta_{it}|\sigma)\right] g(\sigma) \end{aligned} \tag{9}$$

MCMC の手法のうち，メトロポリスヘイスティングス法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．

$$\begin{aligned} &\log a_m \sim N(0.0, 0.2) \\ &\theta_{i0} \sim N(0.0, 1.0) \\ &\theta_{it} \sim N(\theta_{it-1}, \sigma) \\ &\sigma \sim IG(1.0, 1.0) \\ &b_m \sim N(0.0, 1.0) \end{aligned}$$

（2） $\boldsymbol{\theta_i} = \{\theta_{i0}, \cdots, \theta_{iT}\}$ を現在の推定値 $\boldsymbol{\theta_i}'$ に依存する提案分布 $q(\boldsymbol{\theta_i} \mid \boldsymbol{\theta_i}')$ 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．

$$\begin{aligned} &\alpha(\boldsymbol{\theta_i} \mid \boldsymbol{\theta_i}') \\ &= \min\left(\frac{L(\boldsymbol{X_i} \mid \boldsymbol{\theta_i}, \boldsymbol{a}', \boldsymbol{b}', \sigma') \prod_{t=0}^{T} g(\theta_{it})}{L(\boldsymbol{X_i} \mid \boldsymbol{\theta_i}', \boldsymbol{a}', \boldsymbol{b}', \sigma') \prod_{t=0}^{T} g(\theta'_{it})}, 1\right) \end{aligned} \tag{10}$$

提案分布には $N(\boldsymbol{\theta_i}', \sigma \mathbf{1_T})$ を用いる．ここで，1_n は

Algorithm 1 MCMC algorithm

Given maximun chain length S,burn-in B,interval E
Initialize MCMC sample $A \leftarrow \phi$
Initialize $\boldsymbol{\theta}^0$,$\boldsymbol{a}^0$,$\boldsymbol{b}^0$,σ^0
for $s = 1$ to S **do**
  for $i \in \{1 \cdots I\}$ **do**
    Sample $\boldsymbol{\theta}_i^s \sim N(\boldsymbol{\theta}_i^{s-1}, \sigma \mathbf{1}_T)$
    Accept $\boldsymbol{\theta}_i^s$ *with the probability* $\alpha(\boldsymbol{\theta}_i^s \mid \boldsymbol{\theta}_i^{s-1})$
  end for
  for $m \in \{1 \cdots M\}$ **do**
    Sample $a_m^s \sim N(a_m^{s-1}, \sigma 1)$
    Accept a_m^s *with the probability* $\alpha(a_m^s \mid a_m^{s-1})$
    Sample $b_m^s \sim N(b_m^{s-1}, \sigma 1)$
    Accept b_m^s *with the probability* $\alpha(b_m^s \mid b_m^{s-1})$
  end for
  Sample $\sigma^s \sim N(\sigma^{s-1}, \sigma 1)$
  Accept σ^s *with the probability* $\alpha(\sigma^s \mid \sigma^{s-1})$
  if $s \geq B$ *and* $s\%E = 0$ **then**
    $A \leftarrow (\boldsymbol{\theta}^s, \boldsymbol{a}^s, \boldsymbol{b}^s, \boldsymbol{\sigma}^s)$
  end if
end for
return average value of A

$n \times n$ の単位行列を表す．

（3） パラメータ a_m, b_m, σ 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 thining 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 Algorithm1 に示す．

5. 評価実験

本章では，これまで学習過程での学習者の能力値推定に用いられてきた手法 (LHMM [5], IRT [1], HIRT [11], TIRT [8], DKT [13], DKVMN [14], Deep-IRT [15]) と提案手法における学習者の予測精度比較を行う．ここで，IRT 手法とディープラーニング手法の違いに注意する必要がある．IRT 手法における学習者の課題への反応は独立であることが仮定されているため，同じ課題に繰り返し取り組む学習には適応できない．更に，前節までで述べた IRT 手法は学習に複数のスキルが含まれるような多次元のスキルを考慮していない．したがって，ディープラーニング手法と IRT 手法ではスキルに対する能力値を完全に公平に比較することは難しいが，本研究では一般的なオンライン学習環境において各課題への学習者の反応を予測し，予測精度の比較を行う．

5.1 データに最適な Window size の決定

提案手法は忘却パラメータ (Window size)L を学習データごとに最適化する必要がある．本研究では，10 分割交差検証で実験を行い，以下の手順で算出される予測精度を最大にする L を忘却パラメータの最適値とする．

（1） 学習データの 9 割を訓練データとして，L を所与として課題の識別力パラメータ $\boldsymbol{a}$，難易度パラメータ $\boldsymbol{b}$，能力値の分散パラメータ σ を MCMC アルゴリズムで推定する．

（2）（1）で求めたパラメータを所与とし，学習データの残りの 1 割をテストデータとして学習者 i の能力値 $\boldsymbol{\theta}_i$ を MCMC アルゴリズムで推定する．学習者 i が n 番目に解く課題 J_{in} で正答する確率 $P_{iJ_{in}t}$ を求め，$P_{iJ_{in}t} > 0.5$ のとき $\hat{x}_{iJ_{in}} = 1$，$P_{iJ_{in}t} \leq 0.5$ のとき $\hat{x}_{iJ_{in}} = 0$ を予測反応 $\hat{x}_{iJ_{in}}$ とする．$P_{iJ_{in}t}$ の計算に利用する各学習者の能力値 $\hat{\theta}_{it}$ は，以下の方法で求める．

（a） $n \leq L$ のとき

課題 J_{in} 以前のデータ $\boldsymbol{x}_i^{(J_{in-1})} = \{x_{i1}, \cdots, x_{iJ_{in-1}}\}$ を用いて以下の EAP 推定法で求める．

$$\hat{\theta}_{i0} = E[\theta_{i0} \mid \boldsymbol{x}_i^{(J_{in-1})}] = \frac{\int_{-\infty}^{+\infty} \theta_{i0} g(\theta_{i0}) L(\boldsymbol{x}_i^{(J_{in-1})} \mid \theta_{i0}) d\theta_{i0}}{\int_{-\infty}^{+\infty} g(\theta_{i0}) L(\boldsymbol{x}_i^{(J_{in-1})} \mid \theta_{i0}) d\theta_{i0}} \tag{11}$$

ここで，

$$L(\boldsymbol{x}_i^{(J_{in-1})} \mid \theta_{i0}) = \prod_{k=1}^{n-1} (P_{iJ_{ik}t})^{x_{iJ_{ik}}} \tag{12}$$

実際には，式中の積分は $-3.0 < \theta_{i0} < 3.0$ での 100 点の区分求積法を用いて近似値を求める．

（b） $n > L$ のとき

MCMC アルゴリズムで推定した $\hat{\theta}_{it}(t = 1, \cdots, T)$ を用いる．

（3） 学習者 i の課題 J_{in} における実際の反応データ $x_{iJ_{in}}$ と予測反応 $\hat{x}_{iJ_{in}}$ を用いて，各学習者 i における一致割合 Acc_i を次式で求める．

$$Acc_i = \frac{1}{n-1} \sum_{k=2}^{n} \psi(x_{iJ_{ik}}, \hat{x}_{iJ_{ik}}) \tag{13}$$

ここで，$\psi(x_{iJ_{ik}}, \hat{x}_{iJ_{ik}})$ は $x_{iJ_{ik}}$ と $\hat{x}_{iJ_{ik}}$ 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）で求めた一致割合を全ての学習者

表 1　小規模学習データ

学習データ	学習者数	課題数	正解率	平均解答数
プログラミング 1	148	7	60.4%	7
プログラミング 2	75	18	65.8%	18
離散数学	77	125	45.3%	125

表 2　小規模データにおける予測精度

学習データ		DKT	DKVMN	Deep-IRT	LHMM	IRT	HIRT	TIRT	Proposed
プログラミング 1	Acc	0.702	0.669	0.747	0.642	0.696	0.700	**0.758**	0.747
	AUC	0.761	0.690	0.716	0.681	0.754	0.758	**0.866**	0.818
	F1	0.692	0.577	0.620	0.580	0.667	0.663	0.701	**0.720**
プログラミング 2	Acc	0.645	0.611	0.739	0.696	0.712	0.713	0.736	**0.762**
	AUC	0.660	0.661	0.744	0.659	0.757	0.758	**0.828**	0.822
	F1	0.564	0.526	0.695	0.561	0.634	0.632	0.574	**0.702**
離散数学	Acc	0.734	0.650	0.732	0.624	0.732	0.732	0.732	**0.746**
	AUC	0.801	0.722	0.788	0.616	0.799	0.796	0.799	**0.816**
	F1	0.724	0.627	0.724	0.531	0.721	0.717	0.721	**0.734**
Average	Acc	0.694	0.643	0.739	0.654	0.713	0.715	0.742	**0.755**
	AUC	0.741	0.691	0.749	0.652	0.770	0.771	**0.831**	0.819
	F1	0.660	0.577	0.680	0.557	0.674	0.671	0.698	**0.723**

について平均し，予測精度 Acc として次式を求める．

$$Acc = \frac{1}{I}\sum_{i=1}^{I} Acc_i \tag{14}$$

L は $L = 1$ を初期値として徐々に大きくしていき，Acc が最大になる L を最適値として採用する．比較手法のチューニングパラメータも交差検証で最適値を決定する必要がある場合は上記と同様に決定する．

5.2　小規模学習データ

ここでは，e-ラーニングシステム「Samurai」[23], [24], [28]〜[31] を用いて収集した小規模の時系列学習データを用いて実験を行う．学習データはいずれも大学生を対象に一つの授業で収集されたものである．表 1 に各学習データの詳細を示す．表中の正解率は学習データ全体の正答の割合，平均解答数は学習者が各学習で解答した課題数の平均値（学習過程の長さ）を表す．本研究で用いる小規模学習データにはスキルタグが存在しないため，各学習データに一つのスキルのみが含まれると仮定する．

本実験では，各手法の予測精度を比較するために予測反応と実データの一致割合 (Acc)，AUC，F 値を算出した．各手法におけるチューニングパラメータの決定を公平に行うため，チューニングパラメータの候補数を五つに統一した．実験結果を表 2 に示す．提案手法における忘却パラメータの最適値は $L = \{1, 2, 3, 4, 5\}$ から推定し，「プログラミング 1」では $L = 2$，「プログラミング 2」では $L = 3$，「離散数学」では $L = 4$ が最適値となった．Wilson ら [11] には HIRT と TIRT におけるチューニングパラメータの最適化方法が明記されていないため，本研究では先行研究 [11] におけるパラメータ最適値を参考に最適値周辺の五つの候補を用いた．HIRT にはチューニングパラメータとして課題の難易度パラメータのハイパーパラメータ σ, τ が存在する．σ, τ の最適値は{0.05, 0.1, 0.3, 0.5, 1.0} × {0.0, 0.1,0.3,0.5, 1.0}から交差検証を用いて予測精度を最大にする組合せに決定した．TIRT におけるデータの忘却パラメータ σ は $\sigma = \{0.0, 0.01, 0.1, 0.5, 1.0\}$ から HIRT と同様に推定した結果，離散数学では $\sigma = 0.0$，それ以外では $\sigma = 0.1$ が最適値となった．また，ディープラーニング手法の隠れ層の次元数と DKVMN，Deep-IRT のメモリの次元数も同様に $\{10, 50, 100, 150, 200\}$ から最適値を決定した．メモリの次元数以外の調整すべきチューニングパラメータは先行研究 [14], [15] で最適化された値を用いた．また，本実験で用いた各手法のチューニングパラメータの候補値を表 3 にまとめた．

表 2 より，AUC の平均値では TIRT が提案手法より高い値を示したが，Acc と F 値の平均値は提案手法が他の手法と比較して最も良い値を示した．TIRT の結果に注目すると，学習者の平均解答数が増えるほど予測精度が低下する傾向が見られる．つまり，TIRT におけるデータ忘却は学習過程が短い場合に有効であり，学習過程が長くなるほど有効性が減少することが分かる．一方，提案手法は平均解答数にかかわらず高い予測精度を示している．提案手法の忘却パラメータ L は最適化が容易であるために，TIRT の忘却パラメータより有効に機能したと考えられる．

表 3　実験に用いた各手法のチューニングパラメータ候補値

手法	候補値
提案手法	$L = \{1, 2, 3, 4, 5\}$
HIRT	$\sigma = \{0.05, 0.1, 0.3, 0.5, 1.0\}$ $\tau = \{0.0, 0.1, 0.3, 0.5, 1.0\}$
TIRT	$\sigma = \{0.0, 0.01, 0.1, 0.5, 1.0\}$
DKT,DKVMN Deep-IRT	メモリの次元数 {10, 50, 100, 150, 200}

学習者の能力値変化を考慮しない IRT と HIRT では，Wilson ら [11] で示された結果と同様に HIRT が IRT よりわずかに高い予測精度を示した．彼らの実験では，HIRT は他の手法に比べて最も予測精度が高いと報告していたが，本研究では同様の結果を得られなかった．これは，学習データに一つのスキルしか設定されていないために，課題の難易度パラメータにおけるスキル別の事前分布が機能しなかったことが原因であると考えられる．

また，LHMM は IRT の予測精度を下回る結果となった．単純な離散値の知識状態をもつ LHMM に対して，連続値で能力値を推定する IRT は表現力が高く，より真の能力値に近い推定が可能であることが分かる．

ディープラーニング手法では，「離散数学」の学習データにおける AUC と F 値が他の学習データに対して高い精度を示した．「離散数学」は他の学習データと比べて平均解答数が多い学習データであり，ディープラーニング手法は比較的長い学習過程で有効であるといえる．また，ディープラーニング手法のうち，最も高い予測精度を示した Deep-IRT は，TIRT と提案手法を除いた確率的アプローチ手法に対しても優位な結果となった．Deep-IRT は IRT 手法とディープラーニング手法を組み合わせた手法であるため，両者の利点を活かすことで高い予測精度を示したと考えられる．

5.3　大規模学習データ

本節では，オンライン学習システムで収集された大規模なベンチマークデータセット ASSISTments2009 (注1)，Statics2011 (注2)，KDDcup2006-2007 (注3) を用いて予測精度比較を行う．大規模学習データの詳細を表 4 に示す．表中の平均解答数は学習者が解答した全課題数の平均値，すなわち学習過程の長さを表し，括弧内は 1 スキル当りの平均解答数を表す．スパース率は 10 人（5 人）以下の学習者が解答した課題の割合を表し，課題パラメータが少数データから推定された割合を示す．LHMM と IRT 手法を用いた実験では，ASSISTments2009，Statics2011 の学習データについてスキルタグでデータを分割し，スキルごとに独立した学習データと仮定して予測を行った．ASSISTments2009 には課題に取り組んだ学習者数が極端に少ないスキルが存在するため，解答数が 30 人未満のスキルデータを除いたデータセット ASSISTments2009*も作成した．ディープラーニング手法においては IRT 手法と同様に課題への反応を入力値とする場合と，Piech ら [13] と同様に各スキルへの反応を入力とする場合の両方の予測精度を算出する．スキル入力では同じスキルの全ての課題を等価とみなし，共通の課題パラメータをもつ．また，本研究ではデータの偏りを避けるために，入力する学習データの上限を学習者 1 人につき 200 問とした [15].

各手法ごとに算出した Acc，AUC，F 値を表 5 に示す．本実験でも各手法におけるチューニングパラメータは表 3 の候補値から推定した．提案手法における忘却パラメータの最適値は $L = \{1, 2, 3, 4, 5\}$ から推定し，全ての学習データで $L = 2$ となった．HIRT における σ, τ の最適値は{0.05, 0.1, 0.3, 0.5, 1.0} × {0.0, 0.1,0.3,0.5, 1.0}から交差検証を用いて予測精度を最大にする組合せに決定した．TIRT におけるデータの忘却パラメータ σ は $\sigma = \{0.0, 0.01, 0.1, 0.5, 1.0\}$ から HIRT と同様に推定し，Statics2011 では $\sigma = 0.0$，それ以外では $\sigma = 0.1$ が最適値となった．また，ディープラーニング手法の隠れ層の次元数と DKVMN，Deep-IRT のメモリの次元数も同様に $\{10, 50, 100, 150, 200\}$ から最適値を決定した．メモリの次元数以外の調整すべきチューニングパラメータは先行研究 [14], [15] で最適化された値を用いた．

結果より，Acc，AUC，F1 の全ての平均値で提案手法が他の手法の予測精度を上回った．特に，提案手法は最先端のディープラーニング手法である DKVMN と Deep-IRT より高い予測精度を示しており，学習データの忘却を最適化する提案手法はディープラーニング手法より効果的であることが分かる．TIRT は提案手法より予測精度が低く，**5.2** でも述べたとおり学習期間が長くなるにつれて精度が低下する傾向がある．実際に平均解答数が短い ASSISTments2009 では TIRT は提案手法より高い予測精度を示し，KDDcup2006-2007 では TIRT の AUC と F1 は他の手法より高い値を示している．Wilson ら [11] の実験では HIRT が他の IRT

（注1）：https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data
（注2）：https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507
（注3）：https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

表 4　大規模学習データ

学習データ	学習者数	スキル数	課題数	正解率	平均解答数	スパース率
ASSISTments2009	3,776	111	26,587	68.0%	70.8(8.62)	55.2%(33.1%)
ASSISTments2009*	2,944	49	2,635	63.8%	42.2(12.4)	0%(0%)
Statics2011	229	41	1,095	77.7%	180.9(15.3)	2.56%(1.46%)
KDDcup2006-2007	820	43	476	78.3%	11.9(11.9)	57.8%(22.7%)

表 5　大規模データにおける予測精度

学習データ		DKT		DKVMN		Deep-IRT		LHMM	IRT	HIRT	TIRT	Proposed
		item	skill	item	skill	item	skill	skill	item	item	item	item
ASSISTments2009	Acc	0.765	0.759	0.637	0.763	0.683	**0.768**	0.679	0.720	0.724	0.757	0.738
	AUC	0.800	0.781	0.659	0.807	0.710	0.806	0.737	0.785	0.786	0.804	**0.831**
	F1	0.713	0.697	0.602	0.714	0.647	**0.718**	0.526	0.636	0.593	0.559	0.631
ASSISTments2009*	Acc	0.690	0.689	0.681	0.690	0.681	0.699	0.656	0.701	0.701	0.707	**0.753**
	AUC	0.682	0.693	0.696	0.718	0.702	0.719	0.704	0.755	0.755	0.763	**0.819**
	F1	0.608	0.633	0.618	0.641	0.619	0.640	0.533	0.609	0.609	0.621	**0.671**
Statics2011	Acc	0.769	0.777	0.805	0.780	0.817	0.787	0.798	0.816	0.819	0.816	**0.831**
	AUC	0.666	0.652	0.819	0.721	0.822	0.722	0.650	0.819	0.816	0.819	**0.825**
	F1	0.483	0.461	0.679	0.521	**0.681**	0.526	0.471	0.581	0.577	0.581	0.627
KDDcup2006-2007	Acc	0.777	0.784	0.760	0.773	0.779	**0.792**	0.586	0.733	0.733	0.759	0.750
	AUC	0.549	0.538	0.565	0.594	0.561	0.588	0.588	0.614	0.527	**0.676**	0.675
	F1	0.439	0.439	0.464	0.439	0.447	0.455	0.319	0.522	0.522	**0.561**	0.553
Average	Acc	0.750	0.752	0.721	0.751	0.740	0.762	0.680	0.743	0.744	0.760	**0.768**
	AUC	0.674	0.666	0.685	0.710	0.699	0.709	0.660	0.743	0.721	0.765	**0.788**
	F1	0.561	0.557	0.591	0.579	0.598	0.585	0.462	0.575	0.576	0.581	**0.620**

手法に比べて最も予測精度が高いモデルとして示されていたが，本実験では IRT との顕著な差はみられなかった．これは，Wilson ら [11] にはチューニングパラメータの最適化方法が明記されていないため交差検証を用いて最適化を行ったが，チューニングパラメータが連続値であるために探索範囲が広く最適化が難しいことが原因として考えられる．また，パラメータ推定法が MCMC であったことも原因の一つかもしれない．

ディープラーニング手法においては，スキル入力 (skill) が課題入力 (item) を上回る結果となった．課題への反応を入力とした場合には推定するパラメータ数が膨大になり過学習を起こすため，正しく推定が行われていない可能性がある．ASSISTments2009，KDDcup2006-2007 では提案手法はディープラーニング手法より予測精度が低い．表 4 より，ASSISTments2009，KDDcup2006-2007 ではスパース率が高く，1 問当りの解答者数が少ないことを表している．このため，提案方法はディープラーニング手法よりスパースデータに対して脆弱である可能性がある．一方，ASSISTments2009*と Statics2011 では，各課題のデータが十分に大きいため，提案手法が他の手法よりも高い予測精度を示すことが分かる．

6. 課題パラメータ推定

前章までで，提案手法は既存手法と比較して予測精度が高いことを示した．本章では，提案手法と従来の IRT [1] で推定された課題パラメータを比較しながら分析する．表 6 は提案手法と IRT で推定された各小規模学習データの識別力パラメータ $\boldsymbol{a}$ と難易度パラメータ $\boldsymbol{b}$ の相関係数を表している．更に，図 2 に識別力パラメータの推定値の散布図を示した．表 6 と図 2 から識別力パラメータは「プログラミング 2」や「離散数学」のように学習過程が長いほど二つの識別力パラメータの相関係数が小さくなることが分かる．IRT は学習過程で学習者の能力値が固定されているため学習が進捗すると能力値の推定値は収束する．しかし，提案手法は能力値が常に変動するため学習過程が長いほど識別力パラメータの分散が大きくなり，IRT の識別力パラメータと特性が異なってくると考えられる．一方，「プログラミング 2」，「離散数学」を除いた学習データでは識別力パラメータの相関係数が大きい．これらの学習

表 6　提案手法と IRT における課題パラメータ推定値の相関係数

学習データ	識別力 a	難易度 b
プログラミング 1	0.590	0.996
プログラミング 2	0.359	0.985
離散数学	0.065	0.948
ASSISTments2009	0.459	0.860
ASSISTments2009*	0.540	0.938
Statics2011	0.478	0.778
KDDcup2006-2007	0.463	0.882

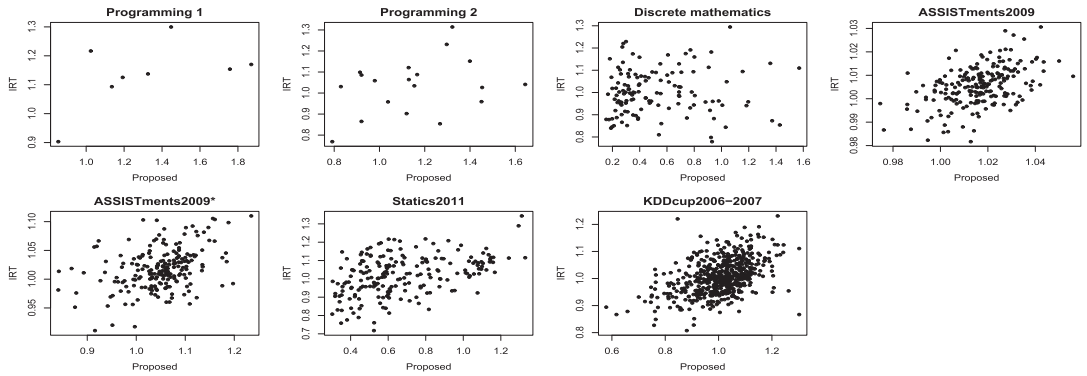

図 2　課題パラメータ推定値の相関図

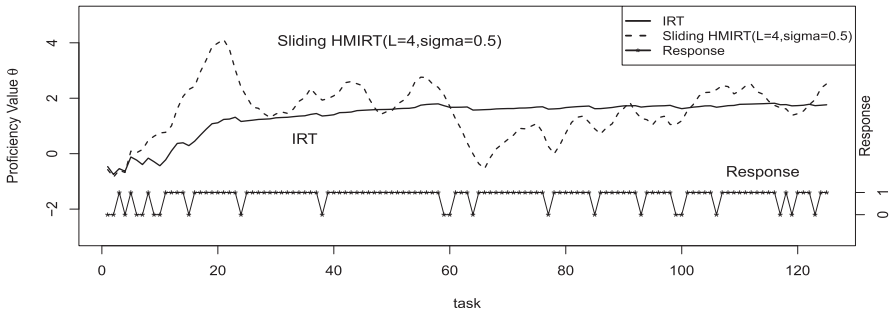

図 3　学習者の能力値推移

データは 1 スキル当りの平均解答数が少ないことから学習過程が短く，推定値の特性に大きな差が出なかったためであると考えられる．難易度パラメータは学習の長さに関係なく，大きな相関係数を示した．

7. 学習者の能力推定

本章では，提案手法と標準的な IRT で推定される能力値推定値の差異について考察する．この実験では長期的な学習過程における学習者の能力値変化を比較するために，以下の二つの指標を算出した．

（1） 各学習者の全ての学習過程における能力値の最大値と最小値の差の平均値（範囲）

$$\frac{1}{I}\sum_{i=1}^{I}\left(\max_{1\le k\le n}\boldsymbol{\theta}_{iJ_{ik}}-\min_{1\le k\le n}\boldsymbol{\theta}_{iJ_{ik}}\right) \tag{15}$$

ここで，$\theta_{iJ_{ik}}$ は学習者 i が k 番目の課題 J_{ik} に解答した時点での能力値を表す．

（2） 課題 J_{ik} を解答した時点での能力値変動の平均値（変動幅）

$$\frac{1}{I\cdot(n-1)}\sum_{i=1}^{I}\sum_{k=2}^{n}|\theta_{iJ_{ik}}-\theta_{iJ_{ik-1}}| \tag{16}$$

表 7　提案手法と IRT における能力値推定値の差異（標準誤差）

学習データ	モデル	範囲	変動幅
プログラミング 1	Proposed	1.272 (0.213)	0.324 (0.026)
	IRT	0.868 (0.080)	0.254 (0.008)
プログラミング 2	Proposed	2.041 (0.291)	0.203 (0.018)
	IRT	1.311 (0.099)	0.132 (0.013)
離散数学	Proposed	4.117 (0.295)	0.219 (0.009)
	IRT	1.547 (0.253)	0.043 (0.002)
ASSISTments2009	Proposed	0.850 (0.011)	0.382 (0.003)
	IRT	0.584 (0.009)	0.331 (0.002)
ASSISTments2009*	Proposed	1.392 (0.019)	0.302 (0.004)
	IRT	0.896 (0.012)	0.238 (0.003)
Statics2011	Proposed	1.399 (0.067)	0.302 (0.013)
	IRT	0.752 (0.016)	0.185 (0.004)
KDDcup2006-2007	Proposed	0.842 (0.053)	0.304 (0.020)
	IRT	0.706 (0.023)	0.283 (0.007)

結果を表 7 に示す．前章でも述べたとおり，IRT では学習過程が長いほど能力値が収束していくのに対し，提案手法では能力値が変動し続けるため能力値の分散が大きくなる傾向がある．このため，上の二つの指標は提案手法が IRT より大きくなっており，学習過程での能力値変化の差を反映している．特に学習過程の長い「離散数学」では提案手法と IRT の差が顕著である．

図 3 は「離散数学」における提案手法 (Window size $L=4, \sigma=0.5$) と IRT で推定された，ある学習者の能

力値推移である．縦軸は左側が学習者の能力値，右側が学習者の課題への反応，横軸は課題を表す．学習者の課題への反応は学習者が課題へ正答したときに 1，誤答したときに 0 となる．図 3 より，IRT は学習がある程度進むと能力値の推定値が収束し，変動しないことが分かる．一方，提案手法は学習者の反応に従って能力値が変動し続けている．**3.** で述べたとおり，提案手法 (Window size $L = 4, \sigma = 0.5$) は L が小さく，σ が比較的大きいため能力値は直前の学習データのみに影響し，能力値の時間的な変動が大きいモデルとして推定されている．図 3 からも提案手法による能力推定値はある時点で正答が続く場合に上昇し，誤答すると低下する傾向が見られる．提案手法は交差検証により忘却パラメータを最適化するため，長期の学習過程の学習者の能力値変動を適切に推定に反映できるので，予測精度が高いと考えられる．

8. むすび

近年，人工知能分野では教育ビッグデータを分析することにより，学習過程における学習者の能力値や知識状態を把握する Knowledge Tracing が注目されている．学習者の能力値変化を正確に把握するためには，能力値推定の際に学習者の学習履歴データを適切に忘却する必要がある．本論文では Knowledge Tracing のために SlidingWindow 方式によって過去の学習データを忘却する SlidingWindow 隠れマルコフ IRT を提案した．提案手法では学習データの忘却度を決定する Window size が離散値であるため，予測を最大にする最適な忘却パラメータを交差検証などで容易に求めることが可能である．評価実験では BKT，IRT，DKT 関連の既存手法と提案手法を用いて学習者の予測精度比較を行い，提案手法が既存手法の予測精度を改善することを示した．

近年，ディープラーニング手法における能力値推定では一つの課題に複数のスキルが存在すると仮定し，スキルごとの学習者の達成度を予測する研究が行われている．提案手法は補償型の IRT モデル [32] に拡張することで，あるスキルにおける能力値が別のスキルの能力値の高さによって補償されるモデルになる．今後は学習者の能力値をより詳細に把握するため，多次元のスキルに適応するモデルに拡張していきたい．

謝辞 本研究は JSPS 科研費 JP19H05663 の助成を受けたものです．

文　献

[1] F.B. Baker and S.H. Kim, Item Response Theory: Parameter Estimation Techniques, Second Edition, Statistics: A Series of Textbooks and Monographs, Taylor & Francis, 2004.

[2] A.T. Corbett and J.R. Anderson, “Knowledge tracing: Modeling the acquisition of procedural knowledge,” User Modeling and User-Adapted Interaction, vol.4, no.4, pp.253–278, Dec. 1994.

[3] González-Brenes, Jose, Huang, Yun, and Brusilovsky, “Fast: Feature-aware student knowledge tracing,” NIPS 2013 Workshop on Data Driven Education., 2013.

[4] M.V. Yudelson, K.R. Koedinger, and G.J. Gordon, “Individualized bayesian knowledge tracing models,” Artificial Intelligence in Education, pp.171–180, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[5] R. Pelánek, “Conceptual issues in mastery criteria: Differentiating uncertainty and degrees of knowledge,” 19th International Conference on Artificial Intelligence in Education, vol.1, pp.450–461, 06 2018.

[6] F. Pennoni, F. Bartolucci, and G. Vittadini, “Assessment of school performance through a multilevel latent markov rasch model,” J. Educational and Behavioral Statistics, 2011.

[7] D. Molenaar, D. Oberski, J. Vermunt, and P.D. Boeck, “Hidden markov item response theory models for responses and response times,” Multivariate Behavioral Research, vol.51, pp.606–626, 2016.

[8] C. Ekanadham and Y. Karklin, “T-skirt: Online estimation of student proficiency in an adaptive learning system,” CoRR, vol.abs/1702.04282, 2017.

[9] A.D. Martin and K.M. Quinn, “Dynamic ideal point estimation via markov chain monte carlo for the u.s. supreme court, 1953–1999,” Political Analysis, vol.10, pp.134–153, 2002.

[10] J.H. Park, “Modeling preference changes via a hidden markov item response theory model,” Handbook of Markov Chain Monte Carlo, pp.479–491, 2011.

[11] K.H. Wilson, Y. Karklin, B. Han, and C. Ekanadham, “Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation,” Proc. 9th International Conference on Educational Data Mining, vol.1, pp.539–544, 2016.

[12] W. Xiaojing, J.O. Berger, and D.S. Burdick, “Bayesian analysis of dynamic item response models in educational testing,” Annals of Applied Statistics, vol.7, no.1, pp.126–153, 2013.

[13] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L.J. Guibas, and J. Sohl-Dickstein, “Deep knowledge tracing,” Advances in Neural Information Processing Systems 28, eds. by C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, and R. Garnett, pp.505–513, Curran Associates, 2015.

[14] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, “Dynamic Key-Value Memory Networks for Knowledge Tracing,” Proc. 26th International Conference on World Wide Web, pp.765–774, WWW ’17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017.

[15] C. Yeung, “Deep-irt: Make deep learning based knowledge tracing explainable using item response theory,” Proc. 12th International Conference on Educational Data Mining, EDM, 2019.

[16] Y. Chi, H. Wang, P.S. Yu, and R.R. Muntz, “Catch the moment:

maintaining closed frequent itemsets over a data stream sliding window," Knowl Inf Syst, vol.10, pp.265–294, March 2006.
[17] G. Piella and H. Heijmans, "A new quality metric for image fusion," Proc. 2003 International Conference on Image Processing (Cat. No.03CH37429), pp.III–173, 2003.
[18] Y. Tao and D. Papadias, "Maintaining sliding window skylines on data streams," IEEE Trans. Knowl. Data Eng., pp.479–491, 2006.
[19] G.K. Venayagamoorthy, V. Moonasar, and K. Sandrasegaran, "Voice recognition using neural networks," Proc. 1998 South African Symposium on Communications and Signal Processing-COMSIG '98 (Cat. No.98EX214), pp.29–32, 1998.
[20] 堤瑛美子，宇都雅輝，植野真臣，"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，" 信学論（D），vol.J102-D, no.2, pp.79–92, Feb. 2019.
[21] J.E. Beck, K. m.Chang, J. Mostow, and A. Corbett, "Does help help? introducing the bayesian evaluation and assessment methodology," Proc. 9th International Conference on Intelligent Tutoring Systems, pp.383–394, June 2008.
[22] M.A.S. Pedro, R.S.J. deBaker, and J.D. Gobert, "Incorporating scaffolding and tutor context into bayesian knowledge tracing to predict inquiry skill acquisition," EDM, 2013.
[23] M. Ueno and Y. Miyazawa, "Probability Based Scaffolding System with Fading," Artificial Intelligence in Education - 17th International Conference, AIED, pp.237–246, 2015.
[24] M. Ueno and Y. Miyazawa, "Irt-based adaptive hints to scaffold learning in programming," IEEE Trans. Learning Technologies, vol.11, pp.415–428, Oct. 2018.
[25] 木下　涼，植野真臣，"深層学習によるテスト理論：Item Deep Response Theory," 信学論（D），vol.J103, no.4, pp.314–329, April 2020.
[26] 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.
[27] M. Uto and M. Ueno, "Item response theory for peer assessment," IEEE Trans. Learning Technologies, vol.9, pp.157–170, 06 2016.
[28] M. Ueno, "Data mining and text mining technologies for collaborative learning in an ILMS "samurai"," IEEE Int. Conf. Advanced Learning Technologies, ICALT 2004, pp.1052–1053, 2004.
[29] M. Ueno, "Intelligent lms with an agent that learns from log data," E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pp.3169–3176, 2005.
[30] M. Ueno, "Animated pedagogical agent based on decision tree for e-learning," IEEE Int. Conf. Advanced Learning Technologies (ICALT'05), pp.188–192, 2005.
[31] M. Ueno and M. Ando, "Analysis of the advantages of using tablet pc in e-learning," 10th IEEE Int. Conf. Advanced Learning Technologies, pp.122–124, 2010.
[32] M.D. Reckase, "Multidimensional item response theory," Springer, New York, 2009.

（2020 年 3 月 23 日受付，6 月 8 日再受付，
8 月 14 日早期公開）

堤　瑛美子

2020 電気通信大学大学院情報理工学研究科博士前期課程了．同年，電気通信大学大学院情報理工学研究科博士後期課程入学，在学中．

木下　　涼

2018 電気通信大学大学院情報理工学研究科博士前期課程了．同年，電気通信大学大学院情報理工学研究科博士後期課程課程入学，在学中．

植野　真臣　（正員）

1992 神戸大学大学院教育学研究科了，1994 東京工業大学大学院総合理工学研究科了．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 より電気通信大学助教授，2013 より教授，現在に至る．