

# 卒業論文

## Deep Knowledge Tracing のための スパースデータに頑健な特徴ベクトルの検証

指導教官 村上 陽平 准教授

立命館大学情報理工学部

情報コミュニケーション学科 4 回生

2600160258-5

津山 陽紀

2019 年度(秋学期)卒業研究 3(2Q)

令和 2 年 1 月 31 日

# Deep Knowledge Tracing のための スパースデータに頑健な特徴ベクトルの検証

津山 陽紀

## 内容梗概

近年、情報技術と教育分野を組み合わせによって生徒の学習に役立つ、Learning Analytics と呼ばれる動きが活発化している。これは MOOC (Massive Open Online Course) などのオンライン学習サービスにより膨大な量の生徒の学習記録を自動で収集することができるようになったためである。オフライン環境では収集の困難であった膨大な学習データの分析を行うことで、個人に最適化された学習サービスを提供することが可能となった。このような学習サービスを実現するために、現在、KnowledgeTracing と呼ばれる、生徒の学習記録からそれぞれの生徒の学習習熟度を予測するタスクが注目されている。その手法の一つとして、リカレントニューラルネットワークを用いた DeepKnowledgeTracing がある。これは、知識同士の関係性と知識獲得の時系列性を考慮した包括的な KnowledgeTracing を行うことができる。具体的には、過去の生徒の問題解答履歴からその生徒が次に解く問題カテゴリの正答確率を予測するものである。しかしながら、従来の方法では個々の問題の正答確率ではなく、問題カテゴリの正答確率のため、個々の問題の差異までは考慮できていなかった。一つの問題を一つの問題カテゴリに割り当てることも可能であるが、カテゴリの数が増えることで予測精度の低下を招く。

そこで、本研究では DeepKnowledgeTracing の手法を利用して入力するデータのパラメータ数を増やし、問題と問題カテゴリを併用することで、次に解く問題の正答確率の予測精度の向上を目的とする。本目的を達成するために取り組んだ課題は以下の2点である。

## パラメータの同定

既存の DeepKnowledgeTracing では問題カテゴリごとに生徒の正誤を予測しており、入力するデータはどの生徒が解いたか、どの分野の問題を解いたか、正解したかという三つ組みを元にカテゴリごとの正誤を予測している。まずはこれを問題ごとの正誤の予測に変更し、もう一つのパラメータをどのデータにすると精度が向上するかどうかを検証する必要がある。

## 学習データのエンコーディング

上記で選定したパラメータを含めた四つ組みをどのようなベクトルとして符号化すると最も精度が向上するのかを検証する必要がある。

提案手法の取り組みのため、既存の DeepKnowledgeTracing のプログラムとオンライン学習サービス ASSISTments が提供するオープン学習データを利用した。ASSISTments から「生徒 ID・問題 ID・スキル ID・問題の正誤・問題の試行回数」のデータを利用した。スキル ID はどの分野の問題であるかを番号で表したものである。問題の試行回数は生徒がその問題を解く時、それが何回目であるかを表す。データとしてこの4つのうちのどれかが欠けているものは省いてデータを抽出し入力に利用している。例えば、ASSISTments のスキル ID は空欄になっている場合があるので、その並びにあるデータは利用しない。

### **パラメータの同定**

「スキル ID」と「問題の試行回数」というパラメータを入力ベクトルに追加し、有効性を検証することで有用なパラメータを同定した。「生徒 ID・問題 ID・問題の正誤」にスキル ID を加えたものでは有効性は確認できなかった。一方で、学習データ量を削減した上で「問題の試行回数」を入力ベクトルに加えると、1.0%ほどの精度向上が見られた。この結果からスパースなデータでは、「問題の試行回数」が予測精度の向上に寄与することが確認された。

### **学習データのエンコーディング**

今回は「生徒 ID・問題 ID・スキル ID・問題の正誤」の入力ベクトルに関して、問題 ID の情報を先に与えるか、スキル ID の情報を先に与えるかという2つのエンコーディングを検証したが、その両方の精度の有意な差は見られなかった。

# **Verification of feature vectors robust to sparse data for Deep Knowledge Tracing**

Akinori Tsuyama

## **Abstract**

Recently, the motion which is Combining information technology and education called 'learning analytics' gets active. This is because we can collect enormous student's learning record automatically by learning services such as MOOC. By analyzing a huge amount of leaning data which was difficult to collect in an offline environment, it became possible to provide learning services optimized for individuals. In order to realize such a learning service, a task called Knowledge Tracing, which predicts each student's learning proficiency from student's learning records, is attracting attention. One of the methods is Deep Knowledge Tracing by using recurrent neural network. It can perform comprehensive Knowledge Tracing considering the relationship between knowledge and the time series of knowledge gaining. Specifically, it predicts the correct answer probability of the next question category that the student solves by using the past student's answer history. However, in the conventional method, the correct answer probability of each question category was considered instead of the correct answer probability of each question, so the difference of each question could not be considered. It is possible to assign one question to one problem category, but the prediction accuracy decreases because of the increasing the number of categories. Therefore, in this research, the number of parameters of input data is increased, and the problem and the problem category are used together by using the method of Deep Knowledge Tracing for the purpose of improving the accuracy of predicting correct answer probability of the next problem. We addressed the following two issues to achieve this purpose.

## **Parameter identification**

Existing Deep Knowledge Tracing predicts student correctness for each problem category. The data to be entered are three: which students solved, which categories of the problem they solved, and whether they answered correctly. We replace this with a problem-by-problem prediction. We need to verify if the other parameter is what data improves accuracy.

### **Encoding of training data**

It is necessary to verify the method that gives the highest accuracy when encoding four data including the selected parameter.

To work on the proposed method, we used the existing Deep Knowledge Tracing program and the open learning data provided by the online learning service ASSISTments. We used the data of "student ID / problem ID / skill ID / correct or incorrect" from ASSISTments. The skill ID is a number that indicates the field of the problem. We omit any of these four data that are missing and use them for input.

### **Parameter identification**

We have added parameters "skill ID" and "number of trials" to the input vector and identified useful parameters by verifying their effectiveness. The validity could not be confirmed by adding the student ID to the "student ID / problem ID / correct or incorrect". From these results, it was confirmed that "the number of trials of the problem" contributed to the improvement of the prediction accuracy for sparse data.

### **Encoding of training data**

We have analyzed the input vector of "student ID / problem ID / skill ID / correct or incorrect". We tested two encodings: giving the problem ID information first or the skill ID information first. However, there was no significant difference between the two precisions.

Deep Knowledge Tracing のための  
スパースデータに頑健な特徴ベクトルの検証

## 目次

第 1 章 はじめに	1
第 2 章 様々な Knowledge Tracing の手法	4
2.1 Bayesian Knowledge Tracing	4
2.2 Deep Knowledge Tracing	5
2.3 本研究でのアプローチ	7
第 3 章 パラメータの同定	7
第 4 章 学習データのエンコーディング手法	12
第 5 章 評価	15
第 6 章 考察	19
第 7 章 おわりに	21
参考文献	21

## 第1章 はじめに

近年、情報技術と教育分野を組み合わせによって生徒の学習に役立てる、Learning Analytics と呼ばれる分野が活発化している。Learning Analytic（以下、LA と略す）では、学習における様々な活動を記録したデータを自動的に収集・分析をして、学習効率の改善に役立てることを目指している。

このような分析は教育現場に ICT を導入することで可能になっている。ICT を用いるメリットとして、オンライン上での課題提出や、時間や場所にとらわれない学習などが注目されるが、教育においては学習者の学習記録や行動の記録、すなわちログデータを集積できる点でメリットが大きい。

オフラインの環境下でも学習者の行動を観察し、その情報について分析することは可能であるが、あくまでも学校の一クラス程度の小規模な環境である。学習者が 50 人、100 人、1000 人と増えたとしても同じようにデータの集積が可能になることがオンラインの強みである。

たとえば、大規模に公開されたオンライン講座である MOOC (Massive Open Online Course) では、オンラインの環境を通じて海外や遠方にある学習機関の講義を視聴することが可能であり、世界中の人々が無償で受講している。MOOC は 2008 年にアメリカで始まったが、2012 年以降に世界のトップ大学が一斉にこれに参入したことによって世界中で急速に普及し、研究テーマとして取り上げられる関心も大きく広がっている。

この MOOC における LA として、感情情報の利用というものがある。学習者の顔の位置を検出し、幸せな状態や驚き、悲しみといった離散的な感情の抽出を行う。また、音声からもポジティブであるかネガティブであるかという感情を読み取るということも可能である。

また、学習の開始時と終了時に学習者の感情の状態を報告させ、テキストの表示やチャットの利用等のイベント情報も収集している。このような感情の情報から学習者の学習効率との相関関係を明らかにできる。

したがって、学習者の感情の取得・分析・利用は LA において重要な役割を果たす要素の一つと言える。

このように、学習者の分析はオンライン上で大規模なデータを取得することによって実現されている。その中でも、これまでは教師が経験則によって行っていた学習者の習熟度の推定は生徒の今後の学習に直接活かせる分野

として非常に重要である。この学習習熟度の推定は、学習者の学習過程をモデル化した学生のモデルを通して行われる。

こういったモデリングの代表的な既存手法として Knowledge Tracing が挙げられる。Knowledge Tracing は学生モデリングの手法の一つであり、デファクトスタンダードとしての地位を確立している。

本研究で扱うのはその一つであるリカレントニューラルネットワークを用いた Deep Knowledge Tracing という手法である。これは過去の生徒の問題解答履歴からその生徒が次に解く問題カテゴリの正答確率を予測するものである。しかしながら、従来の方法では個々の問題の正答確率ではなく、問題カテゴリごとの正答確率なため、個々の問題の差異までは考慮できていなかった。一つの問題を一つの問題カテゴリに割り当てることも可能であるが、カテゴリの数が増えることで予測精度の低下を招く。

そこで、本研究では Deep Knowledge Tracing の手法を利用して入力するデータのパラメータ数を増やし、問題と新たに追加した別のパラメータを併用することで、次に解く問題の正答確率の予測精度の向上を目的とする。

提案手法の取り組みのため、Deep Knowledge Tracing のプログラムを利用した。また、その入力としてオンライン学習サービス ASSISTments が提供するオープン学習データを使った。

本研究の目的を達成するために取り組んだ課題は以下の2点である。

### **パラメータの同定**

既存の Deep Knowledge Tracing では問題カテゴリごとに生徒の正誤を予測しており、入力するデータはどの生徒が解いたか、どの分野の問題を解いたか、正解したかという三つ組みを元にカテゴリごとの正誤を予測している。まずはこれを問題ごとの正誤の予測に変更し、もう一つのパラメータをどのデータにすると精度が向上するかどうかを検証する必要がある。

### **学習データのエンコーディング**

上記で選定したパラメータを含めた四つ組みをどのようなベクトルとして符号化すると最も精度が向上するのかを検証する必要がある。

以下本論文では、2章において Knowledge Tracing の関連研究について述べ、その技術について説明する。続いて3章では検証に利用する特徴ベクトルのパラメータについてオープンデータの中から有効なものを同定する。その後、4章においては特徴ベクトルをどのように符号化すれば解答予測の精

度が向上するかを検証する．5章では3章や4章での回答予測精度の検証についての評価を行う．6章では5章で行なった評価を考察し，7章にて今後の展望や課題について述べて結論とする．

## 第2章 Knowledge Tracing

### 2.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing は学習者のスキルの習得状態が隠れマルコフモデルに従うと仮定した数理モデルである。学習履歴のデータから学習者が課題を解決するために必要な知識をどれほど習得しているかを確率で表すことが可能である。

Bayesian Knowledge Tracing ではある時間  $t$  における学習者のある課題に対する習熟度を以下の 2 通りで表す。

- 学習者が課題を解くのに必要なスキルを習得している
- 学習者が課題を解くのに必要なスキルを習得していない

また、学習者の課題に対する反応を以下の 2 通りで表す。

- 学習者が課題に正答
- 学習者が課題に誤答

学習者の時間  $t+1$  におけるスキルの習得状態に遷移する際に以下の 4 つの確率を用いる。

- $P(L_0)$  は学習者があらかじめ知識を習得している確率
- $P(T)$  は学習者があらかじめ知識を習得していない状態から習得している状態に変化する確率
- $P(S)$  は学習者があらかじめ知識を習得しているにもかかわらず課題に誤答してしまう確率

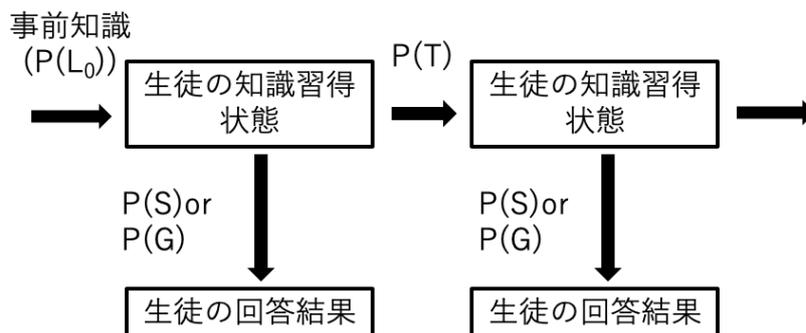


図 1. Bayesian Knowledge Tracing のダイアグラム

- $P(G)$ は学習者があらかじめ知識を習得していない状態にもかかわらず課題に正答する確率

この Knowledge Tracing はあらかじめ設定されたそれぞれの知識に対して図 1 のようなダイアグラムで表される。このダイアグラムは学習者のスキルの状態の遷移とその時点での回答結果を表現している。

Bayesian Knowledge Tracing では学習者のある知識についての習熟度を求めることができるが、その精度は学習者ごとに最適化されていないという点において不完全である。

## 2.2 Deep Knowledge Tracing

Deep Knowledge Tracing (以下, DKT と呼ぶ) はディープラーニングを用いて学生の知識をモデル化する手法である。その構造はリカレントニューラルネットワークの構造に基づいている。

リカレントニューラルネットワークとはあるその出力がもう一度その層へ入力される回帰結合を持つニューラルネットワーク全般を指している。このような結合を持つことによって、ニューラルネットワークは過去の情報を保持することが可能となる。それによって時系列情報を考慮したネットワーク構造となっている。

リカレントニューラルネットワークの利用例としては、言葉を予測するというものが挙げられる。「今日は実に気持ちの良い晴天だ」という文章があるとすると、その文章の中で「晴天」というワードを予測したいとすると、そこまでの文章の「実に気持ちの良い」という情報から関連づけて「晴天」という単語を予測することができる。「晴天」という単語がこの文章だけから予測できて他の文章を必要としないのは過去の学習結果を利用することができるからである。このように単純な場合はリカレントニューラルネットワークで学習がしやすいと考えられるが、あまりにも長期的に依存関係があるものに対しては予測が困難である。

DKT の基本的なモデル構造について、時刻  $t$  における入力を  $x_t$ , 隠れ層を  $h_t$ , 出力を  $y_t$  とすると、その関係性は以下の式で表すことができる。

- $h_t = f(x_t, h_{t-1})$ 
  - $y_t = g(h_t)$

表 1. ASSISTments のパラメーター一覧

		本研究での利用
各パラメータ	生徒ID	有
	問題ID	有
	正誤	有
	ヒントを求めた回数	無
	問題の試行回数	有
	スキルID	有
	問題を解くのにかかった時間	無
	スキル名	無
	最初の行動が問題を解いたか ヒントに行ったか	無

モデル関数は  $f$  と  $g$  によって定義されている。

最適化は解答の正誤予測に関する損失関数を最小化することによって行われる。学習済みの DKT のモデルからは知識獲得の時系列性と知識間の関係性を考慮した包括的な知識間の関係性を表すネットワークを抽出することが可能である。

例えば、ある学習者が数学の問題に取り組んでいるとする。その学習者が2次関数を学んだ後に微分の問題に取り組むと、正答しやすいといった事象を考慮することができる。

DKT は時刻  $t$  における学習者の解答における正誤ベクトル  $x_t$  を入力として時刻  $t+1$  における解答の正誤を予測するベクトル  $y_t$  を出力するものである。入力と出力は微分や箱ひげ図といった各スキルの問題集合に対する解答予測を

表現している。すなわち、あるスキルに関する問題の解答の結果を入力とすると、出力は各スキルに対する次問題の解答の予測である。

したがって、現状は学習者の個々の問題に対する解答を予測するわけではない。スキルごとに対する解答を予測することで、それぞれの学習者の各スキルに対する習熟度や関係性を考慮して学習の推薦に役立てるとというのが一般的利用方法である。モデルによる分野ごとの解答の予測がそのまま学習者の各分野の習熟度であると言い換えることができる。

このモデルでは大まかな知識に関する習熟度や知識全体における分野同士の関係性は考慮されるものの、問題一つ一つに対する違いは軽視されることとなる。同じ分野の問題でも平易な問題もあれば難解な問題もある。そのため、分野ごとの予測では予測精度に限界があるかもしれない。問題ごとの予測にすれば、同じ分野でも多くの生徒にとって難しかったかどうかを区別することができ、より理想的な正誤予測が実現する。

DKT の従来の方法では個々の問題の正答確率ではなく、問題カテゴリの正答確率なため、個々の問題の差異までは考慮できていなかった。一つの問題を一つの問題カテゴリに割り当てることも可能であるが、カテゴリの数が増えることで予測精度の低下を招く。その予測精度の低下を抑えるために、パラメータを変更し、疎（スパース）なデータでも有効なパラメータを検証する。

### 2.3 本研究でのアプローチ

そこで本研究では DKT の手法を利用して入力するパラメータ数を増やす。今回利用した ASSISTments にある全てのパラメータに関しては表 1 にまとめる。

現時点での DKT で利用されているパラメータは以下の 3 つである。

- 学習者の ID
- 学習者の取り組んだ問題カテゴリの ID
- 学習者が取り組んだ問題の正誤

ここに以下のパラメータを追加する。

- 学習者が取り組んだ個々の問題の ID
  - 学習者がその問題を解くのが何回目であるかという試行回数

問題カテゴリの ID と個々の問題の ID あるいは問題の試行回数と個々の問題 ID を併用することによって、学習者が次に解く問題の正答確率の予測精度の向上を目指す。

問題カテゴリに関しては現時点で問題カテゴリの情報を学習したのち問題カテゴリの正誤予測をした時、個々の問題の情報を学習して個々の問題の正誤を予測するよりも高い精度があると考えられる。そのため、その二つの併用によって個々の問題予測に関してもより高い精度での正誤予測ができる可能性がある。

問題の試行回数をパラメータとして加える理由に関しては、例えば学習者が自分の苦手な問題を解く場合はその問題に何度も取り組み、得意な問題なら何度も解いていないということが考えられる。すなわち問題試行回数が多ければ誤答しやすく、少なければ正答しやすいといった傾向が見られる可能性がある。そのためその情報を問題ごとの正誤予測に加えることによって正誤予測の精度向上を期待する。

上記の提案手法の取り組むために、既存の DKT のプログラムとオンライン学習サービス ASSISTment が提供するオープンな学習データを用いた。ここで用いたパラメータは以下の 4 つである。

生徒 ID は学習者ごとに割り振られた ID である。問題 ID は問題ごとに割り振られた ID である。スキル ID は各問題が属する微分や積分、確率といった知識のカテゴリの ID である。問題の正誤は生徒が解答した問題が正答であったか誤答であったかを表す。<sup>1</sup>

以下にはそれぞれの例を示す。

生徒 ID = 1

問題 ID = 10

スキル ID = 7

問題の正誤 = 1

---

<sup>1</sup> ASSISTment

(<https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010>)

上記のような数値なら 1 番目の生徒が 7 番のスキルに属する問題 10 を正解したといった（仮に問題の正誤が 0 ならば誤答であることを意味する）情報が得られる。

表 2. エンコーディングに利用した特徴ベクトルのパラメータ

	組み合わせ 1	組み合わせ 2
パラメータの種類	生徒ID	生徒ID
	問題ID	問題ID
	スキルID	問題試行回数
	正誤	正誤

## 第3章 パラメータの同定

オンライン学習サービス ASSISTments のデータには表 1 で示す 9 つパラメータが含まれている。

(教師の ID など明確に問題の正誤予測と明らかに関係のない要素はあらかじめ省いている) ここで精度向上のために生徒 ID・問題 ID・正誤に加えるものとして利用可能だと考えられるのは以下の二つと考えた。

- スキル ID
- 問題の試行回数

上記の要素を加えた理由について示す。スキル ID は「生徒 ID・問題 ID・正誤」を入力とした正誤予測精度よりも「生徒 ID・スキル ID・正誤」の方が高かったため、これを「生徒 ID・問題 ID・正誤」の特徴ベクトルに加えることによって精度が向上すると考えた。

問題の試行回数は生徒が苦手な問題に取り組むのは試行回数が多く、得意な問題は試行回数が少ないといった性質があると考えた。そのため苦手な問題ならば間違いやすく、得意な問題は正解しやすいとすると、この情報を加えることで正誤予測の精度を上昇させる可能性があるかとみている。

上記の二つに以外の要素を排除した理由について述べる。

「ヒントを求めた回数」は生徒が問題を解く段階では分からず、有用ではない。というのも生徒が問題を解いたときどれほどの確率で正解することができるのかという情報は生徒が問題に取り組む以前に必要なからだ。例えば、「比較的簡単な問題を解き続ける方が学習におけるモチベーションを維持しやすい」という属性の生徒がいた場合にその生徒が 8 割の確率で正解できる問題を提供するといったことがある。

しかし、「ヒントを求めた回数」というパラメータを予測に加えると実際に生徒がヒントを求めた瞬間、あるいはヒントを求めずに問題に解答した段階になってからでないとその情報を得ることができないので例に挙げたような問題推薦の役に立たないのである。

なので「最初の行動が問題を解いたかヒントに行ったか」、というのも同様の理由でパラメータとしては利用できず、「また問題を解くのにかった時間」も同様の理由で適していないといえる。

「スキル名」に関しては数値が入力の前提である DKT のプログラムに対して不適切である。

こうして選択した 2 つパラメータを何らかの形で DKT の入力として与え、学習者の問題の正誤予測の精度向上を目指す。  
パラメータを追加した上でのプログラムへの入力のエンコーディング方法に関しては第 4 章で説明する。

## 第4章 学習データのエンコーディング手法

DKTにおける既存のプログラムに対して、パラメータをベクトルとして符号化する仕組みについての説明をする。図2はDKTのプログラムにおいて入力されるベクトルの配列である。図2のように配列は1が一つと他は全て0で構成されるワンホットベクトルである。スキル数を  $\text{num\_skill}$  個、 $\text{skill}$  を0から  $\text{num\_skill}-1$  までの変数と仮定すると、配列は以下のように表現される。

- 学習者が  $\text{skill}$  番目の問題を誤答した時は、配列の  $\text{skill}*2$  番目が1
- 学習者が  $\text{skill}$  番目の問題を正答した時は、配列の  $\text{skill}*2+1$  番目が1

図1の場合なら学習者は1番目のスキルの問題を正答したということの意味している。

パラメータを1つ追加するにあたって、既存DKTにおける入力ベクトルの形式を考慮した手法を提案し、図2に概要を示す。図3はDKTの入力ベクトルの配列を参考にして、同じ形式の配列を全問題の種類の数 $\times$ 2の大きさの入力ベクトルである仮定して既存の配列に続くように追加した。これはすなわちパラメータの入力によって「スキルの種類の数 $\times$ 2+問題の種類の数 $\times$ 2」という配列の大きさになっている。配列全体で見ると2つの1と他は全て0で構成されている。スキル数を  $\text{num\_skill}$  個、問題数を、 $\text{num\_problem}$  個、 $\text{skill}$  を0から  $\text{num\_skill}-1$  までの変数、 $\text{problem}$  を0から  $\text{num\_problem}-1$  までの変数と仮定すると、配列は以下のように表現される。

- 学習者が  $\text{problem}$  番目の問題に誤答し、その問題が属するスキルが  $\text{skill}$  番目ならば、配列の  $\text{skill}*2$  番目と  $\text{num\_skills}*2 + \text{problem} *2$  番目が1

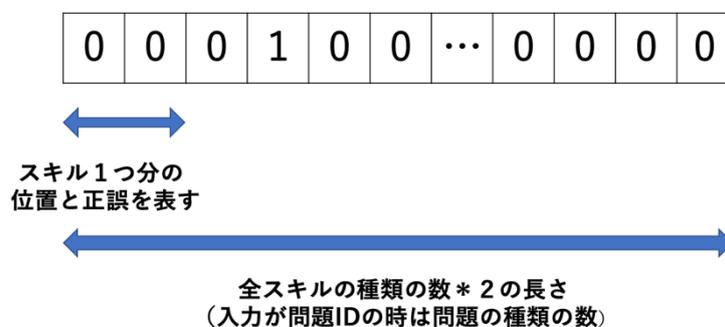


図 2.入力ベクトルとして利用される配列

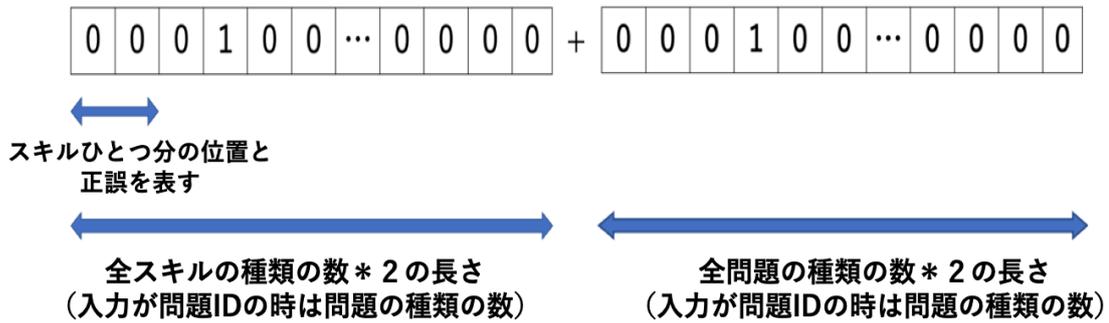


図 3.パラメータを追加した配列

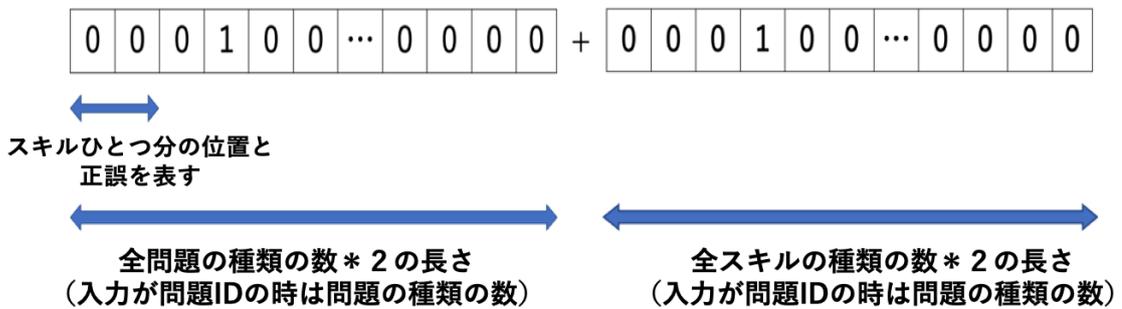


図 4.図 3 のスキルと問題の入れ替え

- 学習者が problem 番目の問題に誤答し、その問題が属するスキルが skill 番目ならば、配列の  $skill*2+1$  番目と  $num\_skill*2+problem*2+1$  番目が 1

これにより学習者が解いた問題とその問題のスキルを考慮した上で正誤の情報の入力が可能になると考えられる。

これをエンコーディング 1 とする。

図 3 の場合なら学習者は 1 番目の問題に正答し、その問題の属するスキルは 1 番目ということになる。

もう一つのエンコーディングの方法として、スキルと問題の入力を逆にして検証した。これについての配列の形は図 4 に示しており、これをエンコーディング 2 とする。

また、問題の予測に対してスキル ID が貢献しているかどうかを検証する方法として学習者を減らして DKT に学習させるという方法も実験した。学習者を減らすことによって学習した問題数が減少する。すると、問題が正答したか否かの予測に問題の情報が一切ない状態で、スキルの情報のみで予測せざ

るを得ない。その条件下ならばスキル ID を加えることによる有効性を確かめることができると考えられる。

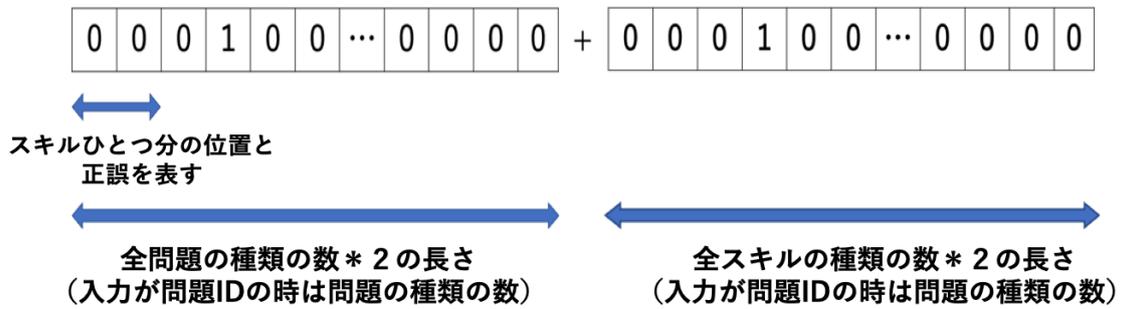


図 5.図 2 のスキルと問題の入れ替え

## 第5章 評価

既存の DKT と条件を統一するため ASSISTments のデータを利用し、提案した特徴ベクトルを検証する。なおデータの利用条件として「データのスキル ID が欠落している場合は学習するためのデータとして考慮しない」というものを加えている。

これは ASSISTments のデータにおいて生徒 ID、問題 ID 正誤の情報は全て揃っているが、スキル ID は空白になっている場合があり、それを取り除くためである。スキル ID を考慮して正誤を予測する際にその情報が欠落しているデータがあると、空白のデータが入力として使われることとなり正しい結果が得られない危険性があるためである。

まずは既存の DKT の条件通りにプログラムを実行する。すなわちスキルごとに正誤データを学習させ、そこからスキルごとの正誤予測の精度を出力させる。

これをパターン A とする。

次に同じ既存のプログラムを用いて入力をスキル ID から問題 ID に変更する。こうすることによって一つ一つの問題ごとに正誤データを学習させ、問題ごとの正誤予測の精度を出力させる。なお、あくまでもこれは既存のプログラムによるものであり、ここから精度がどれほど変化するかという検証に必要な比較対象である。

これをパターン B とする。

そしてパラメータとしてスキル ID を加えた 4 つ組によって問題の正誤を予測させる。

この四つのパラメータに関して第 4 章で述べたエンコーディングの方法 2 つそれぞれでの精度について検証した。

エンコーディング 1 に関しては、パターン C、エンコーディング 2 に関してはパターン D とする。

また、スキル ID ではなく、問題の試行回数を代わりに取り入れたパターンも検証する。

これに関しては第 4 章のエンコーディング 1 で検証している。

これをパターン E とする。

これらの特徴ベクトルのパターンに関しては表 3 に示す。

そして、そのパターンに基づく問題の正誤予測の比較に関しては表4にまとめる。

なお、全ての問題数、全てのスキルの数、学習に使われる生徒の数、学習した後にテストをする生徒数、問題の試行回数のパターンの数は表5で示している。問題の試行回数のパターンとは、問題試行回数そのものではなく、例えばデータの中に問題の試行回数が1回、2回、3回、4回というものだけならば、そのパターンの数は4つといえる。

表 3.特徴ベクトルのパターン

	入力するパラメータ			
パターンA	生徒ID	スキルID	正誤	
パターンB	生徒ID	問題ID	正誤	
パターンC (エンコーディング1)	生徒ID	問題ID	スキルID	正誤
パターンD (エンコーディング2)	生徒ID	問題ID	スキルID	正誤
パターンE (エンコーディング1)	生徒ID	問題ID	問題の試行回数	正誤

表 4.正誤予測精度の比較

	それぞれの精度
パターンA (生徒ID+スキルID+正誤)	73.2652%
パターンB (生徒ID+問題ID+正誤)	69.4113%
パターンC (生徒ID+問題ID+スキルID+正誤) エンコーディング1	69.6692%
パターンD (生徒ID+問題ID+スキルID+正誤) エンコーディング2	69.5776%
パターンE (生徒ID+問題ID+問題の試行回数+正誤)	70.3664%

表 5.Assistments のデータ利用について

問題数	17752
スキル数	123
学習に使われる生徒の数	3361
テストに使われる生徒の数	802
問題の試行回数のパターン	180

表 6.学習者を減らした場合の正誤予測精度比較(スキル ID)

	3つ組	4つ組	テストした生徒の問題数と学習していない問題数
50人の時	51.0916%	51.7815%	12810問中11805問が未知
100人の時	51.7253%	52.2479%	12810問中11550問が未知
500人の時	56.1458%	56.5274%	12810問中7905問が未知
1500人の時	68.6386%	69.6317%	12810問中2725問が未知

表 7.学習者を減らした場合の正誤予測精度比較(問題試行回数)

	3つ組	4つ組	テストした生徒の問題数と学習していない問題数
50人の時	51.0916%	51.6380%	12810問中11805問が未知
100人の時	51.7253%	52.2322%	12810問中11550問が未知
500人の時	56.1458%	56.4857%	12810問中7905問が未知
1500人の時	68.6386%	70.2163%	12810問中2725問が未知

次に、プログラムに入力する学習者を減らした際の「生徒 ID+問題 ID+正誤」の3つ組と「生徒 ID+問題 ID+スキル ID+正誤」の4つ組（エンコーディング1を用いる）について正誤予測精度の比較を表で示す。

これは学習者を減らすことによって学習する問題数を減らすことを目的としている。そのため学習者の人数だけではなく、どれほどの問題数を学習せず、正誤予測をしているか、すなわちスキル ID だけで予測しているかに関しても表6に示す。表6においては問題を学習せずにテストする生徒の正誤を予測している問題数を未知と記す。

そして、プログラムに入力する学習者を減らした際の「生徒 ID+問題 ID+正誤」の3つ組と「生徒 ID+問題 ID+問題の試行回数+正誤」の4つ組（エンコーディング1を用いる）について正誤予測精度の比較を表7で示す。こ

れも表 3 の場合と同様に学習者を減らすことで、学習する問題数を減らすことを目的としている。

すなわち、データがよりスパースになった時に問題試行回数が頑健であるか（精度を高い状態でキープできるかどうか）を検証する。

## 第6章 考察

まず第5章の一つ目の表における正誤予測精度の比較に関する結果について注目する。パラメータを増やすことによる精度の向上を期待した実験だったが、最も学習精度が高くなっているのは既存のDKTに基づく手法であるスキルごとの学習者の正誤予測である。同じ手法を用いて問題ごとの正誤予測をした場合は、当初の予想通り精度が低下した。この原因はスキルの数よりも100倍以上ある問題数の正誤予測の方が困難であると考えられるからである。そのため、スキル数も同時に考慮することで問題が属するカテゴリの情報に「問題ごとの細かい情報が付与された」と仮定すると、スキルごとの正誤予測よりは精度が上昇すると考えられる。しかしながら、実際の結果は異なっており、その原因を考察する。

まず、「生徒ID+問題ID+正誤」の3つ組と「生徒ID+問題ID+スキルID+正誤」の4つ組の比較では0.2%程ではあるが、精度の上昇が見られるので、問題のみで予測するよりは精度向上に有意だとはいえなくとも少しは貢献しているといえるだろう。しかし、スキルのみの精度予測には3%と大きく下回っているように問題IDを入力に加えると精度は低くなっているようだ。なので、入力するデータの種類が多くなると高い精度を保つことは困難になり、少ない方がより精度が高くなると考えられる。

この仮説を検証するために、学習者を減らした場合の予測精度についても注目する。ここでは、学習者を減らしたことで学習データが少なくなり、問題ごとの予測をスキルIDの入力からでしか予測することができなくなっている。それにもかかわらず、学習者を減らした際の「生徒ID+問題ID+正誤」の3つ組と比較して「生徒ID+問題ID+スキルID+正誤」の4つ組の精度全体として0.5%から1.0%程高くなっている。これは全ての学習者を減らす前の0.2%よりも大きな精度の向上である。よって先ほど提唱したスキルIDによってのみ問題の正誤の予測精度が上がるのではないかという説がより強固なものとなる。

さらに、問題の試行回数をスキルの代わりに利用した場合にも注目したい。「生徒ID+問題ID+正誤」の3つ組と比較すると1%ほどの精度の向上が見られた。スキルに関しては123種類あるが、これはある程度その解いた問題は分散していると考えられる。しかし、問題の試行回数に関しては1回

や2回という部分に集中していて、例えば「50回」「100回」「150回」解いた生徒が大量にいるということはかなり考えにくい。なので、追加で入力するデータとして種類が少ない方が精度は高くなると考えられるかもしれない。

エンコーディングの方法による差だが、単純に入力ベクトルのスキルの情報を先にするか問題を先にするかという違いであったが、予測精度の差は0.1%未満であり、ここで用いられている他の検証の精度の差よりも小さくほとんど関係していないと考えられる。

## 第7章 おわりに

本研究において、最も望ましい結果なのはデータの入力方法の違いや、パラメータを追加することによって生徒が解く問題の予測精度の向上という点であったが、期待したほどの精度の向上が見られなかった。

しかしながら、DKTのニューラルネットワークを変えずとも、精度を向上させるために、入力パラメータの数を増やしたり、エンコーディングの方法を変えたりすることは、精度の向上に寄与すると考える。期待したほどの数値の向上はなかったものの、スキルの情報や問題の試行回数情報を加えたものでは、学習に使う生徒の人数を変えて実験した時、いずれもその情報がないものと比較すると精度は上回っていた。

今後の検証しきれなかった方法として考えられる別のものはパラメータの入力としてスキルを入力し、問題の精度を予測するというものである。現時点でスキルからスキルを予測するものが最も高い精度を出すことができている。そのため、スキルから問題の正誤を予測するものがどれだけの精度があるかは試す価値がある。

あるいは、入力する情報としてパラメータを5つにするという方法も考えられる。単純に入力する情報を増やせば予測精度が上がるのは当然だと思われるが、それで本当に精度が向上するかどうかは検証してもいいかもしれない。

他の検証方法としてエンコーディングに関して別の手法について考える。今回の検証では既存のプログラムに従って単純に同じエンコーディングの方法で追加するパラメータの情報を加え、そのどちらを先に入力するかを入れ替えただけであった。

新たに考えられる手法としては、例えば入力するパラメータが「生徒ID+スキルID+問題ID+正誤」だった時に、スキルIDの個数\*問題ID\*2の配列を用意し、生徒が解いた問題とスキル、正誤の情報をワンホットベクトルとして表現する方法である。

今回はこの方法で入力を試した時、配列のサイズが大きくなりすぎたことが原因で実行に失敗してしまった。負荷を減らすような工夫かより強力なサーバがあれば実行可能であると思われる。

もう一つの方法としては，生徒と問題，正誤の情報を「問題の大きさのベクトル+1」で表現する方法である．問題の大きさのベクトルの部分で生徒がどの問題を解いたのかをワンホットベクトルで表現し，+1の部分で正誤の情報を入力する．今回はそこまで検証しきれなかった．

## 謝辞

本研究を行うにあたって熱心にご指導，ご鞭撻を賜りました村上陽平准教授に心より感謝致します。おかげさまで研究の楽しさを知ることができ，この論文の完成によって自らの成長が実感できました。また，日頃から切磋琢磨し，共に研究に励んだ村上研究室の皆様に深く感謝致します。

## 参考文献

- [1] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., and Sohl-Dickstein, J.: Deep Knowledge Tracing, in *Advances in Neural Information Processing Systems*, pp. 505–513 (2015).
- [2] 中川大海: 深層学習を用いた知識獲得予測を最適化する知識分類の抽出, 人工知能全国大会論文集 (2017).
- [3] 浅井孝太: Knowledge TracingとIRTの組み合わせによる大量試験結果データからの学生モデリング手法の提案, 情報処理学会第78回全国大会講演論文集 (2016).
- [4] 堤瑛美子: Bayesian Knowledge Tracingの一般化としての隠れマルコフIRTモデル, 人工知能学会全国大会論文集 (2019) .
- [5] Deep Knowledge Tracingの拡張による擬似知識タグの生成, 人工知能学会論文誌, 33巻3号 (2018).