

平成28年度

卒業論文

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

平成29年3月

指導教員 松尾豊 特任准教授

東京大学工学部システム創成学科知能社会システムコース

03-150984 中川大海

概要

近年、教育と情報技術の融合が進む中で、アダプティブラーニングという言葉が注目されている。個人に最適化された学習内容の自動提供を実現するもので、その社会的影響の大きさからアメリカを中心として世界的に注目が集まっており、関連するスタートアップや大学での研究に、多額の資金が投入されている。学習内容を個人に最適化するという考え方自体は、決して新しいものではないが、一人の教師が複数の生徒に対して同時に教育する形態の現在の学校教育では、全ての生徒に対して最適な学習内容を提供するには、障壁があった。

こうした問題を、情報技術の活用によって解決しようとする動きの一つとして、アダプティブラーニングが浸透し始めているが、その原動力となっているのが、オンライン教育サービスの普及である。オンライン教育サービスは、サービスを利用する生徒の学習行動ログを収集することで、これまで困難であった大規模な学習効果分析を可能にしたことにより加え、オンライン上の学習コンテンツを生徒が個人で利用するという形態を活用し、研究成果を元に学習コンテンツを個人に最適化して提供することを容易にした。

一方、近年、教育に限らない多くの研究領域で、深層学習が注目されている。従来の機械学習では、人間が問題の特徴を捉えて素性を設計する必要があったが、深層学習では、目的に応じた素性を、データから自動で学習することが可能になった。既存の機械学習の手法を上回る性能を得られることに加え、人間が認識できないような、データの複雑な特徴を捉えることが可能になったため、これまで人間が作り上げてきた概念を、大きく塗り替える可能性を秘めている。こうした深層学習の技術は、オンライン学習サービスに蓄積された大規模なデータを用いる学習効果の分析にも活用が期待されており、生徒の知識状態を正しく把握することで、最適な学習内容を特定することを目的とする知識獲得予測の研究に、深層学習を適用した例も報告されている。

しかし、こうした知識獲得予測の研究においては、予測アルゴリズムの部分に深層学習を適用しているものの、素性となる「知識」は、事前に人間が作成した知識分類によって定義されており、人間が設計した素性を利用する旧来の状況から脱していないのが現状である。データから特徴を自動で学習できる深層学習を活用すれば、人間が認識できないよ

うな、知識獲得の複雑な過程を反映した知識分類を学習できる可能性は高く、生徒の学習効率を最適化するという最終的な目標を、真に達成するには、知識分類自体も深層学習によって最適化される必要があるといえる。

本研究では、現在の知識獲得予測に用いられている、人間が作成した知識分類は、人間の複雑な知識獲得過程を表現する上では最適化された表現ではない、という仮定に立ち、知識獲得予測を行う上で最適な知識分類を、深層学習に自動的に学習させることを目的とする。実験の結果、深層学習が学習した知識分類を用いることで、人間が作成した既存の知識分類を用いる場合よりも、高い精度で知識獲得を予測できることが検証され、深層学習によって、知識獲得の予測性において最適化された知識分類を抽出できることが示された。この結果は、人間が認識しきれない、人間の知識獲得の複雑な過程を説明する表現を、深層学習が獲得したことを探しておらず、抽出された知識分類を活用することで、より最適化された学習内容を生徒に提供し、生徒の学習効率を高めることができると可能性を示唆している。さらに、研究の拡張として、本研究で用いた分析手法の教育医学における適用の可能性や、教育学以外の分野への応用の可能性を考察した。

本研究は、教育と情報技術の融合の進展やオンライン教育サービスの普及、教育分野における大規模分析の活発化や深層学習の躍進など、ここ数年の多様な領域の進展によって初めて可能になったものである。本研究が、既存の学問体系の再構築、そして人間の学習や知識の解明につながると信じている。

目 次

| | |
|--|----------|
| 第1章 序論 | 1 |
| 1.1 研究の背景 | 1 |
| 1.1.1 教育の個人最適化と学習科学 | 1 |
| 1.1.2 オンライン教育サービスの普及と学習効果分析の発展 | 2 |
| 1.1.3 深層学習の躍進と知識獲得予測 | 3 |
| 1.1.4 知識獲得予測の問題点 | 4 |
| 1.2 研究の目的 | 5 |
| 1.3 本論文の構成 | 6 |
| 第2章 関連研究 | 7 |
| 2.1 学習科学と情報技術 | 7 |
| 2.2 オンライン教育サービスと大規模な学習効果分析 | 9 |
| 2.2.1 MOOCs と ITS | 9 |
| 2.2.2 学習行動ログの蓄積と大規模分析の活発化 | 11 |
| 2.2.3 実証性の高いプラットフォームとしての性質 | 12 |
| 2.3 深層学習 | 12 |
| 2.3.1 ニューラルネットワーク | 13 |
| 2.3.2 深層学習の概要 | 15 |
| 2.3.3 Recurrent Neural Networks | 16 |
| 2.4 知識獲得の予測 | 20 |
| 2.4.1 Knowledge Tracing の定式化 | 21 |
| 2.4.2 Bayesian Knowledge Tracing | 22 |
| 2.4.3 Performance Factor Analysis | 23 |
| 2.4.4 Deep Knowledge Tracing | 24 |
| 2.4.5 知識獲得予測の手法における DKT の最適性 | 26 |
| 2.5 次元削減手法 | 28 |
| 2.5.1 Principal Component Analysis | 29 |

| | | |
|------------|-----------------------------|-----------|
| 2.5.2 | Autoencoder | 30 |
| 2.5.3 | Embedding | 31 |
| 2.6 | 用語の定義 | 31 |
| 2.6.1 | 知識獲得予測と回答正誤予測 | 31 |
| 2.6.2 | 知識分類と知識タグ | 31 |
| 第3章 | 提案手法 | 33 |
| 3.1 | 分析手法全体の流れ | 33 |
| 3.2 | データセットの作成 | 34 |
| 3.3 | 知識分類の学習と抽出 | 35 |
| 3.3.1 | DKT の拡張による写像関数の学習 | 36 |
| 3.3.2 | 写像関数の離散化による知識分類の作成 | 40 |
| 3.4 | 知識分類の予測性能の検証 | 40 |
| 3.5 | 知識分類の性質の比較 | 41 |
| 第4章 | データセット | 43 |
| 4.1 | ASSISTments 2009-2010 | 43 |
| 4.1.1 | ASSISTments のサービス | 43 |
| 4.1.2 | 対象データセット | 44 |
| 4.1.3 | データの抽出 | 44 |
| 4.2 | Bridge to Algebra 2006-2007 | 45 |
| 4.2.1 | KDDCup | 46 |
| 4.2.2 | 対象データセット | 46 |
| 4.2.3 | データの抽出 | 47 |
| 4.3 | データセットの概観 | 48 |
| 第5章 | 実験 | 49 |
| 5.1 | 実験設定 | 49 |
| 5.1.1 | 知識分類の学習と抽出 | 49 |
| 5.1.2 | 知識分類の予測性能の検証 | 50 |
| 5.1.3 | 知識分類の性質の比較 | 51 |
| 5.2 | 実験結果 | 52 |
| 5.2.1 | 知識分類の予測性能の比較 | 52 |

| | |
|-----------------------------|-----------|
| 5.2.2 抽出タグと既存タグの関係の概観 | 53 |
| 5.2.3 抽出タグと既存タグの比較分析 | 54 |
| 第6章 考察 | 57 |
| 6.1 本手法の有効性と得られた知識分類の解釈 | 57 |
| 6.1.1 知識分類学習モデルの有効性 | 57 |
| 6.1.2 各知識分類の性質と知識獲得予測に与える影響 | 58 |
| 6.2 本手法の汎用性と実用性 | 59 |
| 6.2.1 学習された知識分類の汎用性 | 59 |
| 6.2.2 本手法の他データへの適用可能性 | 60 |
| 6.2.3 教育現場への適用 | 61 |
| 6.3 今後の展望 | 62 |
| 6.3.1 知識分類学習モデルの改善 | 62 |
| 6.3.2 学習科学における対象データの拡大 | 63 |
| 6.3.3 学習科学以外の分野への応用 | 64 |
| 第7章 結論 | 65 |
| 参考文献 | 66 |
| 謝辞 | 93 |

図 目 次

| | | |
|-----|------------------------|----|
| 1.1 | 既存研究と本研究の差分のイメージ | 5 |
| 2.1 | Coursera のイメージ | 10 |
| 2.2 | JMOOC のイメージ | 10 |
| 2.3 | Knewton のイメージ | 11 |
| 2.4 | 各ニューロンの仕組み | 13 |
| 2.5 | 単純パーセプトロンの構造 | 14 |
| 2.6 | 多層パーセプトロンの構造 | 14 |
| 2.7 | RNN の基本構造 | 17 |
| 2.8 | RNN の基本構造(展開) | 17 |
| 3.1 | 分析手法全体の流れ | |
| | 網掛け部は提案手法の比較対象となる手法を表す | 34 |
| 3.2 | モデル構造上の拡張 | 37 |
| 3.3 | ネットワーク作成の流れ | 42 |
| 4.1 | 問題ごとの回答数の分布 | 46 |
| 5.1 | 既存タグネットワークの構造 | 53 |
| 5.2 | タグ関係ネットワークの構造 | 54 |
| 5.3 | 各タグの出現回数の分布 | 55 |
| 5.4 | 多くの抽出タグが紐づく既存タグ | 55 |
| 5.5 | 少数の抽出タグのみ紐づく既存タグ | 55 |
| 5.6 | 内容的関係性の強い既存タグに紐づく抽出タグ | 56 |

表 目 次

| | |
|---|----|
| 2.1 Deep Knowledge Tracing における回答ログデータと対応する入力ベクトル の例 | 25 |
| 4.1 各データセットの統計量 | 48 |
| 5.1 各知識タグの知識獲得予測における予測性能 | 52 |
| 1 「ASSISTments 2009-2010」 | 75 |
| 2 「Bridge to Algebra 2006-2007」 | 82 |

第1章 序論

本章では、本論文の背景、研究目的および本論文の構成について述べる。

1.1 研究の背景

1.1.1 教育の個人最適化と学習科学

近年、教育と情報技術の融合が進む中で、アダプティブラーニングという言葉が注目されている。個人に最適化された教育の自動提供を実現するもので、その社会的影響の大きさからアメリカを中心として世界的に注目が集まっており、関連するスタートアップや大学での研究に、多額の資金が投入されている [Piccioli, 2014].

教育を個人に最適化するという考え方は、決して新しいものではない。現在の学校教育では、一人の教師が複数の生徒に対して同時に教育する形態が一般的であるが、学習の速度や教科による得手不得手は人それぞれであり、同じ教育を施しても、十分な理解ができずにつまずいてしまう生徒もいる。つまずいている生徒に補習を行い、つまずきを克服する手助けをするような、生徒の学習状態を考慮して教育を設計することは昔から行われてきており、こうした指導が巧みな教師は「腕のいい」教師として評価されてきた。しかし、こうした方法では、習熟の遅い生徒を助けることに重きが置かれるため、習熟が周りより早い生徒への対応は後回しにされることが多く、発展的な学習機会や知的好奇心の向上を妨げることに加え、現実的な時間と労力を考慮すると、全ての生徒に個別に対応することは困難である。

より個別に教育を受ける手段として、個別指導形式の塾や家庭教師、通信教育なども利用される。生徒一人一人に教師がつくことで、生徒の習熟度合いを考慮して教育を設計でき、また能力によって優先されたり後回しにされたりすることもないため、学習内容を個人に最適化するという目的の上ではより望ましい。しかし、このように教育の粒度を細かくし、個人最適化を進めようとするほど、教師一人あたりが担当できる生徒の数が減ることによる人材的・金銭的負担や、教師ごとの指導能力の違いなどの問題に直面する。結局、

このような教師のマンパワーに依存した方法では、誰もが平等に最適な教育を受けるという目的を達成するには、障壁が残る。

すべての生徒に最適な教育を提供するには、現代の情報技術を活用して、生徒の学習過程を分析し、学習内容を決定するような、新たな教育システムが必要であり、このようなシステムを作り上げようとする動向は「学習科学」という研究分野として確立され、今日大きな注目を集めている [白水始 et al., 2014].

1.1.2 オンライン教育サービスの普及と学習効果分析の発展

教育と情報技術の融合の象徴とも言えるアダプティブラーニングの躍進には、オンライン教育サービスの普及が背景にある。オンライン教育サービスとは、学校の教室で一人の教師が複数の生徒に対して同時に授業を行う従来の教育形態と異なり、PC やモバイル端末を通じて、オンライン上で提供される学習コンテンツを、生徒が各自で利用するサービスを指す。

オンライン教育サービスの一つである Massive Open Online Courses(MOOCs) [McAuley et al., 2010, Pappano, 2012, Siemens, 2013] は、多様な分野や難易度の講義から、時間や場所を問わずに、生徒が自分のペースで学習したいものを選択して学習できるというもので、生徒が自身の習熟度合いに沿った教育を受けられないという問題を解決するプラットフォームとして、活用が期待されている。例えば、世界最大級の MOOCs の一つである Coursera¹は、2017 年 1 月の時点で、29 の国にまたがる 148 の教育機関とパートナーシップを結び、コンピュータサイエンス、数学や論理、社会科学などに関する 1600 以上の講座を、2200 万人以上に提供している。日本では 2013 年 2 月に東京大学が Coursera に、2013 年 5 月に京都大学が edX²に参加を表明したことから普及し、2013 年 11 月には日本版の MOOCs として JMOOC³が設立されるなど、国内外で MOOCs の利用が拡大している。

多様な講座を多くの人に提供する MOOCs 以外にも、より個人の学習過程をサポートすることに特化して設計される Intelligent Tutoring System(ITS) と呼ばれるオンライン自動学習支援システムの利用も拡大している。世界最大級の ITS である Knewton⁴では、生徒の学力や理解度と、学ぶべき対象をマッピングすることで、生徒に最適な学習過程を設計し、かつ生徒の学習の進捗に応じてその過程を動的に変化させる仕組みを有してい

¹<https://www.coursera.org/>

²<https://www.edx.org/>

³<https://www.jmooc.jp/>

⁴<https://www.knewton.com/>

る [Upbin, 2012]. 近年では ITS と MOOCs の融合も進んでおり [Aleven et al., 2015], オンライン教育サービスの利用は世界中で拡大している.

さらに, オンライン教育サービスは, 新たな学習形態を提供するのみにとどまらず, これまで困難であった大規模な学習効果分析を可能にするプラットフォームとして期待されている. オンライン教育サービスでは, 提供された講義を生徒が学習する際に, その学習行動ログをデータとして蓄積することが可能なため, そうして蓄積された多様な学習者の大規模な学習行動ログから, 多様な学習効果の分析が可能になった. 特に, 演習問題の回答ログは, その問題が問う知識を学習者が獲得しているか否かを表すため, Knowledge Tracing と呼ばれる知識獲得の分析に利用できる [Corbett and Anderson, 1994]. 例えば, 生徒の問題回答ログを利用して知識獲得の予測を行った研究 [MacHardy and Pardos, 2015] は, 世界的に有名な MOOCs の一つである Khan Academy⁵に蓄積された 100 万件以上の問題回答ログを使用しており, 教育の分野における大規模な学習効果分析の一つである. 生徒が個人で利用するというオンライン教育サービスならではの特性によって, このような研究成果を元に最適化された学習コンテンツを, 個人に提供することが容易になったため, 教育の個人最適化を進める動きは, 急速に活発化している.

1.1.3 深層学習の躍進と知識獲得予測

一方, 近年, 教育に限らない多くの研究領域で, 深層学習が注目されている. 従来の機械学習では, 人間が問題の特徴を捉えて素性を設計する必要があったが, 深層学習では, 目的に応じた素性を, データから自動で学習することが可能になり, 画像認識 [Schroff et al., 2015, Szegedy et al., 2014], 音声認識 [Hinton et al., 2012, Bahdanau et al., 2015], 機械翻訳 [Sutskever et al., 2014, Dong et al., 2015] 等, 多様な研究領域で飛躍的な進展が報告がされている. 直近の一年間では, 画像から動画を生成する研究 [Vondrick et al., 2016] や, 会話を人間と同程度に認識できるとする音声認識の研究 [Xiong et al., 2016a], 一部の欧米言語間の文レベルで, ほぼ人間と同等に正確な翻訳を実現したとする機械翻訳の研究 [Wu et al., 2016] なども報告されており, 深層学習によって, 日々驚異的な成果が生まれている. また, 2016 年 3 月に人間のプロを倒したことで一躍有名になった, Google Deep Mind が開発したコンピュータ囲碁プログラムの「AlphaGo」[Silver et al., 2016] は, 過去の人間の対局の記録である棋譜から, 深層学習によって人間の囲碁の打ち方を学習し, 自己対局による強化学習を通して, 今後 10 年は不可能と言われていた, 人間のプロを打

⁵<https://www.khanacademy.org/>

ち負かすほどの棋力を獲得した。AlphaGoは、単に人間の打ち方を真似ただけでなく、それまで人間が考えつかなかつたような斬新な手を学習しており、囲碁界に衝撃を与えてい る。このように、深層学習は、人間が認識できないようなデータの複雑な特徴を捉えるこ とで、これまで人間が作り上げてきた概念を、大きく塗り替える可能性を秘めている。

こうした深層学習の技術は、学習効果の分析にも活用が期待されている。特に、オンライン教育サービスの普及によって盛んになった知識獲得予測の研究も、深層学習の適用によつて大きく進展した。Piechらが発表した、知識獲得予測に深層学習を活用する Deep Knowledge Tracing(以下、DKT)という手法では、時系列分析でよく用いられる深層学習 モデルである Recurrent Neural Networks [Williams and Zipser, 1989] を活用することで、高い性能で知識獲得を予測できること、また、予測モデルを分析することで知識間の関係 性をネットワークとして抽出できることが報告されている [Piech et al., 2015]。

1.1.4 知識獲得予測の問題点

しかし、こうした知識獲得予測に深層学習を適用する DKT の研究においても、ある問 題が存在する。知識獲得を予測するアルゴリズムの部分には深層学習を適用しているもの の、素性となる「知識」は、事前に人間が作成した知識分類によって定義されており、人 間が設計した素性を利用するという旧来の状況から脱していない。人間が作成した知識分 類というのは、「知識の体系はこうなつてはいるはずだ」ないしは「この体系に基づいて教える・教えられることが望ましい」という専門家の仮説や理論に基づいて作られたものであ る。そのため、人間にとつての可読性は高いとしても、実際の生徒の学習過程を定量的に 分析した結果を反映し、知識獲得を予測する上で最適化された分類であるという根拠はな い。今日多様な分野で成果を生んでいる、データから特徴を自動で学習できる深層学習を 活用すれば、人間が認識できないような、人間の学習過程に潜む複雑な知識構造を反映し た、予測を最適化するような知識分類を学習できる可能性は高く、生徒の学習効率を最適 化するという最終的な目標を真に達成するには、知識分類自体も深層学習によって最適化 される必要があるといえる。

以上の問題意識に基づいた、既存研究との本研究の差分のイメージを図 1.1 に示す。

以上、本研究の背景について述べた。

次に、本研究の目的について説明する。

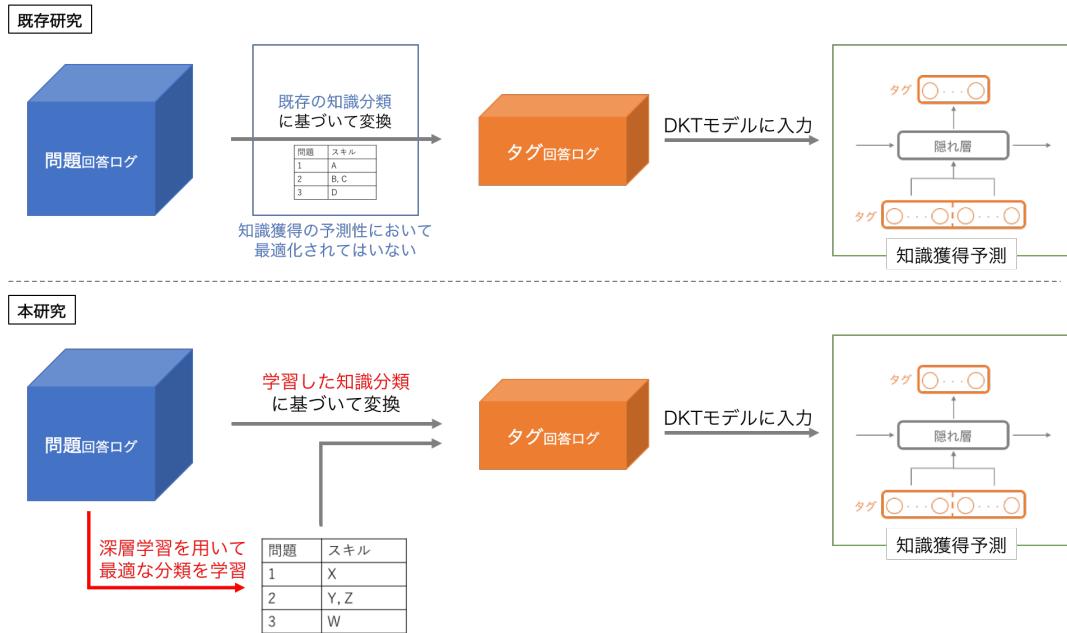


図 1.1: 既存研究と本研究の差分のイメージ

1.2 研究の目的

本研究では、現在の知識獲得予測に用いられている、人間が作成した知識分類は、人間の知識獲得の予測を行う上では最適化された表現ではない、という仮定に立ち、下記を検証することを目的とする。深層学習によって抽出した知識分類を用いることで、人間が作成した知識分類を用いる場合よりも、高い精度で知識獲得予測を行うことができる。

本論文では、人間が作成した知識分類を所与のものとせずに、問題の回答ログデータのみを深層学習に適用し、回答正誤予測を行う過程で、その予測性を最適化する知識分類を抽出することを目的としている。従来、専門家が事前に分類することが必要であった問題コンテンツの集合から、知識獲得の予測性を最適化する知識分類を、深層学習によって自動で抽出できることを明らかにすることは、既存の学問体系やカリキュラム設計の再構築につながるだけでなく、人間の知識状態の推移するメカニズムを解明することにもつながり、学術的な意義が大きいと考える。

以上、本研究の目的について述べた。

次に、背景と目的を踏まえて本論文の構成について述べる。

1.3 本論文の構成

以降の本論文の構成について述べる。

2章では、関連研究について述べる。知識獲得予測の関連研究を俯瞰し、周辺概念を整理することで、本研究の学術的位置づけを明確にする。

3章では、分析手法について述べる。まず、生徒の回答正誤を予測する過程で最適な知識分類を学習する手法について説明し、また、その結果から抽出された知識分類を分析する手法について説明する。

4章では、実験で用いるデータセットについて述べる。オンライン教育サービスにおける生徒の学習回答ログである、2つのデータセットを紹介し、本研究に適用するための事前の処理を説明する。

5章では、実験について述べる。本研究で行う実験を大きく3つに分け、各実験の設定を述べた後、実験結果について述べる。実験結果においては、提案手法によって抽出された知識分類を用いることで、既存の知識分類を用いた場合よりも高い精度で知識獲得を予測できることを示し、予測性を最適化する知識分類が抽出されたことを定量的に確認する。さらに、抽出された知識分類を既存の知識分類と定量的・定性的に比較することで、その性質を解釈する。

6章では、実験結果を踏まえた考察を述べる。まず、本研究で用いた手法や抽出された知識分類について、実験結果を踏まえて、その性質を考察し、実際の教育現場への適用法について述べる。また、本研究で用いた分析手法の、多様なデータへの適用可能性について議論し、教科によらず適用できる可能性があること、教科によって抽出される知識分類の性質が異なる可能性があること、一方で、複合的な学問や専門性の高い学問については、適用可能性の検証実験が必要であることを述べる。さらに、本研究で抽出された知識分類を離散表現に改めることの重要性や可能性について論じ、また、本研究で用いた分析手法が、教育学以外の分野にも応用できる可能性を持つことを論じる。

最後に、7章で結論を述べる。

以上、序論について述べた。次に、関連研究について述べる。

第2章 関連研究

本研究が問題提起を行った知識獲得予測の研究は、学習科学の発達や、教育に関する大規模分析の活発化、深層学習技術やその他の多様な分析技術の進展により発展してきた研究分野である。本章では、知識獲得予測の関連研究を俯瞰し、現状や周辺概念を整理することで、本研究の学術的位置づけを明確にする。

まず、人間の学習効果について研究する学習科学の歴史と情報技術の発達との関係性について整理した後、その情報技術の一つとして期待されているオンライン教育サービスについて、具体的な事例を挙げながら、その効果や関連する研究について述べる。次に、深層学習について概説し、本論文との関わりが深い Recurrent Neural Networks について詳細に述べる。さらに、知識獲得の予測手法である Knowledge Tracing について、その有益性や、伝統的な手法、深層学習を用いた最先端の手法について整理した後、本研究において既存手法を拡張する上で用いる、大規模データから次元削減を行う関連手法について整理する。最後に、以上の関連研究を踏まえて、本論文で使用する類似の用語について、定義を明確にする。

2.1 学習科学と情報技術

20世紀後半、人間の心の働きを理論化する認知科学が、現実社会で実際に役立つ科学として再構築される流れの中で、人を日常の学びの中で今より賢くするために実際に役立つ科学として「学習科学 (Learning Sciences)」の分野が確立された [白水始 et al., 2014]。学習科学は、従来の、実験室環境でのみ確認されるような非実用的な理論研究を避け、学習がうまくいく要因や状況を解明した上で、その学習を人間が積極的に引き起こすことを目指すような、実践の学を新たに打ち立てることを目指したものであった。明確な定義は様々であるが、[三宅なほみ et al., 2002] らは「よりよい教育を実現したいという社会的要請を背景にして、これまでの認知研究に基づき、現実の人の学習、例えば学校教育の中での子どもたちの学習を研究し、現代のテクノロジを駆使して実効性のある教育のシステムを教育実践の中で作り上げようという研究動向」と定義している。

学習科学の発展は、認知科学の進展だけでなく、情報技術の発達が大きな貢献を果たしている。学習科学は、人間の認知過程を解明する基礎研究としての性質に加え、実社会での有効性を検証する実証的な応用研究としての性質を兼ね備えているため、オンライン教育サービスのような教育と情報技術の融合によって、これまで実現しにくかった学習環境を作り検証できるようになったことは、学習科学の発展を大きく加速させた。

こうした情報技術との融合により実証研究が進み、今日注目されている教育システムの例として「アダプティブラーニング」が挙げられる [Carbonell, 1970, Midgley, 2014]。アダプティブラーニングは、個人に最適化された学習内容の自動提供を実現するもので、その社会的影響の大きさからアメリカを中心に注目が集まっており、関連するスタートアップや大学での研究に、多額の資金が投入されている [Piccioli, 2014]。

学習内容を個人に最適化させるという考え方自体は、学習科学研究の分野でも、また、研究という形に上がらないレベルでも、古くから存在し、例えば習熟の遅い生徒に教師が個別で補習に当たったり、個別指導塾や通信教育で生徒各自が自身の習熟度にあった講義を受けたりと、様々な形態を取って実践してきた。しかし、こうした従来の方法は、教育の粒度を細かくし、個人最適化を図ろうとするほど、教師一人あたりが担当できる生徒の数が減ることによる人材的・金銭的負担や、教師ごとの指導能力の違いなどの問題に直面し、誰もが平等に最適な教育を受けるという目的を達成するには、障壁が残っていた。

この事態を開いたのが、教育と情報技術の融合である。中でも、その象徴とも言えるオンライン教育サービスでは、サービスを利用する生徒の学習行動ログを収集することで、これまで困難であった大規模な学習効果分析を可能にしたことに加え、オンライン上の学習コンテンツを生徒が個人で利用するという形態を活かし、研究成果を元に学習コンテンツを個人に最適化して提供することを容易にした。

このように、基礎理論に加え実証性も重視する学習科学の領域は、情報技術との融合により大きく発達してきた。特に、オンライン教育サービスは、データの蓄積と研究成果の実証という二つの目的が達成できるプラットフォームとして、大きな注目を集めている。

以上、学習科学と情報技術の関係性について述べた。

次に、教育と情報技術の融合を象徴する例として、オンライン教育サービスについて述べる。

2.2 オンライン教育サービスと大規模な学習効果分析

オンライン教育サービスの代表的な例として Massive Open Online Courses(MOOCs)と Intelligent Tutoring System(ITS)を取り上げ、具体的な事例を挙げながら、関連する研究について述べる。また、こうしたオンライン教育サービスが大規模分析に活用されている状況や研究について整理する。

2.2.1 MOOCs と ITS

MOOCs は Massive Open Online Courses [McAuley et al., 2010, Pappano, 2012, Siemens, 2013] の略称で、特に日本語で表記する場合は大規模公開オンライン講座と記述することがある。MOOCs は、オンライン上で公開された、大学などの様々な教育機関の講座を、誰もが無償で受講でき、また修了時には修了証も取得できる教育サービスのことを指す。

学びたい人がいつでもどこでも学習リソースにアクセスできる、という MOOCs の概念自体は古くから提唱されていたが、実現化したのは、2008 年にカナダのマニトバ大学で学生向けのオンライン講座を開設した際に、25 人の受講者だけでなく 2000 人以上の人々がその講座に参加したことがきっかけだと言われている [Yuan et al., 2013]。

以前から、大学などの高等教育機関は、オープンコースウェア [Abelson, 2008] という形で講義の動画や資料を公開していたが、MOOCs は、参加人数が非常に大規模で、また、高等教育水準の内容だけでなく、初等中等教育水準の内容の講座も含まれている点で異なる。また、これまでオンラインの講座というものは存在していたが、MOOCs は、参加人数が非常に大規模である点や公開している講座の数が大規模である点、また、その内容が多様であるという点、利用が無料、あるいは無料に近いという点において、これまでのオンライン講座とは異なる。

MOOCs は、従来の、学校の教室で一斉授業形式で提供される教育形態と異なり、オンライン上の多様な講座に、生徒が個人でアクセスし、講座ごとに提供される講義の動画や演習システムなどを通じて、いつでもどこでも、自身の習熟度合いやペースに合わせて、自分の学習したいものを選択して学習できる。従来の教育の、生徒が自身の習熟度合いに見合った学習ができないという問題を解決するものとして注目されていることに加え、産業や社会への影響も注目されている。例えば、大学生だけでなく社会人も、自身の専門領域に関する講座を受講することでより理解を深めたり、あるいは専門領域とは異なる幅広い講座を受講することで、教養を養うことができる。また、公教育の整備が追いついてい

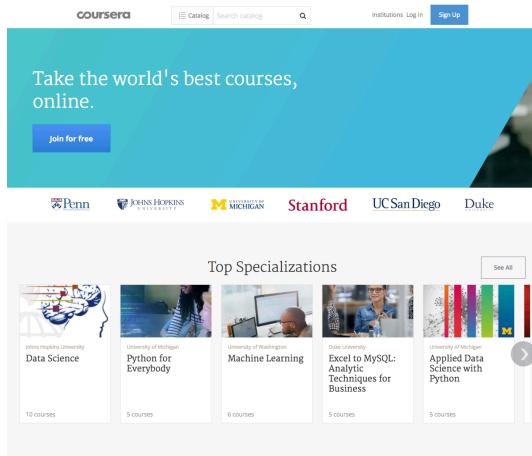


図 2.1: Coursera のイメージ



図 2.2: JMOOC のイメージ

ないような発展途上国においては、MOOCs が教育に与える影響は大きく、その影響や可能性を分析する報告は多い [Trucano et al., 2013, Liyanagunawardena et al., 2013].

このように、MOOCs は社会の多様な場面で、これまでにない学習機会を提供しており、教育や学習といったものの方に大きな影響を与えている。

MOOCs の有名な事例として、世界的に有名な Coursera や、日本発の MOOCs である JMOOC が挙げられる。Coursera と JMOOC のイメージを図 2.1, 2.2 に示す。Coursera は、2017 年 1 月の時点で、29 の国にまたがる 148 の教育機関とパートナーシップを結び、コンピュータサイエンス、数学や論理、社会科学などに関する 1600 以上の講座を、2200 万人以上に提供している¹。JMOOC は、2013 年 11 月に日本版の MOOCs として設立され、10 代から 80 代までと幅広い年代に、アートや医療、自然科学や資格試験対策などの講座を提供しており、2017 年 1 月の時点で、140 の講座を 50 万人以上が受講している²。

多様な講座を多くの人に提供する MOOCs 以外にも、より個人の学習過程をサポートすることを目的として設計された、Intelligent Tutoring System(ITS) と呼ばれるオンライン自動学習支援システムの利用も拡大している [Sleeman and Brown, 1982]。

ITS の有名な事例として、世界最大級の ITS である Knewton³のイメージを図 2.3 に示す。Knewton では、生徒の学力や理解度と、学ぶべき対象をマッピングすることで、生徒に最適な学習過程を設計し、かつ生徒の学習の進捗に応じてその過程を動的に変化させる仕組みを有している [Upbin, 2012]。

¹講座数と利用者数はトップページの記載より引用。

²講座数と利用者数はトップページの記載より引用。

³<https://www.knewton.com/>



図 2.3: Knewton のイメージ

また、近年では、これまで難しいと言われていた ITS の MOOCs への埋め込みを達成したとする研究 [Aleven et al., 2015] も報告されており、ITS が利用される場面は、今後より拡大していくといえる。

2.2.2 学習行動ログの蓄積と大規模分析の活発化

MOOCs や ITS などのオンライン教育サービスは、人々に新たな学習の機会を提供するという側面だけでなく、これまで困難であった大規模な学習効果分析の可能性を高めるという側面もある。

生徒はオンライン上で提供された講義動画や演習問題を通して学習するが、オンライン上で実施されているため、学習行動ログをデータとして蓄積することができ、蓄積されたデータを分析に活用することができる。多様な生徒が利用するため、多様な生徒の大規模な学習行動ログから多様な講座の学習効果の分析が可能となりつつある。

特に、演習問題の回答ログはその演習問題により評価される知識を生徒が獲得しているか否かを表現しているため、知識獲得の分析に利用できる。例えば、MOOCs の演習問題の回答ログを利用して知識獲得の予測を行う研究 [MacHardy and Pardos, 2015] では、世界的に有名な MOOCs である Khan Academy から収集したデータを利用していたが、その問題回答ログ数は 100 万件以上であり、これまでにないほど大規模なデータを対象に分析が実施されたといえる。

2.2.3 実証性の高いプラットフォームとしての性質

さらに、オンライン教育サービスが学習効果分析の価値を大きく高めている要因として、オンライン上のコンテンツを、多様な生徒が、個人で利用するというプラットフォームとしての性質がある。

現在の学校教育の形態では、生徒の学習効果に関する分析を行い、なんらかの知見を得たとしても、それを多様な生徒に適用して効果を検証したり、各個人に提供できるような環境が整備されておらず、学習効果分析が社会に与える影響が限定的であった。また、従来の一般的なeラーニングによる学習支援システムも、大学のような各教育機関が個別に設定し、学内の生徒が利用者の中心であったため、システムの利用者が限定されており、データの多様性や研究成果の活用可能性も狭い範囲に留まっていた。

一方、MOOCsやITSのような大規模なオンライン教育サービスは、教育機関の垣根にとらわれず、多様な背景、適性、能力を持つ生徒が利用していることに加え、学習コンテンツを個人が利用する形態のため、多様なデータを元に得られた一般性のある知見を、多様な生徒に対して、生徒個人の粒度で提供することが可能である。例えば、生徒の知識獲得予測の研究は、得られた成果から、より生徒の学習効率を高めたり、継続を推進するような教材推薦システムを開発し、実際のサービス上で個人個人に適用することで、効果を実証することができる。このような性質から、オンライン教育サービスのデータに基づいた学習効果分析が持つ社会的影響は、大きなものとなっている。

以上、オンライン教育サービスを取り巻く環境と、学習効果分析との関連性について述べた。

次に、深層学習について述べる。

2.3 深層学習

本研究で用いる技術の核となっている、深層学習について述べる。まず、深層学習の基礎となっているニューラルネットワークの概念について説明した後、深層学習の概要について述べ、さらに本研究で用いる深層学習モデルである Recurrent Neural Networks について詳述する。

2.3.1 ニューラルネットワーク

深層学習は、機械学習における一分野であり、その中でもニューラルネットワークという特殊なモデル構造を拡張したものである。よって、まずはニューラルネットワークについて説明する。

ニューラルネットワークは、機械学習におけるモデル構造の一つで、人間の脳の神経回路の仕組みを模したものである。人間の脳は、膨大な数のニューロンと呼ばれる神経細胞から構成され、各ニューロンは相互に連結し、巨大なネットワークを成している。外界からの情報によってあるニューロンが刺激を受けると、そのニューロンの電位は次第に上昇し、電位が一定の閾値を超えるとそのニューロンは発火し、接続している他のニューロンに情報の信号を出力することにより、情報の伝達が行われている。ニューラルネットワークのモデルでは、このニューロン一つ一つの情報伝達の仕組みと、それらが互いに接続してネットワークを成す構造をモデル化している。

各ニューロンは図 2.4 のようにモデル化される。

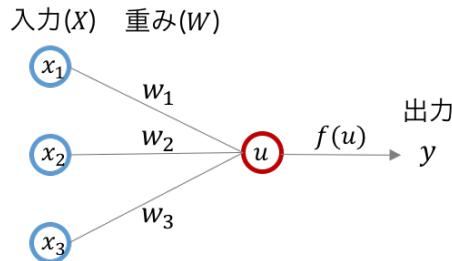


図 2.4: 各ニューロンの仕組み

各ニューロンは他のニューロンから入力信号 x を受け取るが、その信号の伝達効率は一様ではない。それぞれの入力には伝達効率として重み w が設定され、その重み付きの入力 wx が対象のニューロンに加算されていく。その総和 u は事前に定められた活性化関数 f に基づいて正規化され、出力される。活性化関数には様々な種類があり、式 2.1 で表されるような、閾値 θ を境に 0 か 1 を出力するような単純なステップ関数以外に、式 2.2, 2.3 で表されるようなシグモイド関数 (sigmoid), 双曲線正接関数 (\tanh) などの非線形関数も存在する。

$$\varphi(x) = f(x - \theta) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases} \quad (2.1)$$

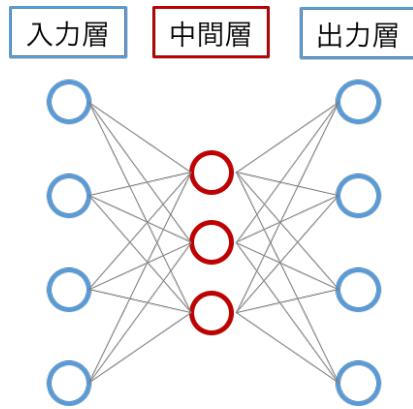


図 2.5: 単純パーセプトロンの構造

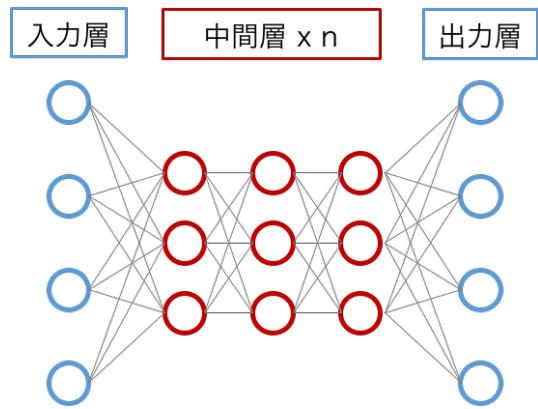


図 2.6: 多層パーセプトロンの構造

$$\varphi(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

$$\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

このようにモデル化された各ニューロンを、人間の脳の神経回路のように、互いに結合させてネットワーク化した例が図 2.5 である。これは 1958 年に Rosenblatt により提案された単純パーセプトロンというネットワークで、ニューラルネットワークの元祖とも言われる最も基本的なネットワークである [Rosenblatt, 1958]。それぞれのニューロン（以下、ユニット）は、各層の間で互いに全結合しており、前の層からの入力 \mathbf{x} に重み \mathbf{W} が掛け合わされ、バイアス項 \mathbf{b} を加算したものに活性化関数 f が適用され、出力される。層 i から層 j への出力の計算は以下の式に基づいて行われる。

$$\mathbf{y}_j = f(\mathbf{W}_{i,j} \mathbf{x}_i + \mathbf{b}_j) \quad (2.4)$$

なお、 \mathbf{x}_i は層 i における出力 ($i = 1$ の時はモデルへの入力) を指し、 $\mathbf{W}_{i,j}$ は重み行列を指し、 \mathbf{b}_j はバイアス項を指し、 f は活性化関数を指し、 \mathbf{y}_j は層 j における出力を指す。

この計算を層ごとに順次行い、最終的な出力が決定され、事前に設定した誤差関数によってモデルの予測誤差が算出される。ニューラルネットワークでは、この誤差関数を重みやバイアスのパラメータによって微分して負の勾配方向を見つけ、パラメータを勾配方

向に修正することを繰り返すことにより、最適なパラメータが得られるように学習を進める。重みの更新の基本的な仕組みは、以下の式によって表される。

$$\nabla E = \frac{\partial E}{\partial \mathbf{w}} = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right] \quad (2.5)$$

(2.6)

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \epsilon \nabla E \quad (2.7)$$

なお、 E は誤差関数であり、 $\mathbf{w}^{(t)}$ は時刻 t における重みベクトルであり、 ϵ は学習率と呼ばれる、1回の学習あたりのパラメータ更新量の大きさを決定する定数である。

このようにニューラルネットワークの基礎として考案された単純パーセプトロンだが、排他的論理和 (XOR) のような非線形問題を解けず、現実の複雑な問題には適用できないことが指摘された [Minsky and Papert, 1969]。また、ニューラルネットワークを多層にすることも早くから考案されていたが、極めて高い計算処理性能を要することが課題であり、長い間実用には堪えない時代が続いていた。このような歴史を経た後、近年の計算機の性能向上や、その他のモデル設計上の技術的進歩を背景に、ニューラルネットワークをさらに多層に重ねて、より複雑な特徴を抽出し、非線形問題も含めた様々な問題を扱えるように設計されたのが深層学習である。

2.3.2 深層学習の概要

深層学習は、機械学習における一分野で、ニューラルネットワークを多層に重ねたものである。画像処理に利用される Convolutional Neural Networks [LeCun et al., 1998] や系列データの処理に利用される Recurrent Neural Networks [Williams and Zipser, 1989] など、目的に応じた様々な拡張があるが、どれも図 2.6 のような多層パーセプトンというモデル構造を基本としている。多層パーセプトロンも、層間の信号の伝搬など、基本的な構造は単純パーセプトロンと大きな違いはない。しかし、隠れ層が多層になったことで、パラメータ更新の際に何層にも渡って微分の連鎖規則を繰り返すことが必要になり、計算コストが膨大になってしまう。そのため考案されたのが誤差逆伝搬法 [Rumelhart et al., 1988] である。誤差逆伝搬法では、出力結果に基づいて、出力層から入力層に向かって順番に重みを修正する手法により、複雑な問題を説明するような、ユニット間の重みを学習できるようになり、非線形問題も解くことが可能になった。

この誤差逆伝搬法や、確率的勾配降下法 [Robbins and Monro, 1951, Kushner and Yin, 2003], AdaDelta [Zeiler, 2012], Adam [Kingma and Ba, 2014], Nadam [?] などのより

効率的な勾配降下法、モデルの過学習を抑制する dropout [Srivastava et al., 2014] と呼ばれる機構など、様々な技術的工夫が考案されたことにより、現実的な計算コストで効率的に深層学習を行うことが可能になり、深層学習を用いた研究活動が急速に活発化した。

深層学習の活用により、画像認識 [Schroff et al., 2015, Szegedy et al., 2014]、音声認識 [Hinton et al., 2012, Bahdanau et al., 2015]、会話認識 [Sak et al., 2015]、機械翻訳 [Sutskever et al., 2014, Dong et al., 2015]、質問応答文生成 [Yin et al., 2015]、画像説明文生成 [Xu et al., 2015, Vinyals et al., 2014] 等、多様な研究領域で飛躍的な進展が報告がされている。特に、直近の一年間だけでも画像から動画を生成する研究 [Vondrick et al., 2016] や、会話を人間と同程度に認識できるとする音声認識の研究 [Xiong et al., 2016a]、一部の欧米言語間の文レベルで、ほぼ人間と同等に正確な翻訳を実現したとする機械翻訳の研究 [Wu et al., 2016] などを始めとする数々の報告がされており、深層学習によって、日々驚異的な成果が生み出されている。

また、2016年3月に人間のプロを倒したことで一躍有名になった、Google Deep Mindが開発したコンピュータ囲碁プログラムの「AlphaGo」[Silver et al., 2016]は、過去の人間が打った大量の棋譜に深層学習を適用した後、自己対局による強化学習を通して、今後10年は不可能と言われていた、人間のプロを打ち負かすほどの棋力を獲得した。AlphaGoは、過去の対局の情報である棋譜の分析によって人間を真似ただけでなく、それまで人間が考えつかなかつたような手を学習しており、囲碁界に衝撃を与えていた。このように、深層学習は、人間が認識できないようなデータの複雑な特徴を捉えることで、これまで人間が作り上げてきた概念を、大きく塗り替える可能性を秘めている。

一般に、深層学習モデルを学習させる際には、大規模な訓練データが必要となる。深層学習モデルが、人の手で素性を設計していない生の訓練データから、特徴的な表現を学習し、最適化するには、膨大な数の内部パラメータを設定して学習することが必要となる。ときには数十万から数百万以上の内部パラメータが設定されることもあり、こうした膨大な数のパラメータを学習するには、大規模な訓練データが必要となる。データ数が不足すると、データの潜在的な特徴を十分に学習できないことに加え、汎用性の低い特徴まで過剰に学習してしまう過学習に陥りやすくなる [Tetko et al., 1995]。

実際に大規模データを利用した研究の例を挙げると、人間より高い精度で人の顔を見分けらるる報告する顔認識の研究 [Schroff et al., 2015] では、数百万人の2億枚以上の顔画像を訓練データに利用している。英語からフランス語に翻訳する機械翻訳の研究 [Xu et al., 2015] では、1200万もの文章を訓練データとして利用している。

2.3.3 Recurrent Neural Networks

深層学習のネットワークには、目的に応じたいくつかの種類があるが、ここでは、知識獲得の予測に深層学習を用いた手法 [Piech et al., 2015] に用いられていたニューラルネットワークである、Recurrent Neural Networks [Williams and Zipser, 1989]（以下、RNN）について説明する。

RNNは深層ニューラルネットワークの一種で、主に系列データの解析に利用される。系列データとは、同質のデータを直列に並べて表現することにより、特定の意味を持ったデータのこと、例えば、時系列に沿って変化する株価のようなデータや、一定の長さで順序を持って並ぶ単語から構成される文章などのデータが系列データにあたる。

近年、RNNはデータの大規模化や計算機性能の向上などにより、幅広い領域の系列データに対して適用されるようになった。具体的には、機械翻訳 [Sutskever et al., 2014, Dong et al., 2015]、手書き文字認識 [Graves and Schmidhuber, 2009, Louradour and Kermorvant, 2014]、音声認識 [Hinton et al., 2012, Bahdanau et al., 2015]、ユーザログ解析 [Hidasi et al., 2015]、画像説明文生成 [Xu et al., 2015, Vinyals et al., 2014]、医療診断 [Choi et al., 2015, Lipton et al., 2015] 等の領域で高い性能を発揮することが報告されている。

伝統的なRNNは、入力層、隠れ層、出力層の3層から構成されている。系列方向を時刻とすれば、時刻 t の入力 \mathbf{x}_t と時刻 $t-1$ の隠れ層 \mathbf{h}_{t-1} の情報を入力として、式2.8のように時刻 t の隠れ層 \mathbf{h}_t が計算されるような、一つ前の情報を繰り返し（recurrent）入力するという構造である。

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}) \quad (2.8)$$

なお、関数 f は活性化関数であり、シグモイド関数や双曲線正接関数、Relu [Nair and Hinton, 2010]、ELUs [Clevert et al., 2015]などの非線形関数が用いられるのが一般的である。

モデル構造は図2.7のように表され、隠れ層の部分を時間方向に展開して図2.8のように表されることもある。

このように、データの系列に沿った情報を反映して学習できるRNNだが、長期的な表現になるほど学習が難しくなるという課題がある [Bengio et al., 1994]。RNNの学習には、様々な勾配法に基づいた最適化が行えるが、いずれの勾配法を用いる場合においても、式2.8に表れるように同じ変換を繰り返し行うため、隠れ層が多層になり、勾配が爆発して

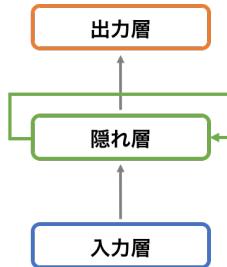


図 2.7: RNN の基本構造

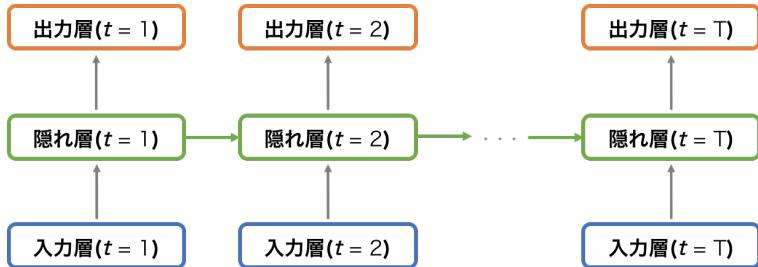


図 2.8: RNN の基本構造 (展開)

学習モデルが壊れてしまうという勾配爆発 [Bengio et al., 1994, Pascanu et al., 2013] という問題や、勾配が消滅して対象データの長期的な特徴量を捉えることができないという勾配消滅 [Pascanu et al., 2013, Hochreiter, 1998] という問題が発生する。

こうした問題を解決もしくは緩和する手段の一つが、学習時の勾配に制約を加える方法である。具体的には、学習させるパラメータの勾配の絶対値の最大値を予め決めておき、最大値以上の場合には、勾配の最大値になるように勾配の値を置き換えることで勾配爆発の影響を緩和する方法 [Mikolov, 2012] や、学習させるパラメータの勾配のノルム⁴の最大値を予め決めておき、最大値以上の場合には、ノルムが最大値以下になるようにノルムを抑制することで勾配爆発の影響を緩和する方法 [Pascanu et al., 2013] などが報告されている。

もう一つの手段が、ゲート付き活性化関数の利用である。先に言及したが、RNN には異なる活性化関数を利用するという形でいくつかの種類がある。うまく設計された活性化関数を利用することで、勾配消滅を緩和してデータの長期的な特徴をよく捉えられたり、計算コストを削減することができます。以降では、よく研究報告で取り上げられる Simple RNN (以下、SRNN) [Williams and Zipser, 1989], Long Short Term Memory RNN (以下、LSTM-RNN) [Hochreiter and Schmidhuber, 1997], Gated Recurrent Neural Networks (以下、GRNN) [Cho et al., 2014] の 3 つについて詳細に説明する。

SRNN

SRNN はゲート付き活性化関数を用いない簡単な構造の RNN である。[Le et al., 2015, Krueger and Memisevic, 2015] で報告される工夫を取り入れることで、データの長期的な特徴を効果的に捉えることができるようになるが、多くの場合で、LSTM-RNN や GRNN のようにゲート付き活性化関数を用いる RNN の方がモデルの性能という点で優れている。

⁴ベクトルの「長さ」の概念を一般化したもの

SRNN による最も単純なモデルの定式は以下の式で定義される。

$$\mathbf{h}_t = \tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \quad (2.9)$$

$$\mathbf{y}_t = \sigma(\mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y) \quad (2.10)$$

ここでは、 t は時刻を指し、 \mathbf{x}_t は時刻 t の入力ベクトルを指し、 \mathbf{h}_t は時刻 t の隠れ層を指し、 \mathbf{y}_t は時刻 t の入力ベクトルを元にした予測値を指し、 \mathbf{W}_{xh} 、 \mathbf{W}_{hh} はそれぞれ重み行列を指し、 \mathbf{b}_h 、 \mathbf{b}_y はそれぞれバイアス項を指し、 \tanh は $(e^x - e^{-x})/(e^x + e^{-x})$ で定義される双曲線正接関数を指し、 σ は $1/(1 + e^{-x})$ で定義されるシグモイド関数を指す。訓練時には、重み行列 \mathbf{W}_{xh} 、 \mathbf{W}_{hh} とバイアス項 \mathbf{b}_h 、 \mathbf{b}_y を学習する。

LSTM-RNN

LSTM-RNN は Long Short Term Memory という活性化関数を用いる RNN で、その名前の通り、SRNN では捉えることが難しかったデータの長期的表現と短期的表現の両方の獲得を目的に開発されたものである [Hochreiter and Schmidhuber, 1997]。LSTM-RNN は SRNN と比較すると、モデルの性能という点で優れているが、内部のパラメータの数が非常に大きく学習コストは大きい。最先端の成果を報告する研究でしばしば利用されているが、LSTM-RNN 自体が開発されたのは 1997 年であり LSTM-RNN が新しいというわけではない。

LSTM-RNN によるモデルの定式にはいくつか種類が存在するが、特に、後述する Deep Knowledge Tracing [Piech et al., 2015] で用いられる LSTM-RNN は下記の式で定義される。

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i) \quad (2.11)$$

$$\mathbf{g}_t = \sigma(\mathbf{W}_{xg}\mathbf{x}_t + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_g) \quad (2.12)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f) \quad (2.13)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o) \quad (2.14)$$

$$\mathbf{m}_t = \mathbf{f}_t \odot \mathbf{m}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \quad (2.15)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{m}_t \quad (2.16)$$

$$\mathbf{y}_t = \sigma(\mathbf{W}_{my}\mathbf{m}_t + \mathbf{b}_y) \quad (2.17)$$

ここでは、 \mathbf{i}_t は Input Gate を指し、 \mathbf{f}_t は Forget Gate を指し、 \mathbf{g}_t はメモリセルへの入力

を指し, \mathbf{o}_t は Output Gate を指し, \mathbf{m}_t はメモリセルを指し, \mathbf{W}_{xi} , \mathbf{W}_{hi} , \mathbf{W}_{xg} , \mathbf{W}_{hg} , \mathbf{W}_{xf} , \mathbf{W}_{hf} , \mathbf{W}_{xo} , \mathbf{W}_{ho} , \mathbf{W}_{my} はそれぞれ重み行列を指し, \mathbf{b}_i , \mathbf{b}_g , \mathbf{b}_f , \mathbf{b}_o , \mathbf{b}_y はそれぞれバイアス項を指し, \odot は要素積を指す.

式 2.15 にあるように, メモリセルへの入力は 1 つ前のメモリセルの状態 \mathbf{m}_{t-1} と入力 \mathbf{g}_t であり, それぞれの入力に対して, 過去のメモリセルからの情報を捨てる Forget Gate と現在からの情報を調整する Input Gate を作用させ, \mathbf{m}_t を得る. 新しい隠れ層 \mathbf{h}_t は式 2.16 のようにメモリセルからの出力を Output Gate で調整したものを入力として受け取る. これらのゲートにより, 長期的な特徴と短期的な特徴が捉えられるとされている.

GRNN

GRNN は Gated Recurrent Unit [Cho et al., 2014] というゲート付き活性化関数を用いる RNN のことで, GRU は LSTM のように, 長期的な表現と短期的な表現を捉えるために提案された活性化関数である. Cho ら [Cho et al., 2014] が 2014 年に発表して以来, GRNN 自体や GRNN の活用に関する研究が多く報告されている [Chung et al., 2014, Zaremba, 2015, Chung et al., 2015, Karpathy et al., 2015, Biswas et al., 2015, Pezeshki, 2015]. LSTM よりもゲートの数が少なく学習コストが小さい傾向にあるが, LSTM-RNN, GRNN の性能を比較した研究 [Chung et al., 2014, Zaremba, 2015] において LSTM-RNN と GRNN が同程度の性能であることが報告されている.

GRNN は下記の式により定義される.

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r) \quad (2.18)$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z) \quad (2.19)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1} + \mathbf{b}_h)) \quad (2.20)$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t \quad (2.21)$$

$$\mathbf{y}_t = \sigma(\mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y) \quad (2.22)$$

ここでは, $\mathbf{W}_{xr}, \mathbf{W}_{hr}, \mathbf{W}_{xz}, \mathbf{W}_{hz}, \mathbf{W}_{xh}, \mathbf{W}_{hh}$ は重み行列で, $\mathbf{b}_r, \mathbf{b}_z, \mathbf{b}_h$ はバイアス項である. \mathbf{r}_t が Reset Gate(LSTM における Forget Gate に相当する機構) で, \mathbf{z}_t が Update Gate(LSTM におけるメモリセルに相当する機構) である. \mathbf{r}_t が 0 に近いほど前の隠れ層からの入力よりも現在の入力をより強く考慮するようになり, \mathbf{z}_t が 0 に近いほど前の隠れ

層をより大きく更新するようになる。

以上、周辺知識を整理した上で深層学習について述べ、本研究と関連の深いRNNについて詳述することで、技術的な前提知識を確認した。

次に、知識獲得の予測について述べる。

2.4 知識獲得の予測

知識獲得の予測は、生徒が対象の知識を獲得しているか否かを予測する問題である。通常、知識を獲得しているか否かは問題回答の正誤を基に評価されるため、知識獲得の予測のタスクは過去の生徒の問題回答履歴から次に解く k 問題の回答正誤を予測するというものである。

最初の定式化の事例は、1994年にCorbettらによって報告されたKnowledge Tracing [Corbett and Anderson, 1994] である。スキルの習熟学習において、領域知識をよく分析し階層的に知識間関係を構築し、階層構造においてより水準の高い知識に着手する前に予め獲得すべき知識が確実に獲得されるように学習体験を設計することで、ほとんどの生徒がスキルを十分に習熟できるとする仮説 [Keller, 1968, Bloom, 1968] や、コンピュータサイエンスの発展を受けて、予め獲得すべき知識が確実に獲得されるように生徒の知識の獲得有無を予測するというのが主な目的であった。Knowledge Tracingにおける生徒とモデルは、生徒が勉強し知識を獲得したら、モデルが生徒が獲得した知識を予測することで生徒の獲得している知識の変化を追跡する (*Knowledge Tracing*)、という関係になっている。

伝統的に、知識獲得の予測には知識獲得の時系列性を重視するものと、知識間の関係性を重視するものがある。[Corbett and Anderson, 1994]で報告されたBayesian Knowledge Tracing という手法は知識獲得の時系列性を重視するもので、問題に予めスキルを割り当て個々のスキルに習熟過程に関する4つの確率変数を定義しモデル化するというもので、スキル間の関係性は考慮しないが、個々のスキルの習熟、つまり時系列性を考慮する手法である。[Pavlik Jr et al., 2009]で報告されたPerformance Factor Analysis という手法は知識間の関係性を重視するもので、個々の知識（あるいは、スキル）に関する過去の回答の正誤を重み付けして、次の問題の正誤を予測しようというもので、Performance Factor Analysisは知識獲得の時系列性より知識間の関係性を重視する手法である。いずれの手法も本論文と関連が深い。

以降では、まず、Knowledge Tracing の定式化について述べ、Bayesian Knowledge Tracing と Performance Factor Analysis の 2 つの手法を説明し、最後に、深層学習を活用した Deep Knowledge Tracing について説明する。

2.4.1 Knowledge Tracing の定式化

Knowledge Tracing は過去の生徒の問題回答履歴から生徒が次に解く問題の正誤を予測するというものである。生徒の時刻 t において観測された問題回答結果を q_t とすれば、 q_1, q_2, \dots, q_t から時刻 $t + 1$ において観測される問題回答結果 q_{t+1} を予測するタスクと表現できる。特に、過去の観測された問題の正誤から将来の正誤確率を算出する場合は、 q_1, q_2, \dots, q_t が観測された場合の時刻 $t + 1$ に着手する問題において当該生徒の回答正解となる事後確率 $p(q_{t+1} = \text{correct} | q_1, q_2, \dots, q_t)$ を求めるタスクであるといえる。予測性能の評価は [Yudelson et al., 2013, FALAKMASIR et al., 2015] では Accuracy⁵で、[Piech et al., 2015] では AUC⁶で行っており、目的に応じてさまざまである。本研究では、AUC によって予測性能を評価する。

なお、モデルの入力となる「問題」の粒度はさまざまである。問題への回答の結果は、その問題を回答するのに必要な知識を生徒が獲得しているか否かを評価するという点で、知識集合を表現しており、また、その粒度もさまざまである。個々の問題をそのままモデルの入力次元とするものや、問題に予めタグを割り当て問題により評価される知識の粒度をある程度整え、そのタグをモデルの入力次元とすることもある。例えば、[Piech et al., 2015] ではモデルの入力次元は演習タグもしくはスキルタグと呼ばれるものであり、演習問題に割り当てられ、それぞれの演習問題で扱われる知識の要素を説明するものである。通常、こうしたタグは専門家によって設計され、利用される。本論文では、個々の問題をそのままモデルの入力次元として用いる。

2.4.2 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing [Corbett and Anderson, 1994]（以下、BKT）はベイズの定理の事前確率と事後確率の関係に基づいて正解確率 $p(q_{t+1} = \text{correct} | q_1, q_2, \dots, q_t)$ をモデリングする手法である。BKT には下記の 4 つの確率変数がある。

⁵正解率。予測結果全体と、答えがどれくらい一致しているかを判断する指標。0~1 で表され、完全な予測時に 1 となる。

⁶正例を正しく分類した割合を縦軸に、負例を正しく分類した割合を横軸に取る ROC 曲線における、曲線より下の面積。0~1 で表され、完全な予測時に 1 となり、ランダムな予測で 0.5 付近を示す。

- 初めから当該スキル理解している確率 $p(L_0)$ (もしくは $p\text{-init}$)
- 生徒が当該スキルを理解していない状態から理解している状態へ遷移する確率 $p(T)$ (もしくは $p\text{-transit}$)
- 生徒が当該スキルを理解しているが誤答する確率 $p(S)$ (もしくは $p\text{-slip}$)
- 生徒が当該スキルを理解していないが推測で正解する確率 $p(G)$ (もしくは $p\text{-guess}$)

これらの4つの確率変数がすべてのスキルに定義されている。つまり、スキル数を N とすれば、確率変数の合計数は $4N$ である。生徒 u がスキル k の問題を時刻 t に解いた場合に正解する確率は下記の式に基づいて更新される。

$$p(L_1)_u^k = p(L_0)^k \quad (2.23)$$

$$p(L_t|obs=correct)_u^k = \frac{p(L_{t-1})_u^k \cdot (1 - p(S))^k}{p(L_{t-1})_u^k \cdot (1 - p(S))^k + (1 - p(L_{t-1})_u^k) \cdot p(G)^k} \quad (2.24)$$

$$p(L_t|obs=wrong)_u^k = \frac{p(L_{t-1})_u^k \cdot p(S)^k}{p(L_{t-1})_u^k \cdot p(S)^k + (1 - p(L_{t-1})_u^k) \cdot (1 - p(G))^k} \quad (2.25)$$

$$p(L_t)_u^k = p(L_t|obs)_u^k + (1 - p(L_t|obs)_u^k) \cdot p(T)^k \quad (2.26)$$

$$p(C_t)_u^k = p(L_{t-1})_u^k \cdot (1 - p(S))^k + (1 - p(L_{t-1})_u^k) \cdot p(G)^k \quad (2.27)$$

右上の k はスキル番号を示し、右下の u はユーザ番号を示すことに注意されたい。まず、生徒 u が初めから当該スキル k を身につけている確率は式 2.23 の通り定義する。正解が観測され、正しく当該スキルを身につけている確率は、式 2.24 で与えられ、不正解が観測されたが、正しく当該スキルを身につけている確率は、式 2.25 で与えられ、それらを合わせて、次の時刻に当該スキルを身につけている確率は、式 2.26 で与えられる。このように定めることで、理解しているがうっかり間違ってしまう場合や、理解していないがあてずっぽうで正解してしまう場合を考慮できる。なお当該モデルでは、身につけたスキルの忘却は無視している。最後に、生徒 u がスキル k の問題を時刻 t に解いた場合に正解する確率 $p(C_t)_u^k$ は式 2.27 のように算出され、この値を次の問題の正誤予測に利用する。

上記に説明したモデルの学習にはいくつかの方法が適用され報告されている。1つは [Corbett and Anderson, 1994] にあるように隠れマルコフモデル (HMM) を用いて生成モデルとして学習させる方法であり、1つは [Yudelson et al., 2013] にあるように勾配法を用いて識別モデルとして学習させる方法である。それぞれ長所と短所があるが、特に、大

規模データへの適用という観点では HMM に基づいた生成モデルの手法では計算量が大きく学習に非常に多くの時間がかかることが多いということもあり, [Yudelson et al., 2013] では勾配法に基づいた識別モデルとして学習させている。具体的には, [Yudelson et al., 2013] では、目的関数に負の対数尤度 (Negative Log Likelihood) を利用し、勾配降下法 (Gradient Descent) で学習させている。

2.4.3 Performance Factor Analysis

Performance Factor Analysis [Pavlik Jr et al., 2009] (以下, PFA) も過去の生徒の問題回答履歴から生徒が次に解く問題の正誤を予測するための手法である。しかし、知識獲得の時系列性を考慮する BKT と異なり、知識獲得の順番を考慮せず知識間の関係性を考慮して予測する手法である。PFA は下記のように定義される。

$$p(i, j \in KCs, s, f) = \sigma(\beta_j + \sum_{k \in KCs} (\gamma_k s_{i,k} + \rho_k f_{i,k})) \quad (2.28)$$

ここでは、 s は事前に正答した問題回答、 f は事前に誤答した問題回答、 p はユーザ i が知識 j に正答する確率、 β_j は知識 j の簡単さ、 γ_k と ρ_k はそれぞれ知識 k の正答と誤答の重み、 $s_{i,k}$ と $f_{i,k}$ はそれぞれユーザ i が知識 k に事前に正答した問題回答、事前に誤答した問題回答である。 σ はシグモイド関数、過去の各知識の正誤を重み付けしシグモイド関数にかけ、別の問題の正誤を予測するというものである。

PFA は知識間の関係性を考慮できない場合に複数の知識がないと獲得できない知識のモデルが難しいという Bayesian Knowledge Tracing やその拡張手法の問題を解決するために提案された。PFA は知識間の関係性を重み付けして考慮しているが、問題回答の順番は考慮しない。

2.4.4 Deep Knowledge Tracing

Deep Knowledge Tracing [Piech et al., 2015] (以下, DKT) は RNN を利用し Knowledge Tracing を行う手法である。2015 年 6 月に発表された。数学の問題回答ログのデータセットで実験され、高い性能で将来の知識獲得を予測できること、予測モデルを分析することで知識間関係をネットワークとして抽出できることが報告された。生徒が獲得している知識から、ある知識の獲得されやすさをそのまま予測しており、得られた知識間関係から抽

出されたネットワークは知識獲得における知識構造を表現しているといえる。DKT の構造と最適化、および知識間関係の抽出手法について順に説明していく。

構造

まず、DKT の構造について述べる。DKT の構造は伝統的な RNN の構造に基づいている。伝統的な RNN は入力のベクトル系列 $\mathbf{x}_1, \dots, \mathbf{x}_T$ を出力のベクトル系列 $\mathbf{y}_1, \dots, \mathbf{y}_T$ に写像する。この写像は、隠れ状態 $\mathbf{h}_1, \dots, \mathbf{h}_T$ を計算することで達成されるが、一連の写像の過程で過去観測から得られる関連情報を将来予測のために連続的に符号化している、とみなせる。確率変数は下記の式で定義されるネットワークにより関連付けられる。

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}) \quad (2.29)$$

$$\mathbf{y}_t = g(\mathbf{h}_t) \quad (2.30)$$

モデルは関数 f と g によって定義されており、これらの関数 f, g には SRNN の式 2.9, 2.10 や LSTM-RNN の式 2.11–2.17, GRNN の式 2.18–2.22 を利用できる。

RNN で生徒の学習行動の観測結果をモデリングするため観測結果を固定長の入力ベクトル \mathbf{x}_t の系列に変換する必要があるが、DKT ではシンプルな変換を行っている。具体的には、生徒の学習行動の観測結果を one-hot ベクトルに符号化し \mathbf{x}_t とする、というものである。観測結果は演習問題と正誤の組み合わせで表現できるため、演習問題の数を M とすれば、 \mathbf{x}_t の長さは $2M$ となる。

表 2.1: Deep Knowledge Tracing における回答ログデータと対応する入力ベクトルの例

| ユーザ ID | 回答ログ | | | 入力ベクトル | |
|--------|-------|------|----|----------------|-------------|
| | ログの順番 | 問題番号 | 正誤 | 変数名 | 値 |
| A | 1 | 1 | 0 | \mathbf{x}_1 | [0000:1000] |
| A | 2 | 1 | 1 | \mathbf{x}_2 | [1000:0000] |
| A | 3 | 2 | 1 | \mathbf{x}_3 | [0100:0000] |
| A | 4 | 3 | 0 | \mathbf{x}_4 | [0000:0010] |
| A | 5 | 3 | 1 | \mathbf{x}_5 | [0010:0000] |
| A | 6 | 4 | 1 | \mathbf{x}_6 | [0001:0000] |

具体例を交えて説明する。例えば、演習問題の数が 4 つで、問題回答は 1 つずつしかで

きないと仮定する。 $M = 4$ であり、 \mathbf{x}_t の長さは8である。ある生徒が、表2.1の回答ログのように問題を回答し正誤が観測されたとする。この時に、例えば、表2.1に記載のような入力ベクトルの系列となる。このようにして、回答行動の観測結果を符号化することで、どの演習問題をいつ正解もしくは不正解したのかをRNNに入力できる。

出力 \mathbf{y}_t は問題と同じ長さのベクトルで、それぞれの要素が当該生徒がそれぞれの問題に正しく回答する確率の予測値となっている。したがって、 $t+1$ の回答 q_{t+1} の正誤予測は $t+1$ に回答される問題 q_{t+1} に対応する \mathbf{y}_t の要素から読み取れる。

最適化

次に、DKTの最適化について述べる。訓練時に用いられる目的関数は、モデルにおいて生徒の回答行動の観測系列の負の対数尤度（Negative Log Likelihood）である。 $\delta(q_{t+1})$ を時刻 $t+1$ にどの問題が回答されたかのone-hotベクトルとし、 a_{t+1} を時刻 $t+1$ に当該問題で正答したか否か（1か0）とし、 l を交差エントロピー誤差とすれば、当該予測結果に対する損失関数は $l(\mathbf{y}^T \delta(q_{t+1}), a_{t+1})$ であり、生徒一人の損失関数は下記の式で与えられる。

$$L = \sum_t l(\mathbf{y}_t^T \delta(q_{t+1}), a_{t+1}) \quad (2.31)$$

学習時はミニバッチごとに確率的勾配降下法で目的関数を最小化する。[Piech et al., 2015]では、モデル学習時には過学習を防ぐため \mathbf{y}_t への入力としての \mathbf{h}_t にはdropout [Srivastava et al., 2014]を適用している（ \mathbf{h}_{t+1} の方向にはdropoutを適用しない）。また、系列方向の誤差逆伝搬 [Werbos, 1990]において勾配が爆発するのを防ぐため、閾値以上のノルムの勾配は [Pascanu et al., 2013]にしたがい、制約を設けている。

知識間関係抽出法

次に、DKTのモデルを利用した知識間関係（あるいは、問題間関係）抽出法について述べる。DKTのモデルは、従来ではよく人間の専門家が行っていたデータの潜在的な構造や概念を発見するタスクに応用できる。問題 i と j のすべての有向ペアのうち下記の条件を満たすものに対して下記の影響度 J_{ij} を割り当てる。

条件 有向ペア (i, j) について、問題 i が出現した後に残りの問題系列の中で問題 j が出現する系列数が問題 i が出現する問題系列数全体の $V\%$ 以上であること。

影響度

$$J_{ij} = \frac{y(j|i)}{\sum_k y(j|k)}$$

ここでは、 $y(j|i)$ は、ある生徒が最初に問題 i に正答した場合に、RNN によって割り当てられる次の時刻に問題 j に正答する確率である。[Piech et al., 2015] では、問題間影響行列からのネットワーク抽出には、 $V = 1$ を用いた。また、ネットワークの可視化に際しては、影響度が 0.1 以上であればエッジを引くというようにしてネットワークを構築した。

さらに、[Piech et al., 2015] は、得られたネットワークは、単に生徒の問題 (i, j) 間の遷移率から構築したネットワークや問題 i の正解が観測された後に問題 j の正解が観測される条件付き確率から構築したネットワークよりもよく知識間関係を捉えていることを指摘している。

こうして得られた行列 J は、問題 i で評価される知識が既に獲得されている場合に、問題 j で評価される知識の獲得されやすさを表現しており、 J は知識間関係行列であるといえ、この知識間関係行列から構築したネットワークは知識獲得における知識構造を表現していると考えられる。

2.4.5 知識獲得予測の手法における DKT の最適性

ここまで知識獲得予測の様々な手法について述べたが、DKT を拡張する手法が、本研究の目的を達成する上で最適な手法であることを、以下の二点に基づいて説明する。

1. 知識間の関係性は、知識獲得予測の文脈において、定量的に検証されて抽出されるべきこと。
2. 知識獲得予測は、複数の知識間の影響関係や、知識獲得の時系列性を考慮して行われるべきこと。

まず、1について述べる。知識間の関係性は、知識獲得予測の過程で抽出されるものと、そうでないものがある。後者については、専門家が作成するという手法や、テキスト解析により概念関係ネットワークを構築するという手法 [Chen et al., 2008] がある。しかし、

これらの手法は、専門家や研究者が立てた仮説に基づいた定性的なものであり、実際の生徒の学習過程をよく説明するものであるという定量的な根拠はない。

問題回答正誤の分析により知識の構造化を行う方法 [参考文献引用] も、2つの問題 i と j の間で問題 i が正解後と不正解後の問題 j の正解率の差と着手順序を基に知識を構造化するが、この手法は2つの問題 i と j の関係性のみを考慮しており、他の問題との関係性は独立だと見なされている。得られた知識間関係は2つの知識の間についてのものを線形に合算したものであり、複雑で密接に関係している複数の知識の獲得順序や影響関係を捉えているものではない。

一方で、知識間の関係性の抽出を、知識獲得を予測する過程で行うものは、生徒の知識状態と行動を元に、知識の獲得を予測しているため、生徒の学習過程を反映した知識間関係を表現している可能性が高い。

したがって、知識間関係を定量的に抽出する手法としては、知識獲得を予測する過程で知識間関係を抽出する手法に絞る。

次に、2について述べる。知識獲得予測には、複数の知識の影響関係や知識獲得の時系列性を考慮するものと、そうでないものがある。後者については、複数の知識を独立なものとして、それらの状態の遷移を定義する Bayesian Knowledge Tracing(BKT) の手法や、過去の回答の結果を1つにまとめて定義する Performance Factor Analysis(PFA) の手法などがある。しかし、BKT の手法は、複数の知識を独立なものとして捉えるため、複数の知識からなる複雑な知識状態を捉えきれず、また、PFA の手法は、直前の回答も十分な時間が立った後の回答も、一つの過去の回答として捉えるため、生徒の時間に沿った知識獲得の状態を、現実的に捉えきれていない。

一方、DKT は、時系列に沿って RNN の隠れ層を更新することにより、複数の知識間の影響関係や、知識獲得の時系列性を考慮して知識獲得を予測しているので、より現実に沿った知識間関係を抽出できる可能性が高い。

現に [Piech et al., 2015] で、既に DKT によって知識間関係を抽出できることが報告されており、複雑で密接に関係している複数の知識の獲得順序や影響関係を捉えている可能性が高く、DKT を利用することが最適であると考えられる。

また、PFA と DKT のいずれの手法も知識間関係を考慮して予測に利用しているが、DKT の方が有効性が高いと考える。なぜなら、[Piech et al., 2015] では言及されていなかった

が、DKTはPFAの拡張になっているためである。DKTはRNNを利用しており、

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}) \quad (2.32)$$

$$\mathbf{p}_t = \sigma(\mathbf{h}_{t-1} \cdot \mathbf{W}_{hp} + \mathbf{b}_p) \quad (2.33)$$

で与えられる。一方でPFAは

$$p(i, j \in KCs, s, f) = \sigma(\beta_j + \sum_{k \in KCs} (\gamma_k s_{i,k} + \rho_k f_{i,k})) \quad (2.34)$$

で与えられる。したがって、

$$f(\mathbf{x}_t, \mathbf{h}_{t-1}) = \mathbf{x}_t + \mathbf{h}_{t-1} \quad (2.35)$$

$$\mathbf{h}_0 = [0, 0, \dots, 0] \quad (2.36)$$

とすると、 \mathbf{h}_t がこれまでの各問題についての正答回数を表現するベクトルと各問題についての誤答回数を表現するベクトルを結合したベクトルになるが、これは、ベクトル \mathbf{s} と \mathbf{f} を結合したものと同じである。したがって、PFAはDKT内部のRNNの繰り返しの部分を表現する関数 f を式 2.35 にした特殊なケースであり、DKTはPFAの拡張になっていいる。

以上、知識獲得の予測研究について述べ、本研究の素地となっている研究について整理した。次に、次元削減手法について述べる。

2.5 次元削減手法

高次元のデータから低次元の特徴表現を抽出する、次元削減手法について述べる。本研究が目的とする知識分類表現の抽出は、一般的な次元削減の手法を拡張したものであり、一般的な次元削減手法について説明することで、基礎となる知識や本研究との差分を明確にすることを目的とする。

一般に、機械学習や統計においては、扱うデータの次元が大きい場合に、次元削減を行うことが多い。これは、データの次元が大きすぎることにより、データのサンプル数に対してモデルが複雑化してしまい、認識精度が悪くなる「次元の呪い」[Bellman and Corporation, 1957, Friedman, 1997] という現象を回避する目的の他、可読性を高めることにより、人間が解釈しやすくなることなどを目的としている。知識獲得予測において用い

られる知識分類も、分類のなされていない生の問題は次元数が大きく、そのままでは人間が解釈したり教育に用いることが困難なため、内容や難易度の類似度など、一定の尺度に基づいて人間の手により次元削減が行われた例である。本研究ではこうした次元削減を、人間の手ではなく深層学習によって行うことで、最適化することを目的としている。

本節では、機械学習が現れる以前から一般的に使用されていた次元削減手法の代表である Principal Component Analysis や、ニューラルネットワークを活用した次元削減手法である Autoencoder、そして、深層学習の過程で行われる Embedding と呼ばれる埋め込み手法を取り上げ、次元削減手法について概観する。

2.5.1 Principal Component Analysis

Principal Component Analysis(PCA) は、日本語では主成分分析と訳される。相関のある多数の変数の中で分散の大きい変数を、データ全体を説明する上で重要な「主成分」と見なし、順にそれまでの主成分と直行するように主成分を定める変換を繰り返し行うことで、変数間の相関をなくし、重要度の高い主成分のみを採用し、次元削減を行う手法である。

その由来は古く、1901年に力学の分野において初めて導入された [Pearson, 1901] ことをきっかけに、以後、経済学や統計学、機械学習などの分野で幅広く利用されてきた [Wold et al., 1987, Ku et al., 1995]。PCA には様々な拡張がある [Schölkopf et al., 1997, Tipping and Bishop, 1999] が、その根底にある数学的な意味は、固有値問題を解くことにある。

一般的なアルゴリズムを以下に示す。

元のデータを \mathbf{X} 、変換後の次元数を N とすると、

1. データの共分散行列を求める。
2. 共分散行列の固有値と固有ベクトルを求める。
3. 固有値の大きい順に、対応する固有ベクトルを並べ替え、 N 個の固有ベクトルを並べた行列 \mathbf{P} を作る。
4. データからその平均ベクトルを引いたデータを \mathbf{X}_{bar} とし、以下の式に基づいてデータを変換する。

$$\mathbf{X}_{pca} = \mathbf{X}_{bar}\mathbf{P} \quad (2.37)$$

2.5.2 Autoencoder

Autoencoder は、日本語では自己符号化器と訳される。1980 年代から考案されていたが、2006 年の [Hinton and Salakhutdinov, 2006] らの研究によって広く普及した。様々な種類のものが考案されているが、基本的な概念は、ニューラルネットワークに入力されたデータを一旦低次元で表現し、その低次元表現から再び入力である自身を再現するように学習させることで、データの特徴を適切に表す低次元表現を獲得することを目的としている。教師データが存在しないが、自身を教師データとして行う教師あり学習のような形を取っており、教師あり学習と教師なし学習の中間に位置するような概念として理解されることもある。

具体的な定式化について述べる。まず、入力層 $x \in \mathbb{R}^d$ に対して、以下の式によって隠れ層 h と復元層 x^r を定義する。

$$h = f_{\theta}(x) = s(\mathbf{W}x + b) \in \mathbb{R}^{d_h} \quad (2.38)$$

$$x^r = g_{\theta'}(h) = s_r(\mathbf{W}'h + b') \in \mathbb{R}^d \quad (2.39)$$

ここで、 $\theta = (\mathbf{W}, b), \theta' = (\mathbf{W}', b')$ は学習されるパラメータであり、 s, s_r は活性化関数である。こうして定義された復元層 x_r が入力層 x にできるだけ近づくように、訓練データ $\mathbf{D} = x_1, \dots, x_n$ に対する損失関数の平均値を最小化する過程で、パラメータ θ, θ' を学習する。

$$\min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^n L(x_i, g(f(x_i))) \quad (2.40)$$

式 2.40 で表される損失関数は、一般に再構成誤差 (Reconstruction Error) と呼ばれる。損失関数 L は、入力がバイナリ値ならば交差エントロピー誤差、実数値ならば二乗誤差を用いるのが一般的であり、本研究においては、入力は問題回答の正誤を意味するバイナリ値なので、交差エントロピー誤差を用いる。なお、活性化関数が恒等写像で、損失関数が二乗誤差の場合は PCA と等価になることが知られており [参考文献]、Autoencoder は、ニューラルネットワークの構造と非線形の活性化関数を用いることにより、PCA の拡張を行っていると見なすことができる。

このように、次元削減を非線形に行うことができる Autoencoder だが、深層学習が発達した今日では、深層学習モデルの各ユニットに良い初期値を与えるための事前学習として利用されることが多い [Erhan et al., 2010]。より頑健な特徴表現を獲得させるためにデー

タにノイズを加える Denoising Autoencoder [Vincent et al., 2008] や、潜在的な特徴表現の分布に特定の確率分布が存在すると仮定する Variational Autoencoder [Kingma et al., 2014] など、様々な工夫が考案されている。

2.5.3 Embedding

Embedding は、日本語では埋め込みと訳される。一般に、ニューラルネットワークにおいて入力データの次元数が大きい場合、一度データを低次元表現に落とし込む層を設けることで、より効率的に学習が進む場合が多く、これを Embedding と呼ぶ。

学習後のモデルにおける Embedding 層は、入力データを低次元で表現する上で最適な特徴表現となっている可能性が高く、この表現を抽出することでデータの特徴を保ったまま次元削減を行うことができる。

最後に、以上の関連研究を踏まえて、本論文で使用する類似の用語について定義を明確にする。

2.6 用語の定義

本節では、本論文で用いられる類似した用語の定義を行い、意味上の違いを明確にすることで、以降の手法の説明を含めた本論文の展開を明確にすることを目的とする。

2.6.1 知識獲得予測と回答正誤予測

一般に Knowledge Tracing と呼ばれ、本研究が問題提起を行っているのは「知識獲得予測」である。しかし、本研究は従来の「知識獲得予測」が「知識」の定義として用いている知識分類を所与のものとせず、問題の回答正誤予測を最適化する過程で適切な知識分類を学習することを目的としている。よって、知識分類を学習する段階では、まだ「知識」の定義はなされていないため、この段階では正確性を期すために「回答正誤予測」という単語を用いる。

一方、学習によって得た知識分類を用いて既存の知識分類と比較を行う際には、既に「知識」の定義がなされているため、一般的な「知識獲得予測」の単語を用いる。

2.6.2 知識分類と知識タグ

本研究で扱うデータセットには、既存の「知識分類」が存在する。問題に対して事前に定義されている分類や、分類されている状況を「知識分類」と呼び、知識分類に基づいて各問題に紐づく、1つ1つの具体的なタグのことを「知識タグ」と呼ぶ。指示示している状況は似ているが、問題全体が分類されている状況を表現するときは「知識分類」の単語を、より個別の問題に紐づく具体的なタグを表現するときは「知識タグ」の単語を使用する。

以上、関連研究について述べ、本研究の学術的位置付けや周辺概念を俯瞰した。

次に、分析手法について述べる。

第3章 提案手法

本章では、提案手法について説明する。

まず、提案手法全体の流れを概説し、全体がデータの事前処理と3つの分析から構成されることを述べたのち、各分析について詳述する。

3.1 分析手法全体の流れ

分析手法全体の流れを説明する。本研究の手法は、データセットの作成という事前処理と、知識分類の学習と抽出、知識分類の予測性能の検証、知識分類の性質の比較という3つの分析から構成される。

まず、事前処理として、データセットの作成について述べる。生徒の知識獲得予測において、生徒が知識を獲得しているか否かの評価には、生徒の問題回答ログデータを対象データセットとして用いる。その際、比較検証に用いるため、知識獲得の予測に利用できる複数のデータセットを利用し、また、本研究に適用するために、いくつかの条件に基づいて対象データを抽出する。

次に、3つの分析の手法について述べる。

まず、知識分類の学習と抽出について述べる。知識獲得の予測性を最適化するような知識分類を抽出するには、問題と知識分類の最適な関係性を深層学習によって学習する必要がある。そのため、問題を知識分類に変換する写像関数をパラメータ化し、回答正誤予測の最適化の過程で同時に学習する。学習された写像関数は連続値表現の行列として表されており、これを適切に離散化することで、「深層学習タグ」として抽出する。

次に、知識分類の予測性能の検証について述べる。学習された知識分類を用いることにより、既存の知識分類や一般的な次元削減によって得た知識分類を用いる場合より、高い精度で知識獲得が予測できることを示すことで、本手法により知識獲得の予測性を最適化する知識分類が得られたことを定量的に示す。

最後に、知識分類の性質の比較について述べる。学習された知識分類の性質や構造を、既存の知識分類の構造と比較することにより、その性質を定性的に検証する。

以上の分析手法全体の流れを、図3.1にまとめた。

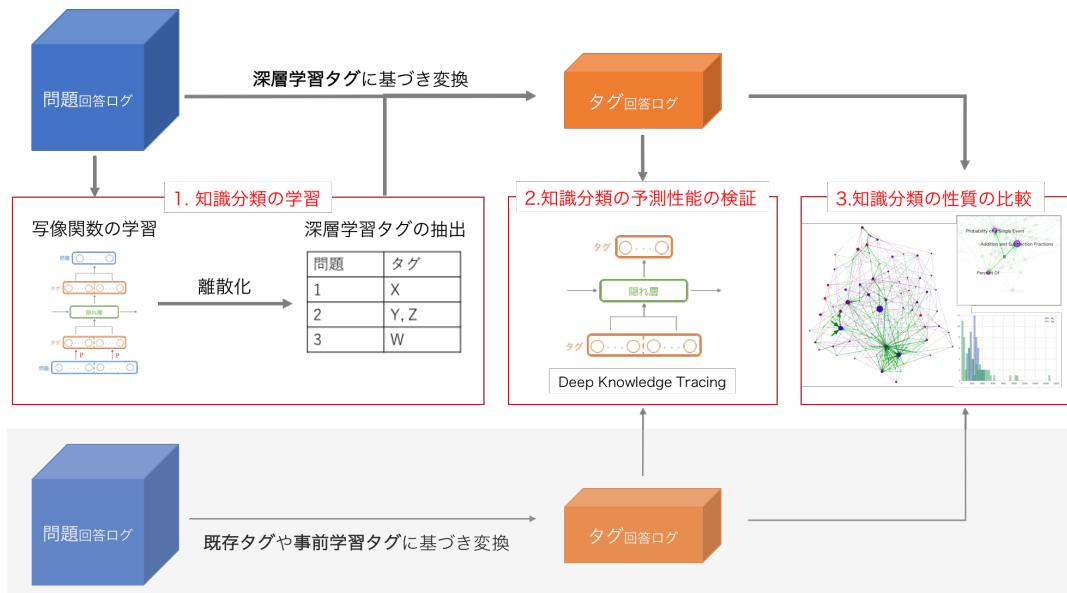


図 3.1: 分析手法全体の流れ
網掛け部は提案手法の比較対象となる手法を表す。

以降では、データセットの作成、知識分類の学習と抽出、知識分類の予測性能の検証、知識分類の性質の比較について順に詳述する。

3.2 データセットの作成

本研究においては、生徒の問題回答ログデータをデータセットとして用いる。生徒の問題への回答結果は、その問題が問う知識を、生徒が既に獲得しているか否かを表現していると捉えることができるため、回答結果が正解であれば、該当の知識を既に獲得しており、回答結果が不正解であれば、該当の知識を未だ獲得していないと捉えることができるからである。

問題回答ログデータから作成するデータセットは、下記の要件を満たす必要がある。

1. データセットが大規模であること。
2. 比較検証できるデータセットが複数存在すること。
3. 既存の知識分類が存在すること。

まず、データセットが大規模である必要について説明する。一般に、深層学習は大量のデータを元に特徴的な表現を抽出するため、深層学習モデルを十分に学習させるには、大

規模なデータが必要であり、これは、本研究で用いる、Recurrent Neural Network(RNN)を活用する Deep Knowledge Tracing についても同様である [Piech et al., 2015]。したがって、大規模なデータを有することがデータセットの要件の一つとなる。また、深層学習によって知識獲得の予測を行う本研究においては、単に全体のデータ数が多いだけでなく、分析対象となる個別の問題や生徒について、十分なデータ数を確保できることが重要である。例えば、一度しか回答されていない問題については、その問題の正答や誤答によって、生徒の知識状態がどのように変化するかが観察できないため、分析に適していない。また、十分な数の生徒のデータがないと、特定の生徒の学習傾向が強く反映され、実験結果的一般性が損なわれる可能性がある。

次に、比較検証できるデータセットが複数存在する必要について説明する。本研究で用いる、問題回答ログからなるデータセットは、そのデータセットが提供されるプラットフォームにより、問題を回答している集団や、扱っている教科、内容のレベルなどが異なる。特定のデータセットのみに対して得られた結果は、そのデータセットの環境においてのみ有効である可能性があり、一般性のある結果や知見が得られたとは言いにくい。そのため、本研究では、教科を数学に絞った上で、複数のプラットフォームにおける問題回答ログから複数のデータセットを作成することで、手法の一般性を検証する。なお、数学に限らない、他教科への適用可能性については、第6章で考察する。

最後に、既存の知識分類が存在する必要について説明する。本研究では、現在の一般的な知識獲得予測に用いられている知識分類は、人間の複雑な知識獲得の過程を表現する上で最適化された表現ではない、という仮定に立ち、問題回答ログのみを利用して、より最適化された知識分類を抽出することを目的としている。抽出された知識分類の妥当性を検証するには、既存の知識分類と比較することが必要であり、そのため、分析対象となるデータセットは、既存の知識分類を有する必要がある。

3.3 知識分類の学習と抽出

本節では、提案手法による知識分類の学習と抽出について述べる。本研究では、知識獲得予測を最適化する知識分類を学習するために、問題を知識分類に変換する関数をパラメータ化し、回答正誤予測の最適化の過程で同時に学習する。なお、以降では、この提案手法のモデルを「知識分類学習モデル」と呼ぶ。この知識分類学習モデルの構造は、Deep Knowledge Tracing（以下、DKT）を元に設計されているため、まず、DKTを拡張する方法について述べる。

3.3.1 DKT の拡張による写像関数の学習

問題を知識分類に変換する写像関数をパラメータ化し, Knowledge Tracing の最適化の過程で同時に学習するために, 本研究では, 既存の DKT の構造を拡張した「知識分類学習モデル」を設計する. DKT のモデルを拡張する方法を, 以下の 3 つの要素に分けて説明する.

1. 入力データの粒度
2. モデル全体の構造
3. 最適化手法

入力データの粒度

まず, 既存の DKT と大きく異なる点として, モデルに入力されるデータの粒度の違いについて述べる.

DKT のモデルにおいては, 使用するデータセットは生徒の問題回答ログデータであるが, モデルへの入力は, 事前に定義された知識分類に基づいて, 知識タグに落とし込まれ, どの知識タグが紐づく問題に回答したかが入力される. これは, 既存の知識分類の範囲内で生徒がどのように知識を獲得していくかを予測することを前提にしているためであるが, 本研究においては, そもそも問題と知識分類の関係性を最適化することを目的とするため, このような入力は適さない. よって本研究では, モデルへの入力は問題に対する回答のままにとどめ, 生徒が次にどの問題に正解するかを予測する過程において, 深層学習モデル自身に最適な知識分類方法を判断させ, 学習させる. こうして学習された知識分類は, 結果的に知識獲得の文脈で最適化されている可能性が高く, 既存の知識分類と比較することで, その性能や性質を解釈することができる.

モデル全体の構造

次に, 具体的なモデル全体の構造の拡張について述べる.

まず, DKT では入力が隠れ層へ直接伝達されるのに対し, 知識分類学習モデルは, まず入力層の問題空間 \mathbb{X} から, 抽出目的の知識タグ空間 \mathbb{U} への写像を行う. ここで言う知識タグ空間とは, 既存の知識分類と同じ次元数に設定された空間で, 問題回答の正誤の情報を探次元の空間で表すことを目的としている.

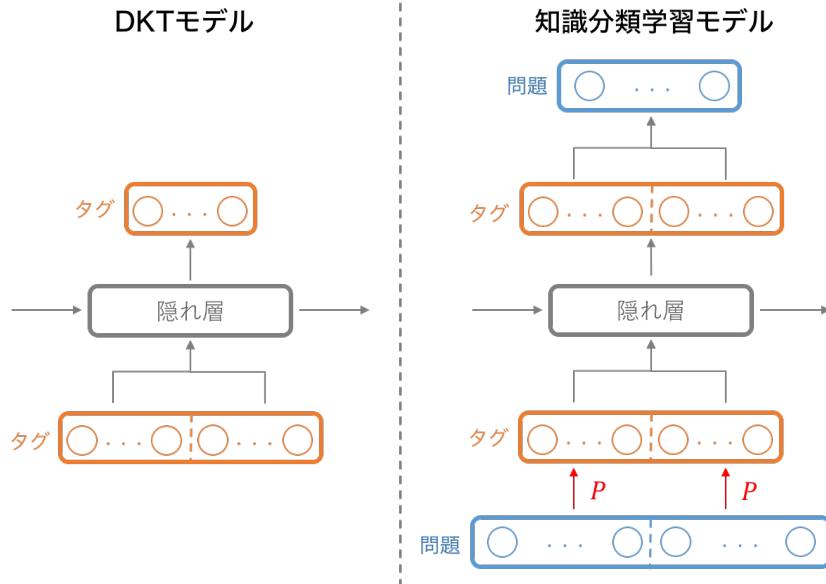


図 3.2: モデル構造上の拡張

具体的には、まず、問題数を M とした場合、正答と誤答を区別するため、モデルへの入力ベクトル \mathbf{x}_t の長さは $2M$ となる。第二層の知識タグ空間の次元数は、既存の知識分類と同じ次元数に揃え、事前に N と定義する。そして、 M 次元の問題空間 \mathbb{X} から N 次元の知識タグ空間 \mathbb{U} へ変換する写像関数 \mathbf{P} を、以下の式により定義する。

$$\mathbf{P}(\mathbf{x}) = \sigma(\mathbf{W}_{xu}\mathbf{x} + \mathbf{b}_u) \quad (3.1)$$

ここで、 \mathbf{x} は長さ M の問題空間ベクトルを指し、 \mathbf{W}_{xu} は $M \times N$ の大きさの重み行列を指し、 \mathbf{b}_u は長さ N のバイアス項を指し、 σ はシグモイド関数を指す。訓練時には、 \mathbf{W}_{xu} 、 \mathbf{b}_u を学習する。

このように定義される写像関数 \mathbf{P} を、 \mathbf{x}_t の前半の正答部分と後半の誤答部分に別々に適用し、連結することによって、正誤の情報を区別したまま、長さ $2N$ の知識タグ空間ベクトル \mathbf{u}_t が生成できる。

$$\mathbf{x}_t = [\mathbf{x}_{t_positive}, \mathbf{x}_{t_negative}] \quad (3.2)$$

$$\mathbf{u}_t = [\mathbf{P}(\mathbf{x}_{t_positive}), \mathbf{P}(\mathbf{x}_{t_negative})] \quad (3.3)$$

ここで、 $\mathbf{x}_{t_positive}$ 、 $\mathbf{x}_{t_negative}$ はそれぞれ問題回答の正答と誤答を表す長さ M のベクトル

である。

こうして得られた知識タグ空間ベクトル \mathbf{u}_t は、一般的な RNN 同様、隠れ層を経由して時系列情報を反映した後、再び長さ $2N$ の知識タグ空間ベクトル \mathbf{v}_t となる。

$$\mathbf{h}_t = \varphi(\mathbf{u}_t, \mathbf{h}_{t-1}, \theta) \quad (3.4)$$

$$\mathbf{v}_t = \sigma(\mathbf{W}_{hv}\mathbf{h}_t + \mathbf{b}_v) \quad (3.5)$$

ここで、 \mathbf{h}_t は時刻 t の隠れ層を指し、 φ は活性化関数を表す。なお、 θ は任意のパラメータを指し、用いる活性化関数によって異なる。訓練時には θ を学習する。この知識タグ空間ベクトル \mathbf{v}_t は、時刻 t までの回答情報を反映した、生徒の知識タグ空間における知識状態を表しているといえる。

最終的に、この知識タグ空間ベクトル \mathbf{v}_t から、予測結果として M 次元の問題回答予測ベクトル \mathbf{y}_t を算出する。

$$\mathbf{y}_t = \sigma(\mathbf{W}_{vy}\mathbf{v}_t + \mathbf{b}_y) \quad (3.6)$$

ここで、 \mathbf{W}_{vy} は重み行列を指し、 \mathbf{b}_y はバイアス項を指す。学習時には \mathbf{W}_{vy} 、 \mathbf{b}_y を学習する。

\mathbf{y}_t は 0 から 1 の間の値を取り、次の時刻 $t+1$ において各問題に正答する確率を表しており、既存の DKT と同様の予測表現となっている。

以上のようなモデル構造上の拡張をまとめた図を図 3.2 に表す。橙色の層は知識タグ空間を、青色の層は問題空間を表し、層が左右で二つに区切られている部分は前半・後半がそれぞれ正答・誤答の情報を表現している。この拡張の目的は、 M 次元の問題空間 \mathbb{X} から N 次元の知識タグ空間 \mathbb{U} へ変換する写像関数 \mathbf{P} をパラメータ化し、回答正誤予測の最適化を行う過程で学習することにある。

最適化手法

最後に、最適化手法の拡張について述べる。

既存の DKT における最適化手法は、時刻 t の出力である問題回答予測ベクトル \mathbf{y}_t と、実際の時刻 $t+1$ の問題回答ベクトル $\tilde{\delta}(\mathbf{q}_{t+1})$ の誤差を損失関数として、これを最小化するものである。 \mathbf{a}_{t+1} を時刻 $t+1$ に対応する問題で正答したか否か (1 か 0) のベクトルと

すれば,

$$\log(p_1 \times p_2 \times \cdots \times p_{m_t}) = \sum_k^{m_t} \log(p_k) \quad (3.7)$$

であることから、損失関数は

$$L_p = \sum_t l(\mathbf{y}_t^T \tilde{\delta}(\mathbf{q}_{t+1}), \mathbf{a}_{t+1}) \quad (3.8)$$

である。この損失関数を、回答正誤予測に関する損失関数 L_p とする。

本研究では、この回答正誤予測に関する損失関数に加え、式 2.38–2.40 で表される再構成誤差も損失関数として導入する。ここで、式 2.38, 2.39 におけるパラメータについては、本研究では、 \mathbf{W} , \mathbf{b} が、入力層で問題空間から知識タグ空間へ写像する際に用いる $\mathbf{W}_{\mathbf{xu}}$, $\mathbf{b}_{\mathbf{u}}$ であり、 \mathbf{W}' , \mathbf{b}' が、出力層で知識タグ空間から問題空間へ復元する際に用いる $\mathbf{W}_{\mathbf{vy}}$, $\mathbf{b}_{\mathbf{y}}$ である。この入力の再構成誤差の損失関数を L_r とすると、結果的に、モデル全体の損失関数 L は以下の式によって定められ、この損失関数を最小化するようにモデルが最適化される。

$$L = L_p + L_r \quad (3.9)$$

ここで、再構成誤差を損失関数に導入する理由について述べる。まず、一般的に再構成誤差を用いる Autoencoder は、深層学習モデルに良い初期値を与えるための事前学習のための仕組みとして用いられるが、本研究では、この Autoencoder の構造を回答正誤予測と同時に学習させるようにモデルに組み込んでいる。Autoencoder の構造を、事前学習ではなく普通の学習モデルに組み込むことは、必ずしも精度向上につながるわけではないため、一般的ではなく、単純に低次元ベクトルへの埋め込み (Embedding) のみに留まることが多い。本研究では、一部、深層学習によって知識分類を学習するには十分とはいえないログ数の問題が存在しており、データの特徴を把握しきれない未学習 (underfitting) の状況に陥る可能性があることを確認した。こうした学習不足への対策として、モデルの学習を矯正し、適切に学習を進めさせる正則化項として再構成誤差を導入している。このように、データ数が不足する場合に、モデルがより適切に学習を進めるように様々な正則化項を設ける手法は一般的であり、有効な手法とされている。学習される知識分類の性質への影響についても検討した結果、Autoencoder の構造は、問題の正答・誤答と知識タグの理解状態は相互に変換できるはずだという教育学的な文脈との整合性と合致し、知識分類

の性質を損ねないと判断したため、適用している。

3.3.2 写像関数の離散化による知識分類の作成

3.3.1 の知識分類学習モデルで得られた写像関数 \mathbf{P} から、実際に知識分類を作成し、既存の知識分類と比較する手法について説明する。

回答正誤予測の最適化の過程で得られた写像関数 \mathbf{P} は、 M 次元の問題空間を N 次元の知識タグ空間に写像する、 $M \times N$ の大きさの行列として表現されている。この行列は 0 から 1 の値を取る連続値表現であるため、そのままでは問題がどの知識タグに紐づくかを特定することはできない。そのため、何らかの方法で、この行列を 0 か 1 の 2 値を取る離散表現に改める必要がある。

離散化の方法としては、各問題において最も大きいタグのみを知識タグとする方法や、行列全体で特定の閾値を定め、その閾値を超えたものを知識タグとする方法、両者を組み合わせる方法など、様々な方法が考えられる。本研究では、知識獲得予測に適用した際に最も良い精度で予測できる分類を、知識獲得の過程を最もよく説明する分類として見なし利用し、その性質も解釈する。まず、各手法において得られた写像行列を、以下の条件に基づいて離散化し、DKTにおいて最も高い精度を発揮したものをその手法による精度の上界として採用する。

1. 各問題の写像ベクトルにおいて、最も値が大きいタグを 1 とする
2. 写像行列全体において、値が閾値 Y 以上のタグを 1 とする
3. 写像行列全体において、1 でない要素を 0 にする

3.4 知識分類の予測性能の検証

次に、抽出された知識分類(以下、抽出タグ)の知識獲得予測における性能を、比較検証する手法について述べる。まず、抽出タグが知識獲得の予測性を最適化する表現となっていることを示すために、抽出タグに基づいた回答ベクトルを一般的な DKT に入力し、既存の知識タグ(以下、既存タグ)を用いた場合よりも精度が向上することを確認する。また、この精度向上が、回答正誤予測の最適化の過程で知識分類を作成したことに起因することを示すため、回答正誤予測と無関係の、一般的な事前学習の Autoencoder で作成した知識タグを用いた場合とも比較を行う。これは、本手法による知識分類の作成が、

- 人間の可読性を目的として人間によって設計された分類か否か
- 知識獲得予測の文脈で最適化された分類か否か

という二つの観点において新規性を有することを受け、性能の変化が何に起因してもたらされたのかという、差分をより明確にするために行う検証である。

3.5 知識分類の性質の比較

最後に、こうした知識獲得予測の精度向上という定量的な検証に加え、抽出タグの性質について、既存タグとの比較を通して、定量的・定性的な分析を行う。これは、知識獲得予測において最適化されている知識分類の性質を解釈し、知見を得ることで、教育における実用性を考察することにつなげることが目的である。

まず、抽出タグと既存タグについて、回答ログに出現する回数の分布に着目し、知識獲得予測の精度を向上させる要因を、データ構造という側面から分析する。また、抽出タグと既存タグが、内容の面においてどのような関係性があり、抽出タグがどのような問題をカバーするタグとなっているのかを分析する。

抽出タグと既存タグの内容を比較する方法として、まず、抽出タグが紐づく問題と、既存タグが紐づく問題から、抽出タグと既存タグの共起行列を作成する。この行列は、各抽出タグと既存タグとの関係性の近さを表現した行列と捉えることができるが、各抽出タグの特徴をより明確に捉るために、TF-IDF法を用いて、特徴の重み付けを行う。TF-IDF法は、元々文書の分類に用いられる手法で、複数の文書がある時に、各文書を特徴づける単語を特定することを目的にしている。具体的には、まず、各文書内での、各単語の出現頻度(Term Frequency, 以下 TF)を求める。 TF は、文書内に多く出現する単語ほど重要だ、という考えに基づいた指標であるが、 TF のみだと、例えば助詞や助動詞といった、いくつもの文書で横断的に使われている単語の重要度が高くなってしまうため、各単語が全文書の内いくつの文書にあらわれているかという制約 (Inverse Document Frequency, 以下 IDF) を設けて、一般的な単語の影響を除外する。単語 i の文書 j における $TF_{i,j}$ 、単語 i の IDF_i と、それらを用いた TF-IDF 値 ($TFIDF_{i,j}$) は以下の式で表される。

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (3.10)$$

$$IDF_i = \log \frac{|D|}{|\{d : d \ni t_i\}|} \quad (3.11)$$

$$TFIDF_{i,j} = TF_{i,j} \cdot IDF_i \quad (3.12)$$

$n_{i,j}$ は単語 i の文書 j における出現回数を指し、 $\sum_k n_{k,j}$ は文書 j におけるすべての単語の出現回数の和を指し、 $|D|$ は総文書数を指し、 $|\{d : d \ni t_i\}|$ は単語 i を含む文書数を指す。

さて、今回の抽出タグの特徴付けにおいては、各既存タグの成分が単語にあたり、抽出タグ1つ1つが文書にあたる。複数の抽出タグに共通して現れる既存タグの成分ほど、特徴づけにおける重みが小さくなり、特定の抽出タグのみに現れる既存タグの成分ほど、特徴づけにおける重みが大きくなる。この手法により各抽出タグの特徴付けを行い、各抽出タグにおいて特徴的な既存タグの成分を強調した共起行列（以下、タグ関係行列）を作成する。

次に、既存タグをノードとする知識間影響ネットワーク（以下、既存タグネットワーク）を、既存のDKTで用いられた手法に基づいて作成し、タグ関係行列を元に抽出タグのノードを既存タグネットワークに追加することで、両タグの関係性を表すネットワーク（以下、タグ関係ネットワーク）を作成する。こうして得られたタグ関係ネットワークは、既存タグを知識間の影響に基づいて配置した上で、抽出タグとの関係性を俯瞰することが可能であり、このネットワークを元に両タグの構造や内容的な関係性を考察する。

以上の処理の流れを図3.3にまとめた。

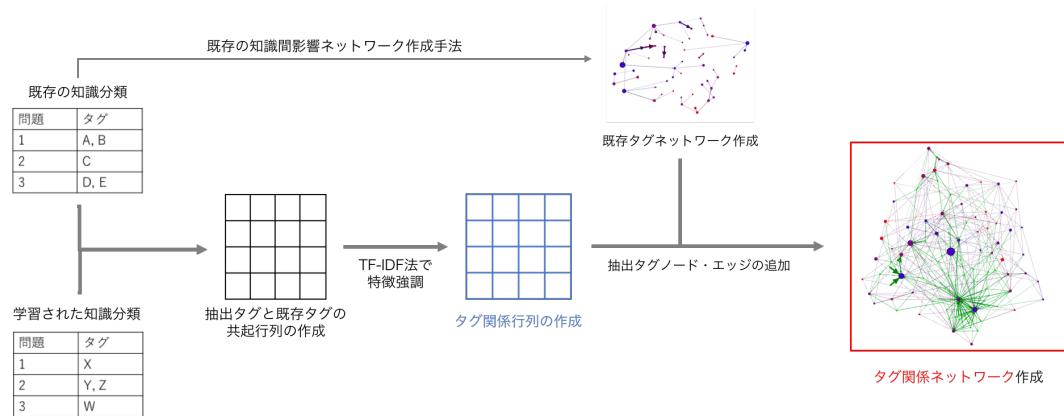


図 3.3: ネットワーク作成の流れ

以上、分析手法について述べた。次章では、実験で利用するデータセットについて述べる。

第4章 データセット

本章では、実験で用いるデータセットについて述べる。

本研究では、オンライン教育サービスにおける、生徒の問題回答ログをデータとして用いる。その際、比較検証のため、教科を数学に絞った上で、2つのデータセットを用意する。いずれのデータセットも、前章で述べたデータセットの要件の一つである、既存の知識分類を有するという要件を満たしている。以下では、各データセットについて概説した後、本研究に適用するためのデータの抽出方法について述べる。

4.1 ASSISTments 2009-2010

本データセットは、オンライン学習サービスの「ASSISTments¹」における、生徒の問題回答ログから生成されている。まず、ASSISTmentsのサービスについて概説した後、本研究で用いるデータセットについて説明する。その際、本研究に適用する際に問題となるデータの性質に言及した上で、その問題点を解消するためのデータの抽出方法を述べる。

4.1.1 ASSISTments のサービス

ASSISTments は Intelligent Tutoring System(ITS) の一つで、2017 年 1 月現在で、14 の国と 42 の州において利用されている。数学や科学、英語や社会と言った科目をカバーしており、レベルは日本における小学校程度のものから高校程度のものまで様々である。基本的な仕組みは、システムが生徒に課した問題を生徒が回答し、その結果をシステムが自動的に採点、間違えた場合はヒントを出すなどして、生徒の知識獲得を促すもので、教師や親がその回答結果や統計情報から生徒の習熟度を確認できること、また、必要があれば、システム上で提供されている教材を自由に編集して、新たな問題を作成できる柔軟性の高さなどから、様々な教育機関やオンライン教育サービス上で活用されている。

¹<https://www.assIstments.org/>

4.1.2 対象データセット

本研究で用いるデータセットは、「ASSISTments 2009-2010」と呼ばれる、ASSISTmentsにおける、生徒の2009年から2010年の間の数学の問題回答ログの内、「skill_builder」²と呼ばれるデータセットである。

元々、「ASSISTments 2009-2010」には「skill_builder」と「non_skill_builder」という、系統の異なる2つのデータセットが含まれている。「skill_builder」は、生徒に知識を段階的に身につかせることを目的にした系統で、ある知識を問う問題に生徒が連続で正答できた場合に該当の知識を習得したものとみなし、次に進ませるというものである。日本の教育現場で言えば、授業ごとの小テストに近いものといえる。一方、「non_skill_builder」は、生徒がそれまで学んできたことを正しく身につけられているかを確認することを目的にした系統で、さまざまな知識を問う問題を、まとめて生徒に課すものである。日本の教育現場で言えば、期末テストに近いものといえる。

このような性質から、「skill_builder」のデータセットの方が、生徒の知識獲得過程の細かな推移を観察する上で適しているため、Deep Knowledge Tracing [Piech et al., 2015]を始めとする多くのKnowledge Tracingの研究で利用されるデータセットであり、本研究でも同様に、「skill_builder」のデータセットを用いる。生徒が解く問題(problem)には、1つ以上の知識タグ(skill)が紐付いている。「skill_builder」のデータセットには、4,217人の生徒の、124の知識タグが紐づく26,688の問題に対する、401,756の回答ログが含まれている。なお、「skill_builder」のデータセットは、行の重複などによって大幅な不備が指摘されたため訂正版が提供されており、それ以前の研究結果は信憑性が低い。本研究では、訂正後のデータセットを用いている。

4.1.3 データの抽出

本データセットは、本研究に適用する上で以下の3つの問題を抱えている。順に、各問題と対策について述べる。

まず、問題(problem)の中には複数の知識タグ(skill)が紐付いているものがあるが、そうした問題が回答された場合には、知識タグの数だけ、ログが別々に作成されている。これは、見かけ上のログ数(401,756)が、実際に回答された回数より多くなっているだけでなく、同時に回答された問題や知識タグが、別々のタイミングで回答されたとみなされる

²<https://sites.google.com/site/assistmentsdata/home/assitment-2009-2010-data/skill-builder-data-2009-2010>

危険性があり、この前提を考慮しない Knowledge Tracing は不適切であることが指摘されている [Xiong et al., 2016b]. このため、同時回答を意味する重複ログを一つにまとめる作業が必要である。

次に、既存の知識タグ (skill) についても、存在はするものの、名前が割り当てられていないものが存在し、これらは抽出された知識タグと比較することが不可能なため、除外する必要がある。

また、本データセットは、全体で見ると十分大規模であるといえるが、個別の問題に関して言えば、ほとんど回答されていない問題が、全体に対して大きな割合を占めている (図 4.1a を参照). 十分なログ数を保有しない問題は、大規模データから深層学習によって知識分類を学習するという本研究の目的を満たさないため、ログ数について一定の閾値を設けてデータセットを切り分けることにより、適切なデータを抽出する必要がある。

以上より、本研究では、元のデータセットから以下の方法で分析対象とするデータを抽出している。

1. 同時回答を意味する重複ログを一つにまとめる.
2. 名前が割り当てられている知識タグを持つ問題の回答ログのみを抽出する.
3. 2 のうち、最低 30 回以上回答されている問題の回答ログのみを抽出する.
4. 3 に含まれる問題を、最低 2 回以上回答している生徒に関するログを抽出する.

なお、3 の 30 回以上という具体的な数字は、深層学習を適用して有意な結果が得られるログ数として実験的に得たものであり、網羅的に検証されたものではない。4 の 2 回以上という数字は、各生徒の知識獲得の推移を観察する上で最低限必要なログ数である。結果的に、3,410 人の、55 の知識タグが紐づく 2,635 の問題に対する、129,317 の回答ログが分析対象である。

4.2 Bridge to Algebra 2006-2007

本データセットが利用された「KDDCup」について概説した後、データセット自体について説明する。その際、ASSISTments 同様、本研究に適用する際に問題となるデータの性質に言及した上で、その問題点を解消するためのデータの抽出方法を述べる。

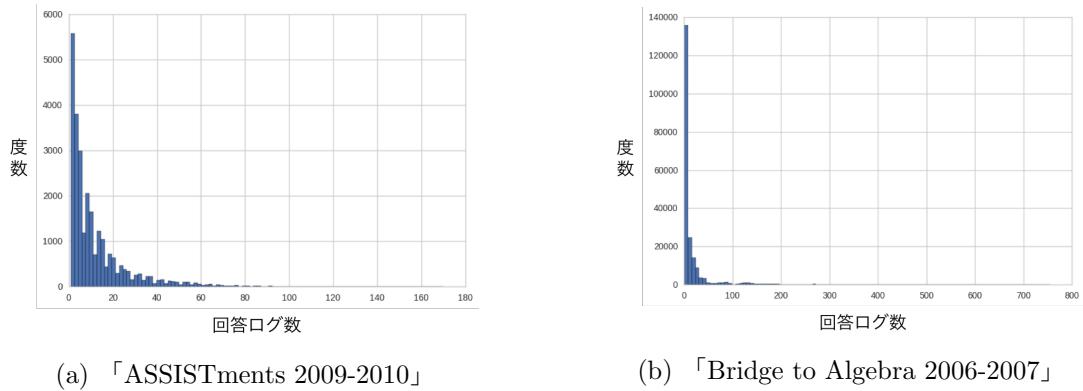


図 4.1: 問題ごとの回答数の分布

4.2.1 KDDCup

「KDDCup (Knowledge Discovery and Data Mining Cup)³」は、100 以上の国にまたがり、10 万人を超える会員を持つコンピューターサイエンス分野の学会である「ACM(the Association for Computing Machinery)」の分科会である「SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining)」が毎年開催する競技会であり、この分野で最も古く権威のある競技会の一つである。

4.2.2 対象データセット

本研究では、2010 年に開催された KDDCup の内の一つの、教育分野の競技会である「Educational Data Mining Challenge」で使用された、「Bridge to Algebra 2006-2007」[Stamper et al., 2010] というデータセットを用いる。これは、オンライン教育サービスの「Carnegie Learning⁴」が提供するオンライン学習支援システム「Cognitive Tutor」における、2006 年から 2007 年の間の、数学の問題に対する生徒の問題回答ログである。Cognitive Tutor は ASSISTments と同じく ITS だが、やや性質の異なるものになっている。ASSISTments は、生徒が毎日の宿題を解く過程をサポートするような、比較的単純な設計となっている一方で、Cognitive Tutor は、より生徒が個別の知識を獲得する過程を緻密にサポートする設計となっている。具体的には、各問題 (problem) が、複数のステップ (step) に分解されており、ステップ 1 つ 1 つに知識タグ (knowledge component) が紐付いている。このデータセットには、1,146 人の生徒の、494 の知識タグが紐づく 19,186 の問題と 19,766 の

³<http://www.kdd.org/kdd-cup>

⁴<https://www.carnegielearning.com/>

ステップに対する、3,679,199の回答ログが含まれている。

4.2.3 データの抽出

本データセットも、「ASSISTments 2009-2010」同様に、本研究に適用する上で3つの問題を抱えている。以下に各問題と対策を述べる。

まず、問題(problem)やステップ(step)という粒度が存在する本データセットにおいて、何を一回の問題回答と見なししてデータを作成するかが問題となる。Cognitive Tutorでは、一つの問題内に複数のステップが用意されており、各ステップについて逐次回答して正答できるまで取り組み、正答できた場合に次のステップに進むようになっている。そのため、生徒の知識獲得の推移を観察する上では、一つ一つのステップに注目することが適切だといえる。よって、本データセットでは、問題内のステップに対する回答を一回の問題回答と見なし、データを抽出する。

次に、既存の知識タグ(knowledge concept)についても、存在はするものの、名前が割り当てられていないものが存在し、これらは抽出された知識タグと比較することが不可能なため、除外する必要がある。

また、本データセットは、全体で見ると十分大規模であるといえるが、個別の問題に関して言えば、ほとんど回答されていない問題が、全体に対して大きな割合を占めている(図4.1bを参照)。十分なログ数を保有しない問題は、大規模データから深層学習によって知識分類を学習するという本研究の目的を満たさないため、ログ数について一定の閾値を設けてデータセットを切り分けることにより、適切なデータを抽出する必要がある。

以上より、本研究では、元のデータセットから、以下の方法で分析対象とするデータを抽出している。

1. 問題(problem)とステップ(step)の組み合わせを一回の問題回答とみなす。
2. 名前が割り当てられている知識タグを持つ問題の回答ログのみを抽出する。
3. 2のうち、最低200回以上回答されている問題の回答ログのみを抽出する。
4. 3に含まれる問題を、最低2回以上回答している生徒に関するログを抽出する。

なお、3の200回以上という具体的な数字は、深層学習を適用して有意な結果が得られるログ数として実験的に得たものであり、網羅的に検証されたものではない。4の2回以上という数字は、各生徒の知識獲得の推移を観察する上で最低限必要なログ数である。結

果的に、1,124人の、115の知識タグが紐づく618のステップに対する、227,612の回答ログが分析対象である。

4.3 データセットの概観

以上の条件から抽出された、本実験に用いるデータセットの統計量を表4.1に示す。

表4.1: 各データセットの統計量

| データセット名 | 生徒数 | 問題数 | 知識タグ数 | ログ数 |
|-----------------------------|-------|-------|-------|---------|
| ASSISTments 2009-2010 | 3,410 | 2,635 | 55 | 129,317 |
| Bridge to Algebra 2006-2007 | 1,124 | 618 | 115 | 227,612 |

なお、「Bridge to Algebra 2006-2007」は「ASSISTments 2009-2010」に比べて、問題数に対する知識タグの数の割合が大きく異なっているが、これは「Bridge to Algebra 2006-2007」では、問題がさらにステップに分割されており、知識タグの粒度がより細かくなっているためである。

以上、データセットについて述べた。次章では、実験について述べる。

第5章 実験

本章では、実験について述べる。

まず、実験設定について述べ、その後、実験結果について述べる。

5.1 実験設定

本研究の実験は、大きく以下の3つに分けられる。

1. 知識分類の学習と抽出
2. 知識分類の予測性能の検証
3. 知識分類の性質の比較

以下では、順に、実験設定について述べる。

5.1.1 知識分類の学習と抽出

生徒の問題回答ログに対し、図3.2に表される知識分類学習モデルを適用し、問題空間を知識空間に最適に変換する写像関数を学習し、離散化してタグとして抽出する。

知識タグ空間の次元数は、既存の知識タグの次元数と統一し、「ASSISTments 2009-2010」では55、「Bridge to Algebra 2006-2007」では115とした。実際のモデルにおいては、正答ベクトルと誤答ベクトルを分けてユニットを作るため、それぞれ2倍のユニット数で表現されている。

RNNの部分にはGRNNを用いる。ハイパーパラメータについては、学習率の初期値を200、モーメントを0.98、1エポックごとに、減衰率0.99として学習率を最小学習率10まで減衰させる。また、勾配のノルムの最大値を0.00001として[Pascanu et al., 2013]に従い勾配に制約を設けた。dropoutは[Piech et al., 2015]と同様に y_t の方向にのみかけ、dropout率は0.5とした。隠れ層のユニット数は400として、各重み行列の初期化は[Glorot

and Bengio, 2010] にしたがつた。時系列方向の誤差逆伝搬は最長で 200 まで伝搬するよう制約を設けた。

これらのハイパーパラメータは実験的に高い予測性能を発揮したため設定しており、網羅的に探索したわけではない。通常、深層学習の手法はハイパーパラメータの数が非常に大きく、また、計算コストが大きいため大規模な探索は行えない。Grid Search や Random Search [Bergstra and Bengio, 2012] といった探索手法が提案されているが、専門家が手で調整した方が優れていることが報告されている [Larochelle et al., 2007, Bergstra and Bengio, 2012]。

最適化手法は、式 3.8 で表される問題回答予測に関する誤差関数 L_p と、式 2.40 で表される問題空間と知識タグ空間の最高性誤差を表す誤差関数 L_r の和である L (式 3.9) を目的関数として最小化するものである。学習時は [Piech et al., 2015] と同様にミニバッチごとに確率的勾配降下法で目的関数を最小化する。評価指標は AUC スコアを採用する。

一般的な機械学習では、モデルの汎化性能を高めるために訓練データと検証データを分けるが、2つのデータセットいずれにおいても、全ユーザの問題回答ログを訓練データとして、モデルを学習させた。この設計と知識分類の汎用性に関しては第 6 章で詳述する。

得られた写像関数を以下の条件に基づいて離散化し、タグとして抽出する。「ASSISTments 2009-2010」では $Y = 0.70$, 「Bridge to Algebra 2006-2007」では $Y = 0.85$ とした。

1. 各問題の写像ベクトルにおいて、最も値が大きいタグを 1 とする。
2. 写像行列全体において、値が閾値 Y 以上のタグを 1 とする。
3. 写像行列全体において、1 でない要素を 0 にする。

実装には Theano [Bergstra et al., 2010, Bastien et al., 2012] を用いた。Theano は多次元行列を含む数学的表現の定義や計算、最適化を効率的に行える Python のライブラリで、深層学習の研究ではよく利用される。

5.1.2 知識分類の予測性能の検証

5.1.1 で抽出した知識分類を Deep Knowledge Tracing(DKT) に用いて知識獲得予測を行い、既存の知識分類を用いた場合との精度の比較を行う。また、本手法で抽出される知識分類と既存の知識分類の差分を明確にするため、以下の方法で作成された知識分類を用いた場合とも比較を行う。

- 回答正誤予測の文脈を考慮せず、一般的な事前学習の Autoencoder によって作成された知識分類
- 回答正誤予測の文脈を考慮するが、再構成誤差を目的関数に導入しない、一般的な埋め込み (Embedding) によって作成された知識分類

RNN の部分には GRNN を用いる。ハイパーパラメータについては、学習率の初期値を 200、モーメントを 0.98、1 エポックごとに、減衰率 0.95 として学習率を最小学習率 10 まで減衰させる。また、勾配のノルムの最大値を 0.00001 として [Pascanu et al., 2013] に従い勾配に制約を設けた。dropout は [Piech et al., 2015] と同様に y_t の方向にのみかけ、dropout 率は 0.5 とした。隠れ層のユニット数は 400 として、各重み行列の初期化は [Glorot and Bengio, 2010] にしたがった。時系列方向の誤差逆伝搬は最長で 200 まで伝搬するように制約を設けた。

最適化手法は、一般的な DKT と同じく、式 3.8 で表される回答正誤予測に関する誤差関数 L_p を目的関数として最小化するものである。学習時は [Piech et al., 2015] と同様にミニバッチごとに確率的勾配降下法で目的関数を最小化する。評価指標は AUC スコアを採用する。

2つのデータセットいずれにおいても、訓練：検証：テスト = 8:1:1 となるようにユーザを分け、訓練ユーザのデータでモデルを構築し、検証ユーザのデータでハイパーパラメータを調整し、検証ユーザのデータで精度が最も高かったモデルをテストユーザのデータに適用し当該モデルの最終的な精度とした。

実装には Theano を用いた。

5.1.3 知識分類の性質の比較

5.1.1 で抽出された知識分類を既存の知識分類と比較分析することで、その性質を検証する。

まず、既存の知識タグ（以下、既存タグ）と抽出された知識タグ（以下、抽出タグ）それぞれにおいて、各タグが回答ログに出現する回数の分布に着目し、知識獲得予測の精度を向上させる要因を、データ構造の側面から定量的に分析する。

次に、抽出タグと既存タグの関係性を可視化し、その構造や内容について比較を行う。まず、既存タグをノードとする知識間影響ネットワーク（以下、既存タグネットワーク）を、既存の DKT で用いられた手法に基づいて作成し、次に、抽出タグと既存タグの共起行列に対して TF-IDF 法を適用して各抽出タグの特徴を強調し、抽出タグと既存タグの関

係性を表す行列(以下、タグ関係行列)を作成し、この行列を元に抽出タグのノードを既存タグネットワークに追加することで、両タグの関係性を表す「タグ関係ネットワーク」を作成する。ここで、既存タグネットワークでは、ノードのサイズは、各既存タグの回答ログにおける出現回数に比例して設定し、ノードの色は、回答ログにおける平均回答順序が早いものほど青く、遅いものほど赤く色を設定し、各既存タグへの影響度が大きい上位3つの既存タグからエッジを引く。また、タグ関係ネットワークにおいて追加される抽出タグは緑色のノードで表現し、各抽出タグにとって関係性の強い上位3つの既存タグに対して緑色のエッジを引く。

以上、実験設定について述べた。

5.2 実験結果

実験結果について述べる。まず、各手法によって作成された知識分類についての知識獲得予測における予測性能を比較し、いずれのデータセットにおいても、提案手法によって学習された知識分類が最も良い精度を記録したことを定量的に確認する。

さらに、学習された知識分類を、既存の知識分類と比較することにより、その性質を定量的・定性的に分析する。

5.2.1 知識分類の予測性能の比較

表 5.1: 各知識タグの知識獲得予測における予測性能

| データセット | AUC | | | |
|-----------------------------|-----------------|--------|-----------|-------------|
| | 既存タグ (marginal) | 事前学習タグ | 深層学習タグ | |
| | | | Embedding | AutoEncoder |
| ASSISTments 2009-2010 | 0.72 (0.61) | 0.67 | 0.69 | 0.74 |
| Bridge to Algebra 2006-2007 | 0.81 (0.72) | 0.79 | 0.81 | 0.82 |

ベースラインとなる既存の知識タグ(既存タグ)と、回答正誤予測の文脈を考慮しない、一般的な事前学習のAutoencoderによって作成された知識タグ(事前学習タグ)、回答正誤予測の文脈を考慮し、提案手法の知識分類学習モデルによって作成された知識タグ(深層学習タグ)をそれぞれ Deep Knowledge Tracing に適用した結果を表2に示す。深層学習タグの「Embedding」と「Autoencoder」はそれぞれ誤差関数に再構成誤差を導入しな

い場合と導入した場合である。marginalは各問題についてそれぞれ正解の周辺確率を予測結果とするものである。[Piech et al., 2015]にも記載されていたため、本稿でも同様にベースラインの参考として記載した。また、値が大きい箇所は太字で記載した。

いずれのデータセットにおいても、提案手法である「深層学習タグ(Autoencoder)」が、最も高いAUCを記録した。この結果より、提案手法によって作成された知識分類が、既存の知識分類よりも知識獲得の予測性において優れていることが示された。以下、このタグを「抽出タグ」とする。

5.2.2 抽出タグと既存タグの関係の概観

抽出タグと既存タグの関係を概観する。既存タグの知識間影響ネットワークである既存タグネットワークを図5.1に、抽出タグと既存タグの関係性を表すタグ関係ネットワークを図5.2に表す。なお、タグ関係ネットワークにおいて接続している抽出タグと既存タグの組み合わせは付録にまとめた。

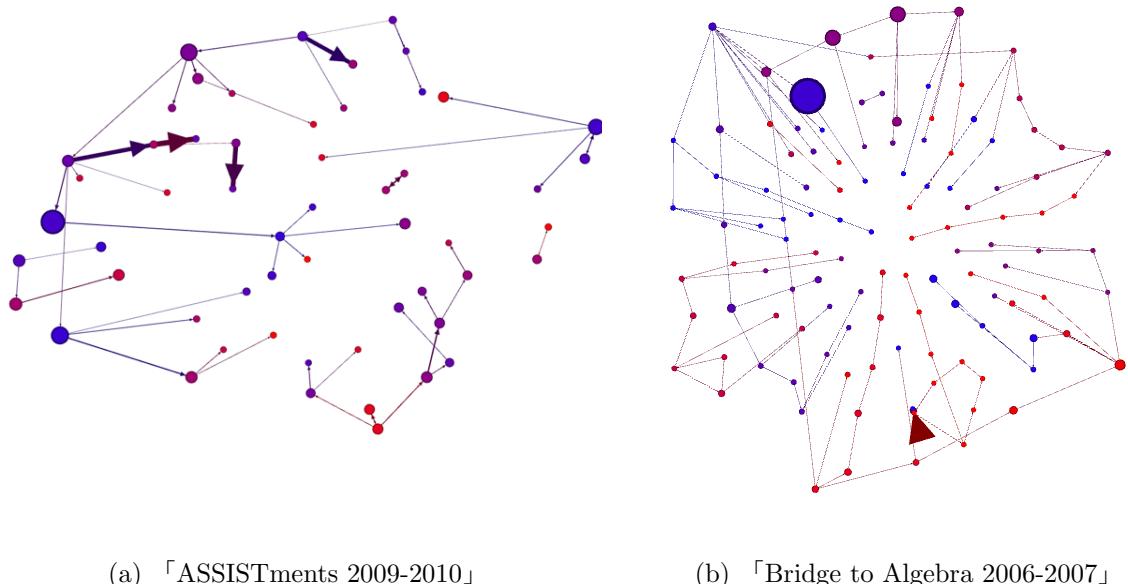


図 5.1: 既存タグネットワークの構造

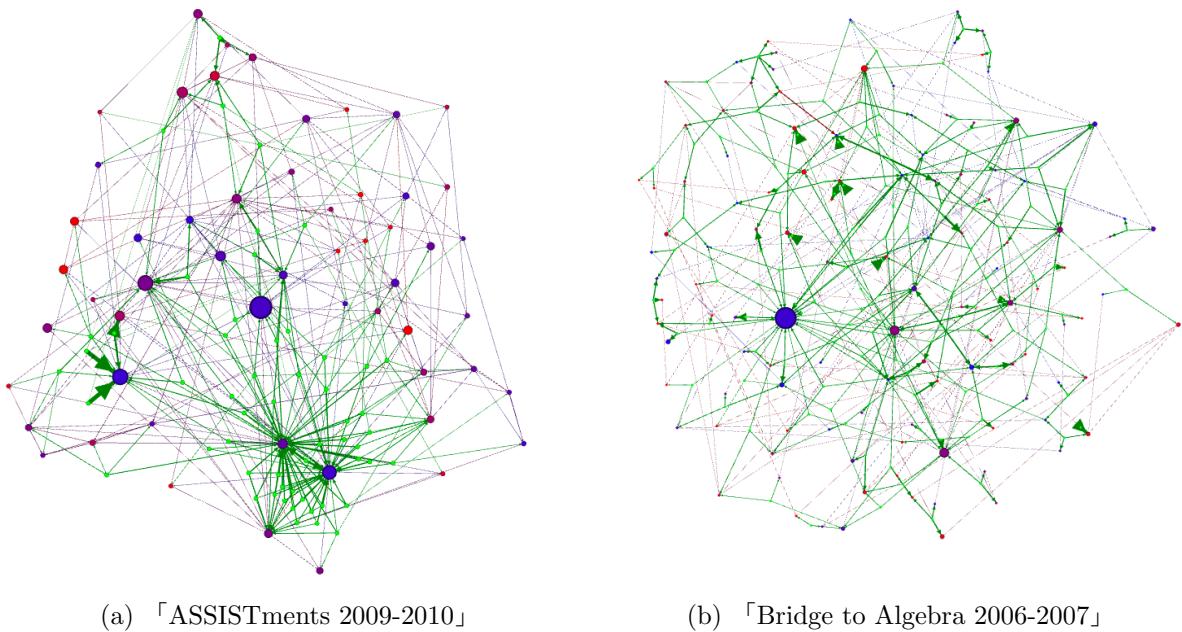


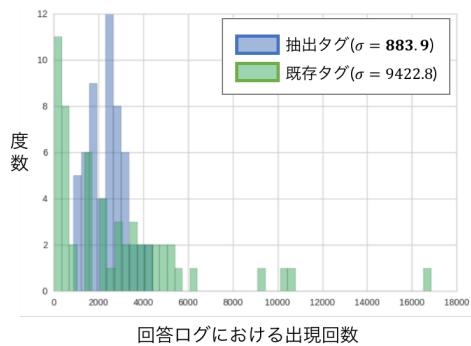
図 5.2: タグ関係ネットワークの構造

5.2.3 抽出タグと既存タグの比較分析

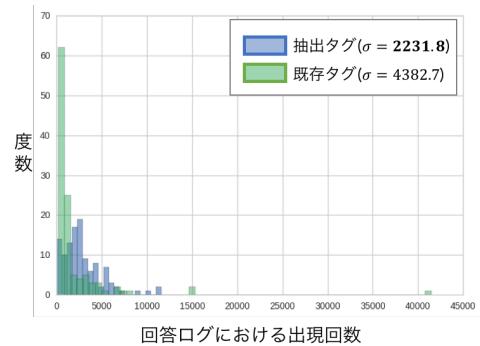
抽出タグを既存タグと比較することにより、抽出タグの性質を定量的・定性的に分析する。

まず、知識獲得予測の精度を向上させる要因を、データの構造から分析した。抽出タグと既存タグそれぞれにおいて、各タグが回答ログに出現する回数の分布の比較と、各分布の標準偏差 σ を図 5.3 に表す。図より、既存タグはタグごとの出現回数の分散が大きい一方で、抽出タグは分散が小さく、特定の値の周辺に集中していることがわかる。この分布の違いと予測精度の関係性については、第 6 章で考察する。

次に、図 5.2 のネットワークの局所的な特性に着目し、抽出タグと既存タグの構造や内容の関係性が観測できる部分を示す。まず、既存タグに注目し、各既存タグに抽出タグがどのように紐付いているかを観察すると、出現回数の多い既存タグ(大きいノード)は多くの抽出タグ(緑のノード)が紐付いている(図 5.4)一方、出現回数の少ない既存タグ(小さいノード)は特定の抽出タグのみ紐付き、多くは紐付いていない(図 5.5)ことがわかる。

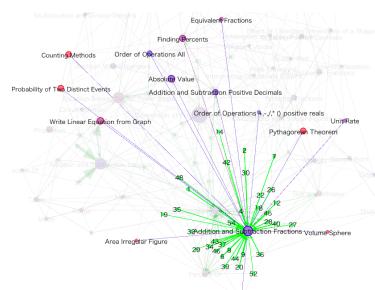


(a) 「ASSISTments 2009-2010」

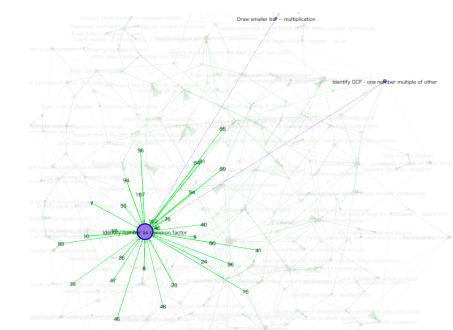


(b) 「Bridge to Algebra 2006-2007」

図 5.3: 各タグの出現回数の分布

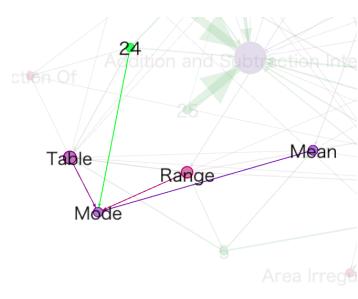


(a) 「ASSISTments 2009-2010」

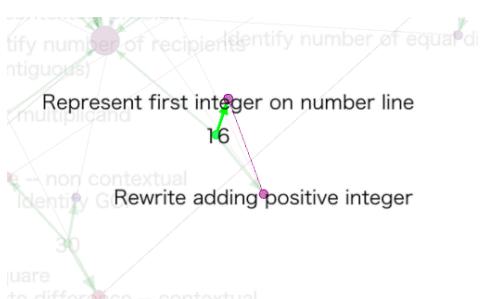


(b) 「Bridge to Algebra 2006-2007」

図 5.4: 多くの抽出タグが紐づく既存タグ



(a) 「ASSISTments 2009-2010」



(b) 「Bridge to Algebra 2006-2007」

図 5.5: 少数の抽出タグのみ紐づく既存タグ

次に、抽出タグに注目し、各抽出タグがどのような既存タグに紐付いているかを観察すると、内容的な関係性の強い複数の既存タグに紐付いている抽出タグが存在する（図5.6）

ことがわかる。このような既存タグと抽出タグの関係性から、どのような性質のタグが抽出されているといえるかは、第6章で考察する。

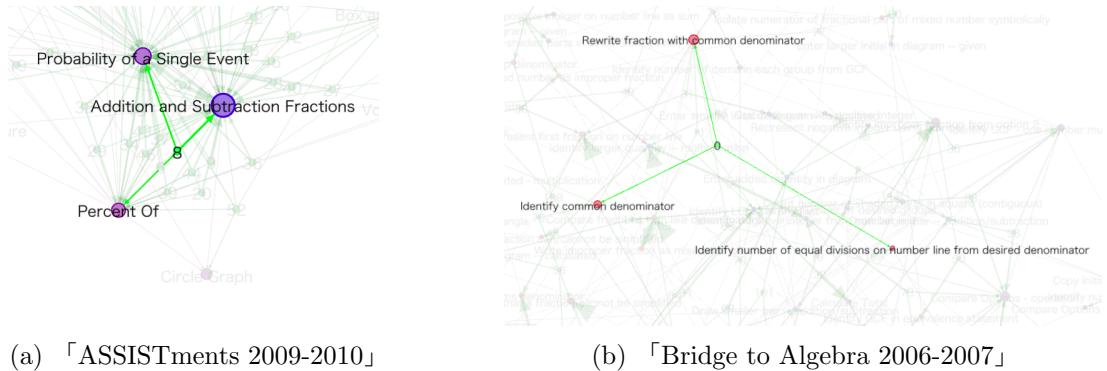


図 5.6: 内容的関係性の強い既存タグに紐づく抽出タグ

以上、実験について述べた。次章では、考察について述べる。

第6章 考察

本章では、実験結果を踏まえた考察を述べる。

まず、知識分類の予測性能の比較実験の結果から、本研究で用いた知識分類学習モデルの有効性について考察する。次に、同モデルにより抽出された知識タグと既存の知識タグを比較し、性質の違いとそれが知識獲得予測に与える影響について考察する。また、本研究の手法や、関連研究が対象としたデータの範囲から、本研究の手法の汎用性や他のデータへの適用可能性、成果の実用性について考察する。

最後に、今後の展望として、より良質な知識分類を得るためにモデルの改善案について述べた後、本研究で用いた手法の学習科学での適用の可能性について述べ、最後に、学習科学以外の分野への応用可能性について述べる。

6.1 本手法の有効性と得られた知識分類の解釈

まず、本手法で用いた知識分類学習モデルと、それによって得られた知識分類について解釈を行う。

6.1.1 知識分類学習モデルの有効性

知識獲得予測の性能の比較実験の結果から、本研究で用いた知識分類学習モデルの有効性について考察する。

まず、既存の知識タグを利用した場合と、各手法によって抽出したタグを利用した場合の、知識獲得予測の精度に関して考察する。

実験結果より、回答正誤予測の文脈を考慮せず、事前学習の AutoEncoder のみから作成した知識分類を用いた「事前学習タグ」は、既存タグを用いた場合よりも精度が悪かつた一方で、知識分類学習モデルにより、回答正誤予測を行う過程で最適化した「深層学習タグ」は、既存タグよりも精度の良いものがあった。この状況を直接的に解釈すれば、知識分類は、データ構造のみに着目する教師なし学習では、知識獲得予測に有効な表現として学習できないが、知識獲得予測という目的に応じた環境情報を制約に加えて学習するこ

とで、その環境と矛盾しないような知識分類が抽出できたと考えられる。また、より教育学的な文脈を踏まえて解釈を試みれば、問題と知識分類はそれ自体で自明な関係ではなく、問題を回答する生徒の回答正誤や知識獲得の推移という状態の観測を通して定義されることで、適切な分類が可能になるものだと見なすことも可能である。

また、回答正誤予測の文脈を考慮した、知識分類学習モデルを用いた手法においても、単純な低次元空間への埋め込み (Embedding) では精度が向上しないものの、問題空間とタグ空間の再構成誤差を導入することにより、精度が向上した。これは、データ量の不足に対する正則化項の導入という、ニューラルネットワークの文脈における、データの量的側面と、問題の回答正誤と知識タグの理解状態は相互に変換できるはずだという、教育学の文脈における、データの質的側面との、双方の性質を活かす最適化の要素として、再構成誤差が効果を発揮したと考えられる。

6.1.2 各知識分類の性質と知識獲得予測に与える影響

次に、既存タグと抽出タグのそれぞれの性質のち外に着目し、それがどのように知識獲得予測に影響をあたえるのかを考察する。

まず、図 5.3 に表される、既存タグと抽出タグの回答ログにおける出現回数の分布から、既存タグは分散が大きい一方で、抽出タグは分散が小さく、特定の値の周辺に集中していることがわかった。既存タグは、人間の可読性や直感的なわかりやすさを重視して作られており、その知識を問う問題の回答される回数は作成時の評価軸に含まれていない。そのため、基礎的・入門的な問題に関しては多くの生徒に回答される一方、専門的であったり難易度の高い問題に関しては、回答される回数が必然的に少なくなるため、タグ間で出現回数に差が出る。しかし、DKT のモデルに入力される際には、どの知識タグも均等に 1 つのユニットで表現されるため、実際の知識獲得過程において各タグが持つ情報量の偏りを十分に表現できない可能性が高い。一方、抽出タグは、DKT を拡張した知識分類学習モデルで学習されているため、各ユニットが均等に情報量を保つことが可能になり、DKT に適用した場合にも特定のタグに関する情報量が失われることを防いでいると考えられる。

この性質は、タグ関係ネットワーク図にも現れている。図 5.4, 5.5 に表されるように、元々出現回数が少ない専門的な既存タグ (小さいノード) は、少数の抽出タグ (緑のノード) のみで表現されているが、逆に元々出現回数の多い基礎的な既存タグ (大きいノード) は、複数の抽出タグにまたがって表現されるなど、より効率的に情報を保持できるタグ構造がとなっていることがわかる。

さらに、こうした情報量の均等な分配構造は、内容と全く無関係に生成されるものではないこともわかる。図5.6に表されるように、「ASSISTments 2009-2010」では「Probability of a Single Event(一つの事象の確率)」「Percent Of(百分率)」「Addition and Subtraction Fractions(分数の足し算と引き算)」、「Bridge to Algebra 2006-2007」では「Rewrite fraction with common denominator(通分)」「Identify common denominator(公分母の特定)」「Identify number of equal divisions on number line from desired denominator(特定の分母からの数直線の等分割数の特定)」というように、類似した内容の範囲内で情報量の分配を行っている抽出タグが見受けられる。こうしたタグは、情報量を適切に保ちつつ関連知識一般をカバーするようなタグである可能性が高く、人間が知識を獲得していく過程を考察する上で示唆に富んでいる。

6.2 本手法の汎用性と実用性

次に、本研究の手法や、関連研究が対象としたデータの範囲から、本研究の手法の汎用性や他のデータへの適用可能性、成果の実用性について考察する。

6.2.1 学習された知識分類の汎用性

知識分類学習モデルによって学習された知識分類の汎用性について考察する。

一般的な機械学習では、モデルの汎化性能を高めるためにデータを訓練データと検証データに分けるが、本研究の知識分類学習モデルでは、全ユーザの問題回答ログを全て訓練データとして、モデルを学習させた。このようなデータの利用の仕方は、訓練データの分布において最適な結果が得られる一方、未知のデータに対する汎用性が担保されないため、一般的ではない。しかし、本研究においては、知識獲得予測を最適化する知識分類を抽出することを目的とし、その手段として、過去の生徒の知識獲得の過程から、帰納的に最適な知識分類を作成するという形を取っており、これはすなわち、将来の生徒の知識獲得過程は、過去の生徒の知識獲得の過程と同じ分布に従うという前提に立っている。また、オンライン教育サービスにおいては、日々新たな学習回答ログが蓄積されるため、その都度知識分類を更新し、より質の高い分類を作成できるという環境が整っている。以上より、本研究の問題設定においては、これまでの知識獲得過程を全てまとめて訓練データとしても、学習された知識分類の汎用性は損なわれず、また、未知のデータが出現した際にも即座にそれを反映し、より適切な知識分類を作成できると判断したため、知識分類の学習の時点では全データを訓練データとした。なお、学習された知識分類の性能を検証するDKT

の部分においては、既存タグとの精度比較の公平性を期すために、データを訓練・検証・テストに分割している。

一方で、現在観測されているデータと将来観測されているデータの分布が大きく異なる可能性がある場合や、未知のデータが出現した場合に即座にそれを分類に反映させることができ困難な場合には、このようなデータの分割方法は相応しくない。そのような場合には、全データを訓練・検証・テストに分割することで、学習された分類の汎化性能を保証する必要があり、環境や問題設定に応じたデータの扱いや、環境に左右されないモデルの設計が今後の課題といえる。

6.2.2 本手法の他データへの適用可能性

本研究の分析手法の、他の科目やオンライン教育サービスへの適用可能性について考察する。

本研究の手法は、データセット作成という事前の処理と、以下の3つの分析から構成されていた。

1. 知識分類の学習と抽出
2. 知識分類の予測性能の検証
3. 知識分類の性質の比較

データセット作成については、オンライン教育サービスから収集される問題回答ログデータは、サービスや科目によらず大規模であると考えられる。また、実験の「2. 知識分類の予測性能の検証」、および「3. 知識分類の性質の比較」は、知識分類学習モデルによって、適切な知識分類を学習できるかに依存する。よって、本手法の他科目や他サービスへの適用可能性は、「1. 知識分類の学習と抽出」に依存すると考えられる。知識分類学習モデルは DKT を拡張したモデル構造において学習されるため、DKT 自体の他科目や他サービスへの適用可能性によって、本手法の適用可能性も検証されると考えられる。

そこで、DKT の他科目や他サービスにおける予測性能を考察する。

[Piech et al., 2015] では、本研究同様、数学に関するデータセットにおいてのみ、DKT の有効性が検証されていた。那須野ら [参考文献引用] はリクルートが提供するオンライン教育サービス「学習サプリ」のデータを使って、算数や数学に関するデータセットと地理や歴史に関するデータセットに DKT を適用した場合、Bayesian Knowledge Tracing から

の精度向上という点では大きな差はないことを確認しており、DKT の適用可能性は科目に依らない可能性が高い。

また、[Piech et al., 2015] を始めとする既存研究では、モデルへの入力次元には問題に割り当てられたタグが利用されており、DKT の有効性はタグを用いた場合のみ、検証されていた。本研究の実験では、問題回答ログのみから知識分類を学習できるため、既存の知識分類が存在せず、タグ付けができないような科目やサービスのデータに対しても、生徒の知識獲得を予測することが可能であることを示している。

一方で、これまで検証されているのは、特定の科目に関する問題回答ログであり、総合的な知識レベルを問うような、複数の科目が含まれている問題回答ログへの適用可能性は示されていない。

また、利用できるデータセットは、生徒が該当のオンライン教育サービスで学習する過程で、段階的に知識を獲得していく前提のデータのみであり、オンライン教育サービス外での学習や、生徒ごとの能力差、事前知識などの情報に関しては、DKT が扱うことは難しい可能性がある。

以上のような考察を踏まえると、本手法は、DKT が分析可能な他サービスや他科目のデータに加え、DKT による分析が困難な、事前の知識分類が存在しないデータに対しても適用できるという側面がある一方、複数科目のデータや生徒に関する事前情報など、現実に即した複雑な情報が多く含まれたデータに対しては、適用可能性が限定的である可能性もあり、検証が必要である。

6.2.3 教育現場への適用

ここまで考察を踏まえ、本研究の成果を、実際の教育現場に適用し、活用する方法について考察する。

そもそも知識獲得の予測は、オンライン教育サービスにおいて、問題を正答するのに必要な知識を生徒が既に獲得しているかを推定することで、不足している知識を補ったり、既に獲得している知識を除外したりと、適切な順序で教材推薦を行うことが主な目的であった。

実験結果から、本手法によって抽出された知識分類を利用することにより、知識獲得の予測精度が向上することが確認された。知識獲得の予測精度が向上するということは、各生徒の知識状態をより的確に把握して、教材推薦の精度を向上させることを意味する。よって、現在オンライン教育サービス上で提供されている問題に対して、既存の可読性重視の

知識分類に加え、本研究で抽出された知識分類を紐付けておくことで、教材推薦の精度を向上させ、生徒個人個人への教材の最適化が進み、生徒の学習効率をより高めることができる。この知識分類の粒度は、本研究では既存の知識分類との比較のために固定していたが、自由に設定することが可能なため、サービスごとに適切な粒度を設定することが可能である。また、本手法は人間による事前の知識分類を必要としないため、人間によって問題が事前に分類されていないようなサービスに対しても、自動で最適な知識分類を作成して知識獲得を予測し、生徒ごとに最適な教材を推薦することが可能になる。

このように、知識獲得の予測性を最適化する知識分類を抽出する本手法は、オンライン教育サービスの教材推薦システムに組み込むことで、より生徒の学習効率を高めができる上、適切な知識分類が存在しないサービスにおいても知識獲得予測を可能にするものであり、教育現場における実用上の価値は高いといえる。

6.3 今後の展望

本研究の今後の展望について大きく3つの方針を述べる。まず、本手法をさらに改善し、より良質な知識分類を学習できるようなモデル構造の可能性について述べ、次に、学習科学における対象データの拡大について述べ、最後に、学習科学以外の分野への本手法の応用について述べる。

6.3.1 知識分類学習モデルの改善

本研究では、知識獲得の予測性を最適化する知識分類を抽出するにあたり、まず、知識分類学習モデルに問題回答ログを入力し、問題空間から知識タグ空間への写像関数を学習し、その関数を離散化することで、タグを得ていたが、連続値の行列を人間の手によって離散化することにより、情報量の減少が避けられなかった。また、連続表現の知識分類の予測性能と、離散表現の知識分類の予測性能が線形関係にあるとは限らないため、最適な連続表現を得てそれを離散化したとしても、最適な離散表現が得られるとは断言できない。

よって、得られた知識分類が最適な離散表現であることを確実に示すには、初めから離散表現のタグを深層学習によって学習できることがより望ましい。これは、問題の集合の背後に離散的な確率分布が存在することを仮定し、知識獲得予測を最適化させるような分布を学習することによって、最適な離散表現のタグを学習するタスクとして設定できる。データが生成された確率分布を深層学習によって学習するには、一般的な機械学習の識別モデルとは異なる、生成モデルの研究領域における、Variational Autoencoder(VAE) の

技術が利用できる [Kingma et al., 2014]. 従来の VAE で学習できるのは連続的な確率分布のみとされていたが、近年の研究により離散分布についても学習することが可能になった [Maddison et al., 2016, Jang et al., 2016]. この手法を知識分類学習モデルに組み込めば、問題と知識タグの関係性として潜在的な離散分布を仮定し、この分布を学習することにより、勾配法による最適化によって直接最適な離散表現を得ることが可能だと考えられるため、今後の研究課題である。

6.3.2 学習科学における対象データの拡大

次に、学習科学における対象データの拡大について述べる。対象データの拡大とは、科目や難易度の多様化、予測期間の長期化、そして複数科目の統合である。

まず、科目や難易度の多様化について述べる。本研究では、数学の問題回答ログに対して深層学習を適用し、知識獲得の予測性を最適化する知識分類を得た。これまでの DKT の研究成果から、数学以外の教科に対しても適用できる可能性は高いが、実際にどのような知識分類が抽出されるかは分析していない。また、今回扱ったデータセットは、小学校から高校程度の数学に関する問題回答ログであり、より高度で専門的な大学レベルの学問に適用する場合についても、どのような知識分類が抽出されるかは分析していない。知識獲得予測の最適化に関する知見は、学校側からの指導や生徒自身の学習設計に活用されており、多様な難易度や科目において知識獲得を最適化する知識構造を明らかにすることは、重要であると考えられる。

次に、予測期間の長期化について述べる。本研究で用いたデータセットの対象期間は、1～2年程度であった。しかし、知識分類学習モデルによって学習される知識分類は、それまでの生徒の知識獲得の過程に依存しており、できるだけ長い期間の知識獲得を分析するほうが、より適切な知識分類を抽出できる可能性は高い。

最後に、複数科目の統合について述べる。本研究や既存研究では、特定の科目について独立に知識獲得を予測し、知識構造を分析している。しかし、実際の生徒の学習の成長過程は、科目間で完全に独立であるとはいえない、例えば、歴史と地理や、数学と物理などの科目間では、知識獲得の過程が密接に関係している可能性がある。一人の生徒の、科目を横断した知識獲得過程に関する研究は、これまで報告されていないが、複数科目を統合したデータに対して本手法を適用し、科目を横断した知識分類や包括的な学習設計に関する知見を得ることは、学術的な意義が大きいといえる。

6.3.3 学習科学以外の分野への応用

最後に、本手法の学習科学以外の分野への応用について述べる。

本手法は、Knowledge Tracing という、学習科学の、知識獲得予測研究における手法だが、より手法を一般化することで、学習科学以外の分野にも応用できる可能性を秘めている。

本手法は、生徒の時系列問題回答ログから、回答を重ねるごとに遷移していく知識状態をモデリングし、知識獲得の過程を適切に表現する知識分類を抽出するというものだが、これをより一般化して捉えると、人間の、何らかのコンテンツ集合に対する時系列行動ログから、行動を重ねるごとに遷移していく人間の何らかの状態をモデリングし、行動の遷移を適切に説明する分類表現を抽出しているといえる。知識獲得予測においては、このコンテンツ集合に対する行動が生徒の問題回答であり、問題回答により遷移する生徒の知識状態をモデリングしているが、これと同様のことは、学習科学に限らず行える可能性がある。例えば、消費者が商品を購買する時系列ログを分析することで、消費者の嗜好が遷移する過程をモデリングし、従来の商品分類と異なる、消費者の嗜好の遷移を反映した分類を抽出することが可能になり、消費者の購買予測の精度が向上したり、これまでとは異なる系統の商品推薦が行える可能性がある。

知識獲得予測では、問題回答の正誤と知識の間の特殊な関係性をモデル設計に反映しているように、手法を適用する領域によって調整は必要であるが、手法の根本的な特性として、コンテンツに対する行動の時系列性を反映した分類を抽出できる可能性は高く、様々な領域で、学術的にも、実用的にも、価値の高い知見や成果を得られると考えられる。

以上、考察について述べた。次章では、結論を述べる。

第7章 結論

本論文では、既存の知識獲得予測における問題として、人間が作成した知識分類を利用していることを指摘し、深層学習を適用することによって、知識獲得の予測性において最適化された知識分類を抽出できることを検証した。

また、抽出された知識分類を定量的・定性的に分析することにより、知識の内容的な関係性と、分類ごとの回答に関する情報量を最適に分配することが、知識獲得の予測性の向上に寄与することがわかった。

本研究の成果を実際の教育現場で活用例として、教材推薦システムへの適用を論じ、既存の手法では困難だった、他の科目やオンライン教育サービスへ適用できる可能性がある一方、複数科目を統合した知識獲得や、大学水準の知識獲得に関しては、検証実験を行う必要が有ることを述べた。

さらに、本研究の拡張として、より良質な知識分類を学習するために活用できる最新の深層学習技術や、適用対象データの拡張について述べ、また、より一般性を高めた議論として、学習科学以外の領域においても本手法が適用できる可能性に触れ、人間行動に関する多様な知見を発見できる可能性を論じた。

本研究は、教育と情報技術の融合の進展やオンライン教育サービスの普及、教育分野における大規模分析の活発化や深層学習の躍進など、ここ数年の多様な領域の進展によって初めて可能になったものである。本研究が、新たな教育システムの構築、そして人間の学習や知識の解明につながると信じている。

参考文献

- [Abelson, 2008] Abelson, H. (2008). The creation of opencourseware at mit. *Journal of Science Education and Technology*, 17(2):164–174.
- [Aleven et al., 2015] Aleven, V., Sewall, J., Popescu, O., Xhakaj, F., Chand, D., Baker, R., Wang, Y., Siemens, G., Rosé, C., and Gasevic, D. (2015). The beginning of a beautiful friendship? intelligent tutoring systems and moocs. In *International Conference on Artificial Intelligence in Education*, pages 525–528. Springer.
- [Bahdanau et al., 2015] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. (2015). End-to-end attention-based large vocabulary speech recognition. *arXiv preprint arXiv:1508.04395*.
- [Bastien et al., 2012] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.
- [Bellman and Corporation, 1957] Bellman, R. and Corporation, R. (1957). *Dynamic Programming*. Rand Corporation research study. Princeton University Press.
- [Bengio et al., 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.
- [Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305.
- [Bergstra et al., 2010] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a

- CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation.
- [Biswas et al., 2015] Biswas, S., Chadda, E., and Ahmad, F. (2015). Sentiment analysis with gated recurrent units. *Advances in Computer Science and Information Technology*.
- [Bloom, 1968] Bloom, B. S. (1968). Learning for mastery. instruction and curriculum. regional education laboratory for the carolinas and virginia, topical papers and reprints, number 1. *Evaluation comment*, 1(2):n2.
- [Carbonell, 1970] Carbonell, J. R. (1970). Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE transactions on man-machine systems*, 11(4):190–202.
- [Chen et al., 2008] Chen, N.-S., Wei, C.-W., Chen, H.-J., et al. (2008). Mining e-learning domain concept map from academic articles. *Computers & Education*, 50(3):1009–1021.
- [Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- [Choi et al., 2015] Choi, E., Bahadori, M. T., and Sun, J. (2015). Doctor ai: Predicting clinical events via recurrent neural networks. *arXiv preprint arXiv:1511.05942*.
- [Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- [Chung et al., 2015] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2015). Gated feedback recurrent neural networks. *arXiv preprint arXiv:1502.02367*.
- [Clevert et al., 2015] Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

- [Corbett and Anderson, 1994] Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278.
- [Dong et al., 2015] Dong, D., Wu, H., He, W., Yu, D., and Wang, H. (2015). Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. ACL.
- [Dozat, 2015] Dozat, T. (2015). Incorporating nesterov momentum into adam. Technical report, Stanford University, Tech. Rep., 2015.[Online]. Available: <http://cs229-stanford.edu/proj2015/054/report.pdf>.
- [Erhan et al., 2010] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660.
- [FALAKMASIR et al., 2015] FALAKMASIR, M., Yudelson, M., Ritter, S., and Koedinger, K. (2015). Spectral bayesian knowledge tracing. In *Proceedings of the 8th International Conference on Educational Data Mining.*, OC Santos, JG Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, JM Luna, C. Mihaescu, P. Moreno, A. Herskovitz, S. Ventura, and M. Desmarais, Eds. Madrid, Spain, pages 360–364.
- [Friedman, 1997] Friedman, J. H. (1997). On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77.
- [Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256.
- [Graves and Schmidhuber, 2009] Graves, A. and Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 545–552.

- [Hidasi et al., 2015] Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.
- [Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.
- [Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.
- [Hochreiter, 1998] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Jang et al., 2016] Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.
- [Karpathy et al., 2015] Karpathy, A., Johnson, J., and Li, F.-F. (2015). Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.
- [Keller, 1968] Keller, F. S. (1968). Good-bye, teacher... *Journal of applied behavior analysis*, 1(1):79.
- [Kingma and Ba, 2014] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Kingma et al., 2014] Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589.
- [Krueger and Memisevic, 2015] Krueger, D. and Memisevic, R. (2015). Regularizing rnns by stabilizing activations. *arXiv preprint arXiv:1511.08400*.

- [Ku et al., 1995] Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30(1):179–196.
- [Kushner and Yin, 2003] Kushner, H. J. and Yin, G. (2003). *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media.
- [Larochelle et al., 2007] Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480. ACM.
- [Le et al., 2015] Le, Q. V., Jaitly, N., and Hinton, G. E. (2015). A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*.
- [LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- [Lipton et al., 2015] Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. (2015). Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*.
- [Liyanagunawardena et al., 2013] Liyanagunawardena, T., Williams, S., and Adams, A. (2013). The impact and reach of moocs: A developing countries’ perspective. *eLearning Papers*, (33).
- [Louradour and Kermorvant, 2014] Louradour, J. and Kermorvant, C. (2014). Curriculum learning for handwritten text line recognition. In *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*, pages 56–60. IEEE.
- [MacHardy and Pardos, 2015] MacHardy, Z. and Pardos, Z. A. (2015). Toward the evaluation of educational videos using bayesian knowledge tracing and big data. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 347–350. ACM.

- [Maddison et al., 2016] Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.
- [McAuley et al., 2010] McAuley, A., Stewart, B., Siemens, G., and Cormier, D. (2010). The mooc model for digital practice.
- [Midgley, 2014] Midgley, C. (2014). *Goals, goal structures, and patterns of adaptive learning*. Routledge.
- [Mikolov, 2012] Mikolov, T. (2012). Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*.
- [Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). Perceptron (expanded edition).
- [Nair and Hinton, 2010] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.
- [Pappano, 2012] Pappano, L. (2012). The year of the mooc. *The New York Times*, 2(12):2012.
- [Pascanu et al., 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*.
- [Pavlik Jr et al., 2009] Pavlik Jr, P. I., Cen, H., and Koedinger, K. R. (2009). Performance factors analysis—a new alternative to knowledge tracing. *Online Submission*.
- [Pearson, 1901] Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.
- [Pezeshki, 2015] Pezeshki, M. (2015). Sequence modeling using gated recurrent neural networks. *arXiv preprint arXiv:1501.00299*.
- [Piccioli, 2014] Piccioli, V. (2014). E-learning market trends & forecast 2014-2016 report. *Athens (GA)-USA*.

- [Piech et al., 2015] Piech, C., Bassan, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., and Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513.
- [Robbins and Monro, 1951] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- [Rumelhart et al., 1988] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.
- [Sak et al., 2015] Sak, H., Senior, A., Rao, K., and Beaufays, F. (2015). Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*.
- [Schölkopf et al., 1997] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer.
- [Schroff et al., 2015] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *arXiv preprint arXiv:1503.03832*.
- [Siemens, 2013] Siemens, G. (2013). Massive open online courses: Innovation in education. *Open educational resources: Innovation, research and practice*, 5.
- [Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.
- [Sleeman and Brown, 1982] Sleeman, D. and Brown, J. (1982). *Intelligent tutoring systems*. Computers and people series. Academic Press.

- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- [Stamper et al., 2010] Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G., and Koedinger, K. (2010). Bridge to algebra 2006-2007. development data set from kdd cup 2010 educational data mining challenge. <http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp>.
- [Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- [Szegedy et al., 2014] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*.
- [Tetko et al., 1995] Tetko, I. V., Livingstone, D. J., and Luik, A. I. (1995). Neural network studies. 1. comparison of overfitting and overtraining. *Journal of chemical information and computer sciences*, 35(5):826–833.
- [Tipping and Bishop, 1999] Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.
- [Trucano et al., 2013] Trucano, M., Kendrick, C., and Gashurov, I. (2013). More about moocs and developing countries.
- [Upbin, 2012] Upbin, B. (2012). Knewton is building the world’s smartest tutor.
- [Vincent et al., 2008] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.
- [Vinyals et al., 2014] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2014). Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*.

- [Vondrick et al., 2016] Vondrick, C., Pirsiavash, H., and Torralba, A. (2016). Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621.
- [Werbos, 1990] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.
- [Williams and Zipser, 1989] Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.
- [Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.
- [Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- [Xiong et al., 2016a] Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., and Zweig, G. (2016a). Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*.
- [Xiong et al., 2016b] Xiong, X., Zhao, S., Van Inwegen, E. G., and Beck, J. E. (2016b). Going deeper with deep knowledge tracing. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016)*, pages 545–550.
- [Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.
- [Yin et al., 2015] Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., and Li, X. (2015). Neural generative question answering. *arXiv preprint arXiv:1512.01337*.
- [Yuan et al., 2013] Yuan, L., Powell, S., and CETIS, J. (2013). Moocs and open education: Implications for higher education.

- [Yudelson et al., 2013] Yudelson, M. V., Koedinger, K. R., and Gordon, G. J. (2013). Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180. Springer.
- [Zaremba, 2015] Zaremba, W. (2015). An empirical exploration of recurrent network architectures.
- [Zeiler, 2012] Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- [三宅なほみ et al., 2002] 三宅なほみ, 三宅芳雄, and 白水始 (2002). 学習環境のデザイン実験 学習科学と認知科学. **認知科学**, 9(3):328–337.
- [白水始 et al., 2014] 白水始, 三宅なほみ, and 益川弘如 (2014). 学習科学の新展開. **認知科学**, 21(2):254–267.

付録

タグ関係ネットワークにおける抽出タグと既存タグの対応関係

表 1: 「ASSISTments 2009-2010」

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|---|
| 0 | 1.076 | Addition and Subtraction Integers |
| 1 | 0.172 | Probability of a Single Event |
| | 0.172 | Addition and Subtraction Fractions |
| | 0.172 | Addition and Subtraction Integers |
| | 0.080 | Calculations with Similar Figures |
| 2 | 0.154 | Addition and Subtraction Fractions |
| | 0.136 | Probability of a Single Event |
| | 0.104 | Absolute Value |
| | 0.050 | Ordering Positive Decimals |
| | 0.033 | Scientific Notation |
| 3 | 0.209 | Probability of a Single Event |
| | 0.096 | Box and Whisker |
| | 0.082 | Median |
| | 0.075 | Pythagorean Theorem |
| | 0.047 | Write Linear Equation from Ordered Pairs |
| | 0.038 | Volume Cylinder |
| 4 | 0.254 | Addition and Subtraction Fractions |
| | 0.181 | Probability of a Single Event |
| | 0.089 | Percent Of |
| | 0.055 | Order of Operations +,-,/,* () positive reals |
| | 0.143 | Probability of a Single Event |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------------------------------|--|
| | 0.106 0.094 0.064 | Table Mean Range |
| 6 | 0.198 0.184 0.149 | Addition and Subtraction Fractions Probability of a Single Event Percent Of |
| 7 | 0.146 0.134 0.115 0.061 | Addition and Subtraction Fractions Probability of a Single Event Multiplication Fractions Scatter Plot |
| 8 | 0.252 0.185 0.141 | Addition and Subtraction Fractions Probability of a Single Event Percent Of |
| 9 | 0.280 0.135 0.085 | Addition and Subtraction Fractions Probability of a Single Event Percent Of |
| 10 | 0.138 0.092 0.091 | Absolute Value Probability of a Single Event Conversion of Fraction Decimals Percents |
| 11 | 0.185 0.167 0.135 | Probability of a Single Event Addition and Subtraction Integers Equation Solving Two or Fewer Steps |
| 12 | 0.188 0.162 0.083 0.040 | Addition and Subtraction Fractions Probability of a Single Event Box and Whisker Interpreting Coordinate Graphs |
| 13 | 0.436 0.215 | Probability of a Single Event Percent Of |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| | 0.064 | Median |
| 14 | 0.118 | Addition and Subtraction Fractions |
| | 0.085 | Addition and Subtraction Positive Decimals |
| | 0.079 | Order of Operations All |
| 15 | 0.237 | Probability of a Single Event |
| | 0.174 | Addition and Subtraction Positive Decimals |
| | 0.136 | Percent Of |
| 16 | 0.184 | Pattern Finding |
| | 0.125 | Subtraction Whole Numbers |
| | 0.110 | Equation Solving Two or Fewer Steps |
| | 0.076 | Multiplication and Division Integers |
| 17 | 0.217 | Addition and Subtraction Positive Decimals |
| | 0.194 | Probability of a Single Event |
| | 0.136 | Conversion of Fraction Decimals Percents |
| 18 | 0.237 | Probability of a Single Event |
| | 0.154 | Addition and Subtraction Fractions |
| | 0.090 | Box and Whisker |
| 19 | 0.110 | Probability of a Single Event |
| | 0.085 | Addition and Subtraction Fractions |
| | 0.085 | Addition and Subtraction Integers |
| 20 | 0.220 | Addition and Subtraction Fractions |
| | 0.105 | Probability of a Single Event |
| | 0.100 | Percent Of |
| 21 | 0.230 | Probability of a Single Event |
| | 0.146 | Conversion of Fraction Decimals Percents |
| | 0.080 | Equation Solving Two or Fewer Steps |
| | 0.046 | Area Irregular Figure |
| | 0.045 | Square Root |
| | 0.188 | Probability of a Single Event |

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------------|--|
| | 0.108 0.088 | Addition and Subtraction Fractions Median |
| 23 | 0.183 | Finding Percents |
| | 0.146 | Pattern Finding |
| | 0.118 | Addition and Subtraction Positive Decimals |
| | 0.040 | Effect of Changing Dimensions of a Shape Prportionally |
| | 0.011 | Perimeter of a Polygon |
| | 0.011 | Stem and Leaf Plot |
| 24 | 0.164 | Equation Solving Two or Fewer Steps |
| | 0.100 | Addition and Subtraction Integers |
| | 0.091 | Table |
| | 0.052 | Fraction Of |
| | 0.046 | Mode |
| 25 | 1.076 | Addition and Subtraction Integers |
| 26 | 0.198 | Probability of a Single Event |
| | 0.132 | Addition and Subtraction Fractions |
| | 0.094 | Box and Whisker |
| | 0.051 | Area Trapezoid |
| 27 | 0.201 | Probability of a Single Event |
| | 0.132 | Addition and Subtraction Fractions |
| | 0.092 | Box and Whisker |
| | 0.064 | Divisibility Rules |
| 28 | 0.205 | Probability of a Single Event |
| | 0.205 | Addition and Subtraction Fractions |
| | 0.092 | Box and Whisker |
| 29 | 0.191 | Probability of a Single Event |
| | 0.096 | Percent Of |
| | 0.091 | Addition and Subtraction Fractions |
| | 0.204 | Addition and Subtraction Fractions |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|-------------------------|---|
| | 0.133 0.110 0.091 | Probability of a Single Event Conversion of Fraction Decimals Percents Ordering Fractions |
| 31 | 0.931 | Write Linear Equation from Graph |
| | 0.359 | Addition and Subtraction Integers |
| 32 | 0.236 | Probability of a Single Event |
| | 0.202 | Percent Of |
| | 0.147 | Addition and Subtraction Fractions |
| | 0.062 | Addition Whole Numbers |
| 33 | 0.230 | Addition and Subtraction Positive Decimals |
| | 0.097 | Order of Operations All |
| | 0.076 | Conversion of Fraction Decimals Percents |
| 34 | 0.227 | Probability of a Single Event |
| | 0.191 | Addition and Subtraction Fractions |
| | 0.139 | Percent Of |
| 35 | 0.173 | Probability of a Single Event |
| | 0.135 | Addition and Subtraction Fractions |
| | 0.113 | Addition and Subtraction Integers |
| 36 | 0.189 | Probability of a Single Event |
| | 0.189 | Addition and Subtraction Fractions |
| | 0.096 | Percent Of |
| | 0.036 | Estimation |
| 37 | 0.264 | Probability of a Single Event |
| | 0.244 | Addition and Subtraction Fractions |
| | 0.193 | Percent Of |
| 38 | 0.223 | Pattern Finding |
| | 0.152 | Subtraction Whole Numbers |
| | 0.109 | Finding Percents |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| | 0.050 | Histogram as Table or Graph |
| 39 | 0.158 | Probability of a Single Event |
| | 0.130 | Addition and Subtraction Fractions |
| | 0.108 | Percent Of |
| 40 | 0.258 | Addition and Subtraction Fractions |
| | 0.167 | Probability of a Single Event |
| | 0.130 | Box and Whisker |
| 41 | 0.302 | Finding Percents |
| | 0.189 | Addition and Subtraction Positive Decimals |
| | 0.099 | Conversion of Fraction Decimals Percents |
| 42 | 0.121 | Addition and Subtraction Positive Decimals |
| | 0.111 | Probability of a Single Event |
| | 0.090 | Addition and Subtraction Fractions |
| 43 | 0.249 | Probability of a Single Event |
| | 0.184 | Addition and Subtraction Fractions |
| | 0.125 | Percent Of |
| 44 | 0.247 | Addition and Subtraction Fractions |
| | 0.129 | Probability of a Single Event |
| | 0.108 | Percent Of |
| 45 | 0.212 | Probability of a Single Event |
| | 0.173 | Addition and Subtraction Fractions |
| | 0.100 | Box and Whisker |
| 46 | 0.204 | Probability of a Single Event |
| | 0.185 | Addition and Subtraction Fractions |
| | 0.118 | Percent Of |
| 47 | 0.326 | Surface Area Rectangular Prism |
| | 0.272 | Pattern Finding |
| | 0.205 | Equation Solving More Than Two Steps |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| | 0.175 | Venn Diagram |
| 48 | 0.145 | Probability of a Single Event |
| | 0.129 | Equation Solving Two or Fewer Steps |
| | 0.099 | Addition and Subtraction Fractions |
| | 0.145 | Addition and Subtraction Positive Decimals |
| 49 | 0.081 | Equation Solving Two or Fewer Steps |
| | 0.064 | Equivalent Fractions |
| | 0.061 | Division Fractions |
| | 0.034 | Volume Sphere |
| | 0.111 | Probability of a Single Event |
| 50 | 0.111 | Addition and Subtraction Integers |
| | 0.090 | Probability of Two Distinct Events |
| | 0.074 | Counting Methods |
| | 0.071 | Proportion |
| | 0.134 | Equation Solving Two or Fewer Steps |
| 51 | 0.121 | Addition and Subtraction Positive Decimals |
| | 0.115 | Probability of a Single Event |
| | 0.175 | Probability of a Single Event |
| 52 | 0.128 | Percent Of |
| | 0.081 | Addition and Subtraction Fractions |
| | 0.079 | Circle Graph |
| | 0.054 | Unit Rate |
| | 0.372 | Equation Solving Two or Fewer Steps |
| 53 | 0.203 | Order of Operations All |
| | 0.130 | Probability of a Single Event |
| | 0.073 | Least Common Multiple |
| | 0.072 | Circumference |
| | 0.538 | Probability of a Single Event |

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|------------------------------------|
| | 0.538 | Addition and Subtraction Fractions |

以上

表 2: 「Bridge to Algebra 2006-2007」

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| 0 | 0.388 | Identify common denominator |
| | 0.344 | Rewrite fraction with common denominator |
| | 0.290 | Identify number of equal divisions on number line from desired denominator |
| 1 | 0.707 | Identify Fraction using fraction shape |
| | 0.561 | Calculate product of two numbers |
| | 0.561 | Identify GCF in written question |
| 2 | 0.842 | Enter smaller initial in diagram – calculated |
| | 0.774 | Draw larger bar – addition/subtraction |
| | 0.726 | Identify number of desired groups |
| 3 | 0.442 | Calculate difference with positive integer |
| | 0.357 | Identify multiplier in written question – number line |
| | 0.304 | Identify fraction associated with each piece of a square |
| 4 | 0.774 | Draw larger bar – addition/subtraction |
| | 0.689 | Draw smaller bar – addition/subtraction |
| | 0.518 | Identify multiplier in equivalence statement |
| 5 | 0.383 | Identify number as common factor |
| | 0.293 | Identify number of items |
| | 0.220 | Calculate Part |
| 6 | 0.574 | Represent multiplier visually |
| | 0.574 | Calculate product – multiply statement |
| | 0.355 | Label equivalent fraction on number line |
| 7 | 0.574 | Identify equal parts for multiplier |
| | 0.440 | Identify number as common factor |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| | 0.436 | Enter subtracted quantity in diagram |
| 8 | 1.590 | List consecutive multiples of a number |
| | 0.734 | Identify number as common factor |
| 9 | 1.436 | Identify that a fraction can/cannot be simplified |
| | 1.090 | Enter items denominator |
| | 0.837 | Identify larger quantity – multiplication |
| 10 | 0.957 | Identify whole number of improper fraction symbolically |
| | 0.734 | Identify number as common factor |
| | 0.726 | Identify fractional part of improper fraction symbolically |
| 11 | 0.611 | Rewrite fraction with common denominator |
| | 0.536 | Identify LCM - is product |
| | 0.407 | Compare fractions with like denominators |
| 12 | 0.410 | Identify width of overlap |
| | 0.332 | Identify number of equal divisions on number line from desired denominator |
| | 0.311 | Represent second fraction on number line as difference |
| 13 | 1.149 | Represent negative integer using number line |
| | 0.733 | Enter added quantity in diagram |
| | 0.550 | Identify improper fraction from option 2 |
| 14 | 1.090 | Calculate sum – contextual |
| | 0.861 | Count number of shaded parts in square (discontiguous) |
| | 0.728 | Draw larger bar – multiplication |
| 15 | 1.684 | Identify number of equal divisions in visual from fraction |
| | 0.720 | Identify number of recipients |
| | 0.489 | Identify number as common factor |
| 16 | 0.957 | Rewrite adding positive integer |
| | 0.842 | Represent first integer on number line |
| | 0.720 | Identify number of recipients |
| 17 | 2.526 | Identify number of equal divisions (vertical bar) |

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| | 0.734 | Identify number as common factor |
| 18 | 0.957 | Identify number of desired groups–construct |
| | 0.720 | Identify number of recipients |
| | 0.485 | Draw larger bar – multiplication |
| 19 | 1.090 | Identify number of desired groups |
| | 1.034 | Identify negative integer from number line |
| | 0.795 | Identify Fraction using fraction shape |
| 20 | 0.718 | Represent multiplicand visually |
| | 0.718 | Identify fraction associated with each piece of a horizontal bar |
| | 0.652 | Identify improper fraction from option 1 |
| 21 | 1.436 | Draw smaller bar – multiplication |
| | 1.263 | Identify fraction of desired items |
| | 1.263 | Identify fraction associated with each piece of a circle |
| 22 | 3.953 | Calculate sum – non contextual |
| 23 | 0.734 | Identify number as common factor |
| | 0.689 | Identify number of equal groups from fraction |
| | 0.485 | List factor of large number |
| 24 | 0.971 | Draw larger bar – multiplication |
| | 0.749 | Identify number of items |
| | 0.489 | Identify number as common factor |
| 25 | 2.179 | Identify number of items in each group |
| | 1.456 | List factor of large number |
| 26 | 0.821 | Identify larger quantity – subtraction |
| | 0.722 | Identify GCF in written question |
| | 0.629 | Identify number as common factor |
| 27 | 1.436 | Identify whole number upper bound |
| | 1.263 | Identify number of equal divisions (circle) |
| | 1.090 | Identify fractional part of improper fraction symbolically |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| 28 | 0.774 | Identify multiplier in written question – rectangle |
| | 0.726 | Enter subtracted quantity in diagram |
| | 0.633 | Label equivalent fraction in equivalence statement |
| 29 | 2.872 | Write improper fraction as mixed number |
| | 1.833 | Compare fractions with like denominators |
| 30 | 1.222 | Calculate difference – contextual |
| | 0.957 | Identify GCF |
| | 0.842 | Calculate difference – non contextual |
| 31 | 1.010 | Enter smaller initial in diagram – calculated |
| | 0.760 | Represent first fraction on number line |
| | 0.550 | Rewrite fraction with common denominator |
| 32 | 2.526 | Calculate Total |
| | 1.206 | Identify LCM - is product |
| 33 | 0.383 | Identify whole number lower bound |
| | 0.337 | Identify fraction associated with each piece of a vertical bar |
| | 0.294 | Identify number as common factor |
| 34 | 1.305 | Identify improper fraction from option 1 |
| | 1.124 | Identify number of items |
| 35 | 0.916 | Identify GCF - one number multiple of other |
| | 0.687 | Identify improper fraction from option 2 |
| | 0.603 | Identify LCM - is product |
| 36 | 0.397 | Identify number of items |
| | 0.338 | Enter smaller initial in diagram – given |
| | 0.323 | Compare Options - operation |
| 37 | 2.161 | Identify number of recipients |
| 38 | 1.149 | Write mixed number as improper fraction |
| | 0.760 | Represent first fraction on number line |
| | 0.710 | Label equivalent fraction on number line |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| 39 | 0.409 | Identify number of items |
| | 0.396 | Represent second fraction on number line as difference |
| | 0.393 | Identify number of recipients |
| 40 | 0.454 | Identify Fraction using fraction shape |
| | 0.321 | Identify number of items |
| | 0.315 | Identify number as common factor |
| 41 | 1.553 | Identify multiplier in equivalence statement |
| | 0.652 | Identify improper fraction from option 1 |
| | 0.367 | Identify number as common factor |
| 42 | 0.718 | Identify equal parts for multiplicand |
| | 0.545 | Calculate sum – contextual |
| | 0.540 | Identify number of recipients |
| 43 | 2.609 | Identify improper fraction from option 1 |
| 44 | 1.263 | Calculate Part |
| | 0.728 | Draw larger bar – multiplication |
| | 0.687 | Compare Options - operation |
| 45 | 0.516 | Identify multiplier in written question – number line |
| | 0.489 | Identify number as common factor |
| | 0.459 | Identify number of equal groups from fraction |
| 46 | 1.468 | Identify number as common factor |
| 47 | 0.381 | Identify number as common factor |
| | 0.236 | List consecutive multiples of a number |
| | 0.213 | Identify whole number of mixed number symbolically |
| 48 | 0.442 | Calculate area of overlap |
| | 0.339 | Identify number as common factor |
| | 0.335 | Compare fractions from contextual problem |
| 49 | 0.536 | Identify LCM - is product |
| | 0.484 | Enter items numerator |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| | 0.362 | Identify larger quantity – addition |
| 50 | 1.010 | Identify GCF in equivalence statement |
| | 0.550 | Compare Options - operation |
| | 0.550 | Identify improper fraction from option 2 |
| 51 | 0.281 | Represent first fraction on number line |
| | 0.250 | Identify number of items |
| | 0.240 | Identify number of recipients |
| 52 | 0.451 | Calculate sum – contextual |
| | 0.409 | Identify fraction associated with each piece of a square |
| | 0.373 | Identify number of recipients |
| 53 | 0.447 | Label equivalent fraction in equivalence statement |
| | 0.397 | Identify number of items |
| | 0.345 | Identify number as common factor |
| 54 | 1.149 | Represent second positive integer on number line as difference |
| | 0.827 | Draw smaller bar – addition/subtraction |
| | 0.587 | Identify number as common factor |
| 55 | 1.436 | Convert improper fraction to whole number |
| | 0.728 | List factor of large number |
| | 0.540 | Identify number of recipients |
| 56 | 0.629 | Identify number as common factor |
| | 0.543 | Enter larger initial in diagram – calculated |
| | 0.393 | Rewrite fraction with common denominator |
| 57 | 0.722 | Identify number of equal divisions in visual from fraction |
| | 0.722 | Identify GCF in equivalence statement |
| | 0.524 | Compare fractions with unlike denominators |
| 58 | 0.957 | Calculate sum with positive integer |
| | 0.611 | Enter added quantity in diagram |
| | 0.458 | Identify improper fraction from option 2 |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|---|
| 59 | 0.774 | Identify multiplier in written question – rectangle |
| | 0.726 | Enter items numerator |
| | 0.591 | Label equivalent fraction on number line |
| 60 | 0.957 | Isolate numerator of fractional part of mixed number symbolically |
| | 0.957 | Enter larger initial in diagram – given |
| | 0.842 | Identify fraction of desired items |
| 61 | 1.206 | Identify LCM - is product |
| | 0.734 | Identify number as common factor |
| 62 | 0.581 | Identify number of desired items |
| | 0.545 | Compare fractions from contextual problem |
| | 0.540 | Identify number of recipients |
| 63 | 1.090 | Represent second fraction on number line as sum |
| | 0.652 | Identify improper fraction from option 1 |
| | 0.562 | Identify number of items |
| 64 | 1.206 | Identify LCM - is product |
| | 0.734 | Identify number as common factor |
| 65 | 0.550 | Rewrite fraction with common denominator |
| | 0.392 | Identify number as common factor |
| | 0.367 | Compare Options - operation |
| 66 | 0.530 | List consecutive multiples of a number |
| | 0.375 | Identify number of items |
| | 0.363 | Calculate sum – contextual |
| 67 | 0.647 | List factor of large number |
| | 0.561 | Calculate Total |
| | 0.484 | Label equivalent fraction in visual |
| 68 | 0.518 | Identify multiplier in equivalence statement |
| | 0.479 | Enter quantity from diagram by calculating |
| | 0.345 | Identify number of equal groups from fraction |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| 69 | 1.222 | Enter added quantity in diagram |
| | 0.749 | Identify number of items |
| | 0.489 | Identify number as common factor |
| 70 | 0.397 | Identify number of items |
| | 0.297 | Identify number of items in each group from GCF |
| | 0.284 | Identify LCM - is product |
| 71 | 1.553 | Identify multiplier in equivalence statement |
| | 1.456 | Draw larger bar – multiplication |
| 72 | 0.957 | Identify number of equal divisions (horizontal bar) |
| | 0.774 | Count number of shaded parts in vertical bar (discontiguous) |
| | 0.659 | Calculate sum – non contextual |
| 73 | 1.034 | Identify negative integer from number line |
| | 0.687 | Compare Options - operation |
| | 0.652 | Identify improper fraction from option 1 |
| 74 | 2.749 | Identify improper fraction from option 2 |
| 75 | 0.734 | Identify number as common factor |
| | 0.611 | Identify that a fraction cannot be simplified |
| | 0.485 | Draw larger bar – multiplication |
| 76 | 0.687 | Rewrite fraction with common denominator |
| | 0.631 | Identify number of items in each group from GCF |
| | 0.408 | Identify larger quantity – addition |
| 77 | 0.367 | Identify improper fraction from option 2 |
| | 0.367 | Rewrite fraction with common denominator |
| | 0.337 | Identify fraction associated with each piece of a circle |
| 78 | 3.665 | Compare fractions with unlike denominators |
| 79 | 0.760 | Represent first fraction on number line |
| | 0.733 | Identify that a fraction cannot be simplified |
| | 0.621 | Identify common denominator |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| 80 | 1.060 | Identify Fraction using fraction shape |
| | 0.971 | Draw larger bar – multiplication |
| | 0.489 | Identify number as common factor |
| 81 | 0.821 | Enter total in diagram - calculated - addition |
| | 0.664 | Count number of shaded parts in vertical bar (discontiguous) |
| | 0.623 | Label equivalent fraction in visual |
| 82 | 0.707 | List consecutive multiples of a number |
| | 0.638 | Identify number of total items |
| | 0.561 | Identify fraction associated with each piece of a vertical bar |
| 83 | 1.350 | Identify number as common multiple |
| | 1.090 | Identify fractional part of improper fraction symbolically |
| | 0.367 | Identify number as common factor |
| 84 | 3.347 | Identify larger quantity – multiplication |
| 85 | 0.631 | Identify proper fraction from option 1 |
| | 0.562 | Identify number of items |
| | 0.430 | Count number of shaded parts in square (discontiguous) |
| 86 | 0.827 | Draw smaller bar – addition/subtraction |
| | 0.636 | Identify Fraction using fraction shape |
| | 0.636 | List consecutive multiples of a number |
| 87 | 0.842 | Identify number of equal divisions (circle) |
| | 0.689 | Enter quantity from diagram by reading |
| | 0.633 | Enter larger initial in diagram – calculated |
| 88 | 1.222 | Identify GCF - one number multiple of other |
| | 0.916 | Compare Options - operation |
| | 0.804 | Identify LCM - is product |
| 89 | nan | Count number of shaded parts in circle (contiguous) |
| | nan | Convert whole number to improper fraction |
| | nan | Represent second positive integer on number line as sum |
| | nan | Enter initial in diagram – given |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|-------------------------|--|
| | nan nan | Write mixed number as improper fraction Enter group denominator |
| 90 | 2.068 1.674 | Identify negative integer from number line Identify larger quantity – multiplication |
| 91 | 1.590 1.456 | Identify Fraction using fraction shape List factor of large number |
| 92 | 0.776 0.540 0.458 | Identify common denominator Identify number of recipients Compare fractions with like denominators |
| 93 | 0.638 0.484 0.345 | Enter total in diagram - calculated - multiplication Enter group numerator Identify common denominator |
| 94 | 2.179 0.734 | Enter items denominator Identify number as common factor |
| 95 | 3.799 | Represent first fraction on number line |
| 96 | 0.450 0.392 0.310 | Identify number of items Identify number as common factor Count number of shaded parts in vertical bar (discontiguous) |
| 97 | 3.665 | Identify that a fraction cannot be simplified |
| 98 | 1.178 0.664 0.524 | Compare Options - operation Draw larger bar – addition/subtraction Identify GCF - one number multiple of other |
| 99 | 0.522 0.522 0.359 | Identify length of overlap Identify proper fraction from option 2 Identify fraction associated with each piece of a square |
| 100 | 1.915 0.916 0.870 | Identify fractional part of mixed number symbolically Compare Options - operation Identify improper fraction from option 1 |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|--|
| 101 | 0.611 | Enter added quantity in diagram |
| | 0.611 | Compare fractions with unlike denominators |
| | 0.530 | Identify Fraction using fraction shape |
| 102 | 1.468 | Identify number as common factor |
| 103 | 1.123 | Calculate difference – non contextual |
| | 0.500 | Identify number of items |
| | 0.439 | Identify fraction associated with each piece of a square |
| 104 | 0.957 | Copy initial in diagram |
| | 0.957 | Compare Options - simplified |
| | 0.543 | Identify number of equal divisions (square) |
| 105 | 3.665 | Compare fractions with like denominators |
| 106 | 0.733 | Enter added quantity in diagram |
| | 0.669 | Identify larger quantity – multiplication |
| | 0.550 | Rewrite fraction with common denominator |
| 107 | 0.551 | Identify number as common factor |
| | 0.545 | Enter subtracted quantity in diagram |
| | 0.458 | Compare fractions with unlike denominators |
| 108 | 0.500 | Identify number of items |
| | 0.484 | Enter items numerator |
| | 0.484 | Enter group denominator |
| 109 | 0.929 | Identify number of equal divisions on number line from desired denominator |
| | 0.636 | Identify Fraction using fraction shape |
| | 0.550 | Identify improper fraction from option 2 |
| 110 | 0.611 | Compare Options - operation |
| | 0.516 | Identify number of desired items |
| | 0.484 | Identify number of items in each group |
| 111 | 1.100 | Identify improper fraction from option 2 |
| | 1.010 | Count number of shaded parts in square (contiguous) |

次ページに続く

前ページからの続き

| 抽出タグ | TF-IDF 値 | 既存タグ |
|------|----------|---|
| | 0.827 | Draw smaller bar – addition/subtraction |
| 112 | 0.475 | Identify that a fraction can be simplified |
| | 0.359 | Enter total in diagram - calculated - subtraction |
| | 0.344 | Compare Options - operation |
| 113 | 0.442 | Enter initial in diagram – given |
| | 0.335 | Enter items numerator |
| | 0.335 | Enter group numerator |
| 114 | 0.965 | Identify LCM - is product |
| | 0.760 | Enter larger initial in diagram – calculated |
| | 0.733 | Compare fractions with like denominators |

以上

謝辞

本研究の遂行や本論文の執筆にあたり、非常に多くの方からご指導、ご支援をいただきました。心より御礼申し上げます。

指導教官である松尾豊特任准教授には、研究構想の相談や論文の書き方、本論文の論理構成について、貴重なご指導をいただきました。ここに、深く謝意を表します。

分析サーバや GPU 解析環境の用意等、物理的な研究環境の構築に多大なご協力を下さった研究室の教官である中山浩太郎先生に、深く感謝致します。

上野山克也助教授には、研究の方向性や論文の構成について、多大なご指導をいただきました。深く感謝致します。

松尾研究室や GCI の皆様には、多大なご協力、ご支援いただきました。秘書の中野佐恵子さん、永本登代子さん、浪岡亮子さん、木全弥栄さんは、日頃から研究室の環境を整えて下さり、研究生活を支えてくださいました。松尾研究室の博士・修士課程の先輩である岩澤有祐さん、飯塚修平さん、野中尚輝さん、鈴木雅大さん、金子貴輝さん、那須野薫さん、黒滝紘生さん、保住純さん、富山翔司さんには、研究の相談に幾度も乗っていただき、多大なご助力をいただきました。特に、研究テーマや、研究全体の設計について何度も相談に応じていただいたことに加え、日常の議論を通じて多くの知識や示唆をいただいた那須野薫さんには、多大なるご助力に深く感謝致します。研究室の同期である大野峻典氏、田村浩一郎氏は、卒業論文の構想や執筆に関して率直に意見を交わし、互いに切磋琢磨し合いながら研究を進めさせていただきました。

ここに、松尾研究室の皆様へ謝意を表します。

東京大学工学部
システム創成学科知能社会コース
松尾研究室 学部四年
中川大海
平成 29 年 3 月