

手書き数式認識を用いた計算過程の正誤フィードバック

森重湧太^{†1} 中川正樹^{†2}

数学的思考力を測るには、マークシートで採用されている選択問題や Web 試験などで利用されているマウス選択あるいはキーボード入力可能な問題ではなく、数式を筆記させる問題が有用である。しかし、採点者が採点して返却するまでに時間を要すると、テスト受験者の記憶が薄れてしまい、返却後に十分復習されない問題がある。そこで、本稿では初等数学において、数式を複数行に展開した手書きの計算過程について、それを行単位で正誤判定し、フィードバックを即時に提供する方法を提案する。

Immediate marking feedback by handwritten mathematical formula recognition

YUTA MORISHIGE^{†1} MASAKI NAKAGAWA^{†2}
SHUNSUKE UEMURA^{†3}

In order to assess mathematical thinking abilities, letting examinees to answer questions by writing formulas is effective rather than answering by selection or keyboard input. If marking takes time, however, the examinees likely forget what they have considered to answer the questions and the effect of reviewing them decreases. Therefore, we propose to employ a handwritten mathematical formula recognizer, which judges the correctness of expansions for solving math questions and make the immediate feedback.

1. はじめに

近年、教育現場への ICT 導入が進んでいる。教育アプリケーション、e-ラーニングシステム、ICT を利用したテストシステムなどの利用が拡大するとともに、校務の情報化も進んでいる。一方、様々な教科科目（国語、数学、英語等）における学力を測る手段として、現状ではペーパーテストが主流である。とくに、数学においては解答だけでなく、解答を導き出す過程も重視されているため、ICT を活用した数学学習支援や採点支援では対応が困難である。

初等数学（中学校までの算数や数学）においては、数式の変形や展開等のいくつかの過程を経て、解を導出する。教師は回答が正解かどうか以外にも導出過程も整合的に確認し、採点を行う。また、教師が学習者に適切な助言を行う際にも導出過程はその材料となっており、ペーパーテストで数学的思考力を測ることが望ましいと考えられる。

ところが、ペーパーテストにも欠点がある。答案の返却が遅れると、採点がうまく復習に生かされない。学生はテストの終了後から記憶の減衰が始まるため、記憶定着のためにも早期のフィードバックが必要である。しかし、教師は担当生徒全員分の全計算過程を確認する必要があり、返却には時間を要するという問題がある。

これらの問題を解決するため、学習システムや学習デザインが研究されている。しかし、正誤フィードバックの方法、

入力インタフェース、認識精度などの点で実用的なシステムは実現されていない。

本研究では、この問題への解決に、急速に普及しつつあるタブレットと手書き認識を利用することを考えている。タブレットではペンや指による直接操作が可能であり、何より、ユーザは筆記により入力することができる。様々なインタラクティブ性を持ったアプリケーションが開発されており、それは教育用のアプリケーションとしても応用されている。

本研究では、インタラクティブ性の高い手書き入力方式を採用し、当研究室の Le Duc Ahn ら[1]が研究する最新の数式認識エンジンを利用することで実用に耐えうる認識精度を確保し、計算過程の正誤フィードバックを実現する。従来手法における問題点とその原因を考察し、基本機能を実現したプロトタイプによる学習方法を提案する。

2. 先行研究と本研究の位置付け

2.1 先行研究

この問題を解決しようとする先行研究としては、鈴木ら[2]の研究と大前ら[3]の研究がある。

鈴木らの研究では、手書き数式解析に基づき、基礎数学学習支援システムを開発した。このシステムでは、教師側が専用の問題作成ツールにより問題を作成し、学習者がその問題について計算過程をペン入力する。入力されたスト

^{†1} 東京農工大学大学院 工学府 情報工学専攻
Department of Information and Technology,
Tokyo University of Agriculture and Technology

^{†2} 東京農工大学
Tokyo University of Agriculture and Technology

ワークから数式認識処理[4]により認識結果が表示される。そして、認識結果から計算過程の整合性検証を行う。この整合性検証は、まず、それぞれの式を標準系に変換し、その係数ベクトルを比較することで式の等価性を判定する。そして、式変形に適用する規則（分配・交換・結合・約分など）を何回使用して実現できるかという「導出ステップ数」を比較する。問題の導出ステップ数と比較して、学習者の導出ステップ数が大きくなるほど、飛躍した計算をしていると判断し、指導対象であると教師にフィードバックする。これらの検証を通して、妥当であれば「正解」という文字が一言表示される。

しかし、この導出ステップ数は、学習者の習熟度により大きく変化するため、妥当な検証方法とはいえず、これに対応できる柔軟性が必要である。正解のフィードバックがどこで行って間違えたかをフィードバックしないため、教師・学習者共にひと目でどの行が間違っているかを判断するのが困難であり、そこに時間を要する問題がある。

これに対して大前らはスキル獲得[5][6]の考えに基づき、積分解法スキル獲得について誤答箇所指摘型の学習デザインを提案し、検証実験を行った結果、妥当であることを示した。これを自動化するためのプロトタイプを作成したが、数式パーツをドラッグアンドドロップする入力インターフェースであり、実用的であるとは言えなかった。

2.2 本研究の位置付け

鈴木らが行った手書き数式の学習支援では、高い習熟度を持つ学習者に対しては適切なフィードバックを行えず、計算能力の獲得についても言及されていない。これに対し、大前らは誤答箇所指摘型の学習デザインを提案し、計算能力の獲得に有効であることを示した。その学習デザインを適用したシステムを開発したが、実用に耐えるものではなかった。

これらを踏まえ、本研究では、手書きの計算過程について、誤答箇所指摘型の学習デザインを適用し、学習者の習熟度によらず、即座にフィードバックを行うシステムを開発し、問題点の考察を行う。システムには、研究室で開発中の手書き数式認識エンジンを搭載し、その有用性を考察する。

また、複雑な問題設定ツールを用いず、問題の数式を入力するだけで正誤判定が可能なシステムとして実現することで、問題設定の負担軽減を目指す。さらに、学習用途だけでなく、数学テストの自動採点への利用についても考察を行う。

本研究で扱う数式は現行の中学校数学の範囲の中で行う単純に展開・因数分解等を必要とする計算式に限定した。

3. 学習プロセス

計算式の即時正誤フィードバックを用いた学習プロセスを説明する。従来の学習プロセスを図1に、これに対す

る誤答箇所指摘型の学習プロセスを図2に示す。

図1従来の学習プロセスについて説明する。従来の方法では学習者が参考書等を参照し問題を解いて、解き終わってから解答チェックを行い、正当の場合は次の問題へ進む。誤答の場合は学習者が全計算過程から誤答箇所を推測しなければならず、正答である行についても参考書と照らし合わせるなどの行為が起こる可能性があり、誤答箇所の発見に時間を要し、学習効率が下がると考えられる。誤答箇所が不明であり、参考書を参照しても理解できない場合、その問題はあきらめる。

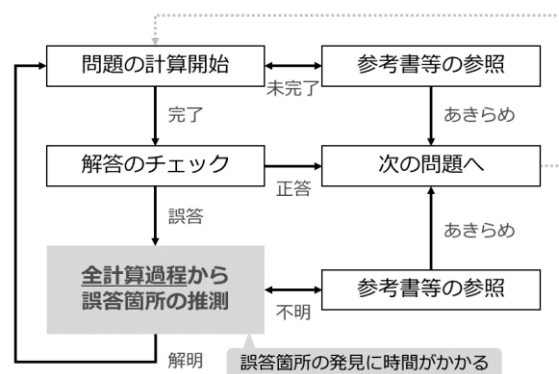


図1. 従来の学習プロセス

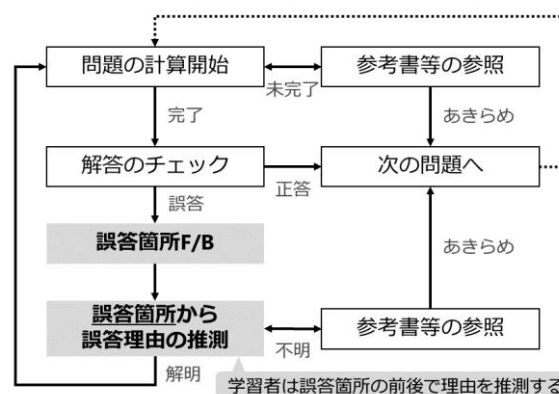


図2. 誤答箇所指摘型の学習プロセス

図2の誤答箇所指摘型の学習方法は、解答チェックの後、誤答箇所のフィードバックがある。このフィードバックを行う利点として学習者は誤答である行あるいは前後で誤答理由の推察を進めることが可能である点である。誤答箇所の探索に要する時間を、誤答理由の推測のための時間に充てることで学習効率が高まると考えられる。

この誤答箇所指摘型の学習プロセスを用い、システムにより自動化する箇所と、フィードバックのタイミングについて説明する。図3を参照されたい。

学習者は提示された問題についてペン入力デバイスを用いて計算を記述する。学習者の記述した計算式を正誤判定システムが確認し、判定した結果についての誤答であれば、「誤答であると判定した行」をフィードバックする。学習者はその指摘を参考に、誤答の式を修正する。修正した式が正しいと判定されれば、次の行あるいは問題へ進む。計算が正しい場合は正しいことをフィードバックする。

フィードバックのタイミングは学習者が計算式を1行書くごとにフィードバックを返す方式をとっている。

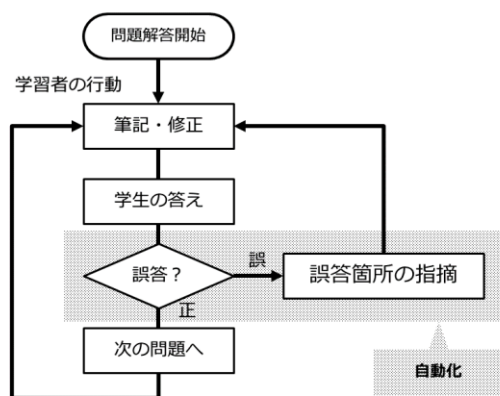


図 3. 学習者の行動と正誤判定の自動化

4. プロトタイプの試作

基本機能を実現したプロトタイプシステムの試作を行った。OS は Windows8.1 を使い、Visual Studio C++ で開発を行った。システムの概要を図 4 に示す。

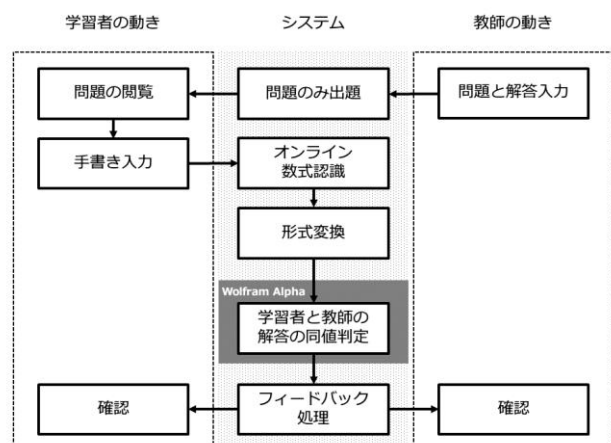


図 4. システム概要図

出題者である教師は、問題を作成し、システムに問題と解答を入力するだけである。学習者は問題を見て、数式を入力する。入力されたストロークは、オンライン数式認識により、中学校校までで出現する数式文法の辞書から、認識

結果を出力する。認識結果、解答の正誤判定を行う Wolfram Alpha にリクエストできる形に前処理を行った後、Wolfram Alpha にリクエストし、XML でレスポンスを受け取る。そのレスポンスは同値判定を行った結果である。その結果によってフィードバックを表示する。

4.1 正誤判定に利用する Web API

正誤判定には Wolfram Alpha Web Service API を利用する。Wolfram Alpha は、Mathematica 上に構築された Web サービスであり、無料で公開されている API を通して正誤判定のリクエストを行う。

本研究では、この API 中の「Equal (lhs, rhs)関数」の組み込みシンボルを利用した。Equal 関数は lhs と rhs の2つの引数を持ち、2つの引数について数値的等価性と記号的等価性の両面から検証を行う。

数値的等価性の検証では小数点第 10 位以内まで同値であれば等価と判定する。ルートがある場合は、Wolfram Alpha の独自の関数により単一のルートオブジェクトに還元してから判定する。

記号的等価性においては、2式 lhs と rhs を昇べき整理し、記号的な恒等式として等価かどうかを検証する。展開が必要であれば、Expand 関数という数式を展開する関数において展開を行っている。

この2つの等価性を検証した結果、lhs と rhs が等価である場合は「True」、等価でない場合は「False」、数式文法上の誤り等によるエラーに対しては「Failure」が返される。

今回は問題の模範解答と、学習者の記述した式を、それぞれ lhs と rhs に入力した Equal 関数でリクエストする。

初等数学での扱う数式を7種類に分け、正しく等価性を判定できたものを表 1 に示す。証明問題などの文字や図による説明を伴う数学は扱っていない。

表 1.判定可能な数式の種類

数式の種類	具体例	等価性判定
加減乗除	$1+1$	可能
四則混合計算	$1+2\times 3$	可能
文字式	$2a+a$	登録文字のみ
1元1次式	$2(x+1)$	可能
n元n次式	$x^2+y^3+y^3$	可能
1次方程式	$2x=4$	可能
2次方程式	$x^2+x^2=8$	可能

本システムでは表 1 のように、文字式以外の数式について正しく判定できることを確認している。文字式については、Wolfram Alpha のサービスが取り扱っていない文字は正しく処理されなかった。a,b,c,d,x,y,z を始めとする初等数学

でよく扱う文字は正しく処理された。

Wolfram Alpha のソースコードについては非公開であるため、詳細なアルゴリズムは不明である。

4.2 手書き数式認識エンジン

手書き数式認識エンジンは当研究室の Le Duc Anh らが開発しているものを利用した。確率文脈自由文法による構造解析と、SVM によるシンボル分割、MRF によるオンライン認識と MQDF によるオフラインによってシンボル認識を行っている。初等中等教育段階の数式認識に注力しており、市販の電子教材に利用されている。

4.3 GUI

試作したプロトタイプの入力画面（図 5）と、結果表示画面（図 6）を示す。

図 5 の入力画面ではペン入力デバイスを用いて、白いキャンパスの上に 1 行ずつ筆記する。

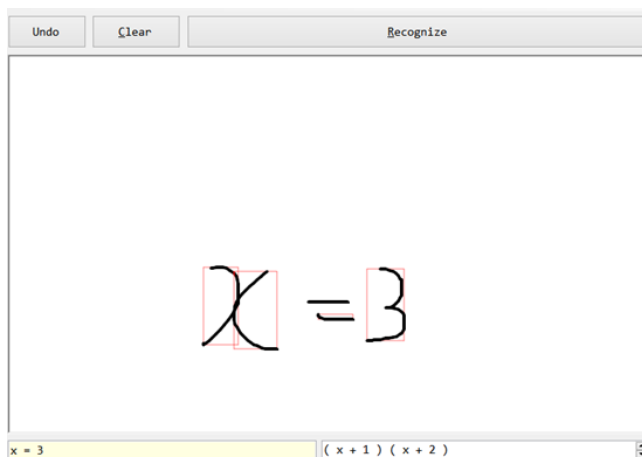


図 5. 入力画面

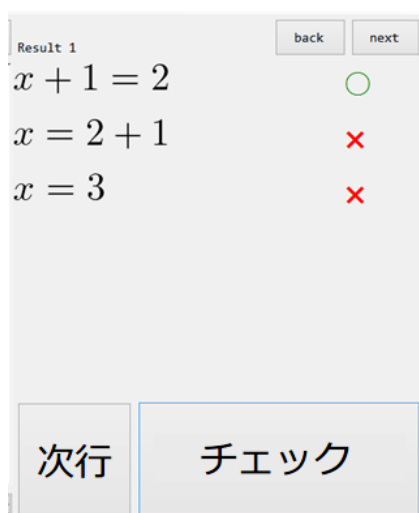


図 6. 結果表示画面

図 6 の結果表示画面は、入力画面と同一画面内に表示されており、数式手書き入力後、「チェック」を押すことで、認識からフィードバックまでの処理を行う。

学習者は与えられた問題に対して計算を行い、1 行書き

終わる度に、「チェック」を押す。「チェック」を押すと、筆記ストロークを認識した結果と、正誤フィードバック（○か×）が画面右側に表示される。「○」の場合、問題ないため「次行」を押し、続きの計算を行う。「×」の場合、直前の行に誤りがあるため、学習者はその行について誤りの推察を行う。誤り訂正の後、再び「チェック」を押して「○」であれば、「次行」を押し、次の行で計算を続ける。これを解答にたどりつくまで行う。

誤りの訂正方法は、1 ストローク毎に戻る機能を持ったボタンと、全てのストロークを消す機能を持ったボタンを使用して、誤答箇所を削除し、ペンで再記入する。

学習者が筆記したものと、認識結果が違っていた場合は、認識エンジンが認識候補を 11 個分用意しており、それを選択することで誤認識の訂正を行うことができる。候補の中にも学習者が書き記した数式がなければ、全てのストロークを消し、もう一度書き直す必要がある。

5. 考察

作成したプロトタイプにはいくつかの問題点や考察する点がある。

5.1 複数行

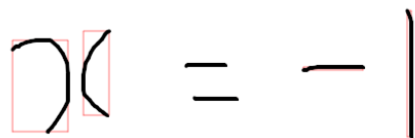
まず、1 行毎に正誤フィードバックを行うため、一度に全て解き、解答までたどり着いてからフィードバックを得たい場合に対応できない点である。1 行ごとの正誤フィードバックにユーザが煩わしく感じる可能性があると考えられる。そのため、複数行を一度に認識できるようにしなければならない。オンライン文字認識ではペンダウンとペンアップの位置情報から行の推定を行うことができるが、数式では分数等の文字列が複数あっても 1 行である場合が多く出現するため、容易に処理できる問題ではなかったため、今後のオンライン手書き数式認識の課題となっている。

他の方法で実現するためには、簡単な方法ではアフォーダンスを利用することが考えられる。入力部分を複数に分割し、それぞれの境目をノートの罫線のように表示することで、罫線の間にユーザが書こうとする。ユーザにとってはノートに数式を書く感覚を得ることができ、システム側では確実に行毎にストロークを得ることができるため、現状ではこれを採用している。

5.2 誤認識・誤採点

次に、誤認識による問題がある。誤認識をしてしまった場合、学習者は認識候補の参照を行うが、認識候補の中に解答が含まれてしまう可能性がある。テストでの入力に使用する場合は答えが分かってしまう危険性がある。当然テストでは解答のフィードバックは教師のみが閲覧できるものであるため可能性としては非常に低い。正しい解答がテスト受験者の目に触れることで、解答だと気づくケースも考えられる。例としては図 7 のような場合が挙げられる。

問題 $x - 1 = 0$ を x について解け



認識結果の候補

- | | |
|----------------|-----------------|
| (1) $x = -1$ | (7) $x \pm -1$ |
| (2) $x -- -1$ | (8) $2c = -1$ |
| (3) $7c = -1$ | (9) $2c -- -1$ |
| (4) $x = 1$ | (10) $7c = 1$ |
| (5) $7c -- -1$ | (11) $7c -- -1$ |
| (6) $x -- -1$ | |

図 7. 認識候補に正答が含まれるケース

図 7 は問題に対して移項ミスをした場合を想定したときの認識候補の一覧である。今回は認識結果の候補は 11 個まで類似度の高い順に番号を付けた。本来であれば、移行ミスとして減点の対象となるべき解答であるが、認識候補を見てしまった結果、候補(4)に正しい解答を発見し、移行ミスを修正するケースが考えられた。このように、システムによって不本意に解答を見せてしまう問題を発見した。

5.3 GUI

GUI の問題では、入力と結果表示を別々にしている点で視線移動を増やし、煩わしさを感じさせる原因になりうる事が考えられる。この解決方法として、認識結果はあえて隠し、フィードバックのみにする方法を考える。学習者が正答だと思うのにフィードバックが間違っていると判断した時は認識結果を閲覧できるようにする仕組みを取り入れることを考えた。手書きのストロークはチェック毎に消去し、筆記領域を確保しているが、学習者自身が書いた手描きの数式が残っていた方がより自然な形であるため、現状ではこれを採用している。

5.4 即時性

従来の場合、テスト返却には時間を要し、記憶の減衰によって、テスト内容を忘却し、復習が困難となる場合があった。テスト受験者の記憶が新しいうちに結果を返却することで、受験者はそのテストの内容を覚えたまま復習することができ、学習効果が高まると推察される。テストにおいてシステムが間違った採点をしていても、即時にフィードバックを返すことは利点がある。結果の返却が遅くなると、成績が既につけられてしまい、誤採点の訂正によって

成績の変更が困難となる問題があるが、即時フィードバックによって解決できる。他にも学校の入試試験での可否通知後に誤採点の発覚がある問題なども解決できると考えられる。ただし、その場合は答案者に解答を必ず返却できるような仕組みが必要となる。

6. おわりに

実用性のある数学学習支援あるいは数式自動採点を目指し、研究室で有し、改造できる手書き数式認識エンジンにより、手書きストロークを認識し、認識結果を Web API で同値判定した結果を即時フィードバックするプロトタイプを試作した。実用化されている認識エンジンと数式計算 Web サービスを利用することで、手書き数式採点や数学教育支援システムの実現のための実用的な基礎的機能を実装した。チェックボタン押下からフィードバック表示までの時間は、事前に用意した 10 個の手書き数式データで計測した結果、平均は 2.011 秒、最短で 1.29 秒、最長で 4.17 秒であった。プロトタイプを作成し、様々な問題を発見した。特に誤って解答したときに認識候補に正しい解答が出現するという手書き数式の自動採点における新しい問題を発見した。これらの問題を解決する方法を考察し、より実用性を高めた後、検証実験を行いたい。検証実験では、学習効果を測定するためにシステムを利用する場合と利用しない場合に分け、事前テストと事後テストを行い、その差を分析する。テスト結果だけでなく、ヒューリスティック分析や、エスノグラフィック分析等のアプローチ、ユーザビリティテストを行い、さらなる問題の発見やその改善を目指したい。

謝辞 数式認識エンジンを提供してくださった Le Duc Anh ら、東京農工大学中川研究室の数式認識研究班に深謝する。

参考文献

- 1) A.D. Le, T.V.Phan, and M.Nakagawa. "A System for Recognizing Online Handwritten Mathematical Expressions and Improvement of Structure Analysis", 11th DAS, pp.51-55, Tours France, April 2014
- 2) 鈴木雅人, 松本章代, 吉本定伸, 大塚友彦, "手書き数式解析に基づく基礎数学学習支援システムの開発", 電子情報通信学会技術研究報告. ET, 教育工学 109(453), 147-152, 2010-02-26
- 3) 大前佑斗, 福村好美, 中平勝子, "ICT を活用した積分演習の効率化手法に関する検討", 全国大会講演論文集 2012(1), 813-815, 2012-03-06
- 4) 鈴木雅人, 大川知, 松本章代, 吉本定伸, 大塚友彦, "基礎数学学習支援システムのための手書き数式解析法の検討", 電子情報通信学会総合大会講演論文集 2010 年_情報・システム(1), 150, 2010-03-02
- 5) 神宮英夫, 『スキルの認知心理学』, 川島書店, 1993
- 6) E.D.Gagne, 『学習指導と認知心理学』赤堀侃司訳, パーソナルメディア株式会社, 1989

記載内容の訂正＜正誤表＞

「手書き数式認識を用いた計算過程の正誤フィードバック」において、掲載内容に誤りがございました。皆様にご迷惑お掛けしましたこととお詫びするとともに、下記のとおり訂正させていただきます。

1. 英文の著者名欄(1 ページ目)

【訂正箇所】

誤	正
SHUNSUKE UEMURA	(空欄)

以上