Characterizing learner behavior from touchscreen data

Zachary A. Pardos 1,*, Leah F. Rosenbaum 2, Dor Abrahamson 3

University of California, Berkeley, United States of America

ARTICLE INFO

Article history:
Received 4 November 2020
Received in revised form 24 June 2021
Accepted 14 July 2021
Available online xxxx

ABSTRACT

As educational approaches increasingly adopt digital formats, data logs create a newfound wealth of information about student behaviors in those learning environments for educators and researchers. Yet making sense of that information, particularly to inform pedagogy, remains a significant challenge. Data from digital sensors that sample at the millisecond level of granularity, such as computer mouses or touchscreens, is notoriously difficult to computationally process and mine for patterns. Adding to the challenge is the limited domain knowledge of this biological sensor level of interaction which prohibits a comprehensive manual feature engineering approach to utilize those data streams. In this paper, we attempt to enhance the assessment capability of a touchscreen-based tutoring system by using Recurrent Neural Networks (RNNs) to predict students’ strategies from their 60hz data streams. We hypothesize that the ability of neural networks to learn representations automatically, instead of relying on human feature engineering, may benefit this classification task. Our classification models (including majority class) were trained and cross-validated at several levels on historical data which had been human coded with learners’ strategies. Our RNN approach to this difficult classification task moderately advances performance above logistic regression. We discuss the implications of this performance for enabling greater tutoring system autonomy. We also present visualizations that illustrate how this neural network approach to modeling sensor data can reveal patterns detected by the RNN. The surfaced patterns, regularized from a larger superset of mostly uncoded data, underscore the mix of normative and seemingly idiosyncratic behavior that characterizes the state space of learning at this high frequency level of observation.

Practitioner notes

What is already known about this topic:

• Observable sensor data on student behaviors can offer insight into their underlying understanding or conception of the task at hand.
• Real-time identification of pedagogically valuable behaviors is an especially useful role for digital learning tools, as this work often exceeds the capacity of an individual teacher with a classroom full of students.
• Recurrent Neural Networks (RNNs) have been used to model language and, in education, to predict the navigational behavior of learners in an online course.
• The computational field has struggled to apply RNNs successfully to lower-level data, such as eye-tracking outputs.

What this paper adds:

• Empirical evidence that different behavioral strategies can look very similar when enacted, underscoring the difficulty of this classification task.
• A behavior visualization technique to visually cluster and interpret results of the higher-dimensional model.
• Classification accuracy of RNNs used to model learner strategies from a high frequency sensor level (60hz) of data collection.

Implications for practice and/or policy:

• Aligned with theories of embodied cognition, it is not learners’ movements per se but rather the organizational structures underlying those movements that shape learning processes.

* Corresponding author.
E-mail address: pardos@berkeley.edu (Z.A. Pardos).

1 Zachary A. Pardos is Associate Professor at the Graduate School of Education. He directs the Computational Approaches to Human Learning lab, researching knowledge representation modeling and recommender systems in education.

2 Leah F. Rosenbaum is a Ph.D. candidate at the Graduate School of Education in the Learning Sciences and Human Development Program. Her research focuses on the design and evaluation of mathematics learning environments, particularly those involving collaboration and play.

3 Dor Abrahamson is a Professor at the Graduate School of Education in the Learning Sciences and Human Development program, where he directs the Embodied Design Research Laboratory. He researches practical and theoretical considerations around mathematics learning, particularly as guided by and contributing to theories of embodied cognition.

https://doi.org/10.1016/j.ijcci.2021.100357
2212-8689/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Tutoring systems, whether virtual or in-person, need to devote resources to identifying the organizational structures underlying students’ observable behaviors in order to most effectively adjust instruction.

The model's demonstrated ability to classify learning strategies means that such models could be used as part of real-time tutorial interventions in a computer-assisted learning environment.

1. Introduction

Student behaviors within a learning environment can offer a key glimpse into their work and reasoning processes. For example, a student who repeatedly deletes and re-types text into a free-response dialogue box may lack confidence in the subject matter, suggesting a need for greater support. A student who enters their responses quickly may need more challenging material. Digital learning environments offer particular promise for flagging and supplying these pedagogical supports at scale if the data logs they produce can be accurately processed.

We explore techniques to detect patterns in students’ high frequency touchscreen data in order to characterize their strategic behaviors within an intelligent mathematics tutoring program. The particular application we study is the Mathematics Imagery Trainer for Proportionality (MITp), a teaching tool for the concept of proportion (see Sec. The Proportionality Tutor App). This application prompts students to move two markers up and down on a touchscreen, signaling the desired height ratio between the left and right touch positions by changing the screen's color. When working with a small number of students, strategies can be classified by human experts either live or from video recordings. However, if the app were used at a larger scale, automatic detection of these strategies would be necessary. The telemetry data from the app may be an essential source for making these automatic classifications. The data consists of these two touch positions, sampled at 60 Hz, and a human-ascribed label of the students’ interaction strategy. Our data set also includes a larger set of student telemetry without strategy labels.

We use a variety of data segmentation techniques to prepare the telemetry data for machine learning models to classify students’ interaction strategies. The machine learning models include Long Short-Term Memory, meant to learn patterns of movement corresponding to these strategies from the raw timeseries data, and multinomial logistic regression, meant to serve as a contrasting simpler classification model. We also apply nascent visualization techniques to synthesize behaviors with the tutor through the prism of a behavioral Recurrent Neural Network (RNN) model that includes unlabeled student data. This work aims to explore the classification of high frequency data using deep learning methods and to extend previous research (Abdulrah, Adil, Rosenbaum, Clemons, Shah, Abrahamson, et al., 2017) in order to enable an intelligent virtual tutor to more effectively guide students through the target learning activity.

2. Background

High frequency sensor data has been used to adapt instruction or study learning processes in a variety of contexts. For example, learners’ affective state, as determined by galvanic skin response, facial feature recognition, and posture analysis, has been used to embed emotional supports and adjust lesson difficulty in intelligent tutoring systems (Arroyo, Cooper, Burleson, Woolf, Muldner, & Christopherson, 2009; Arroyo, Mehranian, & Woolf, 2010) and to predict learning outcomes (Joshi, Allesio, Magee, Whitehill, Arroyo, Woolf, Sclaroff, & Betke, 2019). Eye-tracking technology has been used to study how learners use graphical representations to construct knowledge (Rau, Bowman, & Moore, 2016), to diagnose cognitive traits (Rapits, Katsini, Belk, Fidas, Samaras, & Avouris, 2017), to identify domain-relevant gaze patterns that precede verbal explanations (Abrahamson, Shayan, Bakker, & Van Der Schaaf, 2015), and to understand the emergence of joint-attention in human tutoring scenarios (Shvarts & Abrahamson, 2019). And a host of physiological measures have been used to inform and adapt intelligent tutoring systems (Lane & D’Mello, 2019). In these works, the sequence data were typically pre-processed to a coarser level of frequency before use. The use of these data has been referred to as multimodal learning analytics (Abdulrah, Baker, & Heffernan, 2017; Jiang, Bosch, Baker, Paquette, Ocumpaugh, Andres, Moore, & Biswas, 2018), trace skills in a tutor (Piech, Bassen, Huang, Ganguli, Sahami, Guibas, et al., 2015), and to anticipate events (Tang, Peterson, & Pardos, 2017) and completion outcomes (Körösi, Esztelecki, Fors, & Tóth, 2018) in MOOCs. Another common application is to eye-gaze data, used to predict attention levels on educational videos (Hutt, Mills, Bosch, Krasic, Brockmole, & D’Mello, 2017), model teacher classroom activity, and student behavior in educational games (Emerson, Sawyer, Azevedo, & Lester, 2018). One study used deep neural networks to analyze posture and facial features to predict mind-wandering during instructional sessions (Bosch & D’Mello, 2019) and another used them in concert with depth cameras to provide feedback in a CPR tutor (Di Mitri, Schneider, Trebing, Sopka, Specht, & Drachsler, 2020). Research applying deep-nets to higher frequency data has been sparse, with nascent work finding only null results in predicting student future performance based on tutor eye-tracking data (Rau & Pardos, 2016).

The study is situated within a larger research program at the Embodied Design Research Laboratory (Abrahamson, director) seeking to evaluate the relevance of the “embodiment” turn in the cognitive sciences (Anderson, 2003) and human–computer interaction (Dourish, 2004) for developing theoretical models of mathematical cognition, teaching, and learning as well as educational resources. Research efforts have centered on collecting and analyzing empirical data of study participants’ individual interactions with the Mathematics Imagery Trainer for proportionality (MITp) (Howison, Trinic, Reinholz, & Abrahamson, 2011). The MITp is an activity genre in which students learn to move physically in new ways believed to create the sensorimotor basis of targeted mathematical concepts, such as proportionality, even before encountering normative symbolic forms such as numerical notations (Howison et al., 2011).
The process of designing and implementing this technological application in a variety of interaction platforms has created an empirical context by which to revisit and corroborate seminal philosophical and theoretical approaches to research on education, including the works of Francisco Varela (Hutto, Kirchhoff, & Abrahamson, 2015), Jean Piaget (Abrahamson, Shayan, Bakker, & Van der Schaaf, 2016), and Vygotsky (Shvarts & Abrahamson, 2019) as well as ethnomethodological approaches to conversation analysis (Flood, 2018; Flood, Neff, & Abrahamson, 2016).

2.1. The proportionality tutor app

Our data were obtained from the MITp application (Howison et al., 2011), collected as part of an effort to provide automated tutoring to children using a virtual agent (Fig. 1). The MITp is an activity designed to support students in learning the concepts of ratio and proportion, important yet difficult topics for many students. Understanding proportionality involves appreciating multiplicative relations between extensive quantities; a change in one quantity is always accompanied by a change in the other, and these changes are related by a constant multiplier (Boyer & Levine, 2015; Lamon, 2007). The MITp approach to support students in developing multiplicative understanding of proportions draws on embodiment theory, which views the mind as extending dynamically through the body into the natural-cultural ecology. Thus, human reasoning emerges and is expressed through situated sensorimotor interactions (Abdullah et al., 2017). The MITp (Fig. 2) poses the physical coordination challenge of moving two hands on a touch screen to make it green, a result which occurs when the ratio of hand heights matches the pre-programmed ratio of 1:2. Through engaging in this embodied-interaction activity and building particular movement schemes related to proportions, students can develop a pre-symbolic quantitative understanding of this mathematical notion. By then introducing specific tools into the environment, here a grid and numbers, students receive progressively more mathematical tools with which to express those strategies.

The MITp has traditionally been operated by a human technician who explains how the system works, sits beside the child, and makes suggestions as they interact. This technician also decides when to introduce scaffolding artifacts, such as grid lines or numbers. Interaction sessions can last an hour and require a trained technician to guide each student through every step of the process, significantly limiting the feasibility of scaling up the MITp to large audiences. Previous work introduced a virtual pedagogical agent with a limited ability to guide students through the process (Abdullah et al., 2017). It remains an open challenge to enable automatic assessment at the strategy level and to provide subsequent appropriate instruction, valorization, feedback, correction, prompts to reflect, and introduction of new artifacts and challenges. The nature of strategies exhibited during interaction with MITp have been deduced through manual analysis of data streams collected during tutoring in previous work and were determined to be helpful transition states in students’ learning processes (Abrahamson, Lee, Negrete, & Gutiérrez, 2014). Three distinct strategies were identified as the most important and are described below.

In trying to achieve a green screen (and thus the 1:2 ratio of cursor positions), learners can discover “the higher, the bigger” strategy: that the gap between their hands is bigger as they make the same proportion higher up on the screen. Moving one hand and then adjusting the other leads to the “A-per-B” strategy: that when they move their left hand one unit, students must move their right two units (for 1:2) to achieve green. When learners create continuous green, they learn the “speed” strategy, that the right hand’s speed is twice that of the left hand.

Determining which strategy a child is using at any given moment both offers great insight into their learning process and is highly relevant in determining the most effective response for a pedagogical agent to undertake. For example, stasis in a “higher, bigger” strategy or reversion to “higher, bigger” after demonstrating “A-per-B” could indicate a need for correction or re-focusing. Successful and sustained performance of a strategy can indicate that it is time to encourage the child to explore other strategies. Successful performance of the set of strategies can indicate that it is time to introduce new artifacts and advance to the next stage in the tutorial process.

The MITp system has been extensively tested for its educational effectiveness. Using qualitative analyses, researchers demonstrated a variety of manipulation strategies students developed to accomplish the task objective of moving their hands while keeping the screen green (Howison et al., 2011). Moreover, students engaged in deep mathematical reflection as they were guided to compare across the strategies (Abrahamson et al., 2014). These studies have presented empirical data of students shifting from naïve manipulation to mathematical reasoning as they engage the frames of reference introduced into the problem space. These studies also illustrate the tutor’s critical role in facilitating this shift (Abrahamson, Gutiérrez, Charoening, Negrete, & Bumbacher, 2012).

2.1.1. Development context and theoretical grounding

The study is situated within a larger research program seeking to evaluate the relevance of the “embodiment” turn in the cognitive sciences (Anderson, 2003) and human–computer interaction (Dourish, 2004) for developing theoretical models of mathematical cognition, teaching, and learning as well as educational resources. Research efforts have centered on collecting and analyzing empirical data of study participants’ individual interactions with the MITp (Howison et al., 2011).

Previous studies required a trained human tutor to facilitate students’ optimal engagement with the interactive technology, such as orienting the students toward selected regions of the perceptual display, reminding them of their own earlier observations, encouraging them to persist in a particular line of exploration (Abrahamson et al., 2012), or co-constructing with them mutually intelligible articulations of emergent mathematical constructs (Flood, Harrer, & Abrahamson, 2017). To support wider deployment of these learning frameworks, a virtual pedagogical agent with speaking and gesturing capabilities was integrated into the app. This virtual agent’s interaction repertory was drafted on vast audio–video data of human tutor activity from earlier work (Abdullah et al., 2017; Flood et al., 2016; Flood, Schneider, & Abrahamson, 2014). The virtual agent employed a simple rule-based regime to trigger particular scaffolding responses.

In this paper, we seek to augment the virtual agent’s capabilities to include assessment of students employing particular strategies. A key challenge herewith is that hand movements...
are external manifestations of different internal sensorimotor schemes (Abrahamson & Bakker, 2016). Consider, for example, the movement of a rising hand. This movement can be described objectively in terms of the kinematics or event structure of a material object (the hand) following a particular dynamic trajectory through space. Further assume that this movement satisfies some interaction trigger of an interactive platform. What the platform would not necessarily encode is how the individual oriented perceptuomotorically toward the situation such that they were able to enact this particular movement. There are multiple perceptuomotor schemes that would result in moving one's hand that way, such as visualizing the hand as being pushed from below or as pulled from above (Abrahamson and Shulman, 2019). Whereas an individual’s subjective phenomenology of enacting a movement may be irrelevant to triggering some target response in a technological platform, such as flicking on a light switch, it is of great importance for a pedagogical regimen that seeks to ground mathematical concepts in perceptuomotor schemes, because these alternative subjective constructions of the same objective events bear on the mathematical meanings inherent to performing the movement, for example addition vs. subtraction. Therefore, the human or virtual tutor supporting the students’ learning process need to go beyond registering what movements the student is performing to understanding how the student is moving that way. If the pedagogical agent can accurately understand the student’s thought process, it can provide more effective guidance.

3. Data sources

We used students’ time-stamped interaction sequences that record the student touch positions as they progress through instructional prompts, guided by the virtual tutor. As illustrated in Table 1, the data set contains 49 students’ files with the following variables:

1. Left-y and Right-y indicate the touch positions of the left most finger and the right most finger on the 2-D coordinate plane of the touchscreen. In practice, only the Y values are used, as the task criteria uses only the ratio of the two heights.
2. Color indicates how close the screen is to the goal color of green. When the student has achieved the target ratio of 1:2 between the heights of her two fingers, the color value was 1.
3. Time shows time-stamp taken from the data stream file, measured to the second. The data were collected at the 60Hz level and were logged in proper sequential order, in spite of the less granular time stamp.

4. ID is a student-specific identifier.
5. Prompts indicate the task given to the student by the tutor (examples in Table 2).
6. Label is the human-coded strategy exhibited by the student at various segments throughout their interaction with the tutor. Labels are described in Table 3. Analysis will focus on strategy labels only. Of the total 49 files, only five included human-ascribed strategy labels due to the significant time cost of manually annotating these data.

The length of each data file ranged from 39,736 to 176,283 entries, capturing between 11 and 48 min of interaction, sampled at 60 Hz. For the training of our label prediction models, we used only the 5 students with human-labeled strategies. Descriptive statistics of these 5 students are displayed in Table 4. Two students did not make it to the last of three phases of the tutor session, and thus never demonstrated the SP (speed) strategy. While we could have filtered this out, low subject count and missing values are commonplace real-world classification tasks. We chose to include the SP (speed) label to keep the task authentic and because it was an important label to predict within the tutor’s pedagogical scheme. While filtering out students with such missing data may be commonplace when more subjects are on hand, further reducing our subject pool would not have been tenable.

4. Methods

In this section we describe the classification models we used to predict students’ movement strategies. Our baseline is predicting the majority class. Our two statistical models are a simple multinomial logistic regression and a variant of a Recurrent Neural Network called Long Short-Term Memory (LSTM). We describe a variety of cross-validation methods which correspond to different ways in which the tutoring algorithm could segment and predict students’ interaction strategies. Lastly, we describe a data visualization technique built upon the classic RNN model and outline its use to explore patterns in data from the 44 remaining students whose strategies were not hand-coded by human researchers.
4.1. Multinomial logistic regression

We used a simple neural network equivalent to a standard multinomial logistic regression classifier for students’ interaction strategies. We used six features as the inputs to the model:

1. The left hand-y coordinate.
2. The right hand-y coordinate.
3. The student ID represented as a one-hot dummy variable.
4. Whether or not at least one finger was touching the screen.
5. Instructional prompt represented as a one-hot dummy variable.
6. A Boolean representing if the student had reached the goal state of exhibiting the 1:2 ratio within the tutor’s specified margin of error.

We define our neural multinomial logistic regression model as follows:

\[ z = \theta^T x \]  
\[ \text{softmax}(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \text{for } j = 1, \ldots, K \]

where \( x \) is a vector representing the input feature values outlined above for a single student and time slice. \( \theta \) is a matrix of coefficients associated with each input variable with respect to each of the labels. \( \theta^T \) is the transpose of the matrix used for the multiplication of matrix \( \theta \) and \( x \) to produce \( z \). The softmax function is an activation function which produces a probability distribution over the K labels. The label with the highest probability is the predicted label.

This neural multinomial logistic regression model represents the degree to which a label can be predicted by a linear combination of the student, prompt, and instantaneous value of the left and right hand positions in that time slice. In contrast, the RNN-based method, described in the next subsection, is hypothetically better equipped to identify a pattern characterized from a series

\[ \text{of finger movements exhibited over time. The difference in performance between the two will suggest the degree to which the LSTM is able to learn sequential features essential to the strategy classification task.} \]

4.2. Long short-term memory augmented RNN classification

Long short-term memory (LSTM) is an augmentation to the classic Recurrent Neural Network (RNN) model (Medsker & Jain, 2001) and was first proposed by Sepp Hochreiter and Jürgen Schmidhuber (Hochreiter & Schmidhuber, 1997). It maintains a hidden state and a “longer-term” cell state. A persistent problem with RNN models is a phenomenon known as the vanishing gradient, in which an RNN can struggle to utilize observations the further back in time they have occurred. Within our virtual tutor scenario, the vanishing gradient would limit the tutor’s ability to detect student demonstrations of particular strategies that manifest over longer periods of time. Through its architecture, LSTM has demonstrated the ability to classify patterns based on longer sequences than the classic RNN, diminishing the vanishing gradient.

To predict student interaction patterns within MITp work, we use an LSTM model with the times series of left and right hand coordinates as inputs and the hand coded labels as categorical outputs.

The LSTM model, depicted in Fig. 3, involves input gates, forget gates, output gates. These gates work in unison to determine what aspects of past observations are kept, forgotten, and combined with newer observations. As inputs, we use the x- and y-positions of both touch points, sampled at 60 Hz (4 features in total). As ground truth, we use the human-ascribed strategy labels. This model predicts the current strategy label either at every time step or for an entire action sequence, depending on the specific setting discussed below in the Evaluation section.

Because some instances in the time series are unlabeled (e.g. when the student was not touching the screen), we use a default value of \(-1\) as a place-holder label to pad all the instances when no real label exists. A custom loss function is defined to only calculate the loss, or model fit error, when the predicted label takes on a value other than this default value. A small search of the LSTM’s hyper-parameters (Greff, Srivastava, Koutnik, Steunebrink, & Schmidhuber, 2017), or user-tunable parameters, was conducted where the dimension of the hidden state (e.g., 64, 128, 256) and stochastic gradient descent fitting algorithm (e.g., RMSprop, Adamax, Adagrad) were varied.

The programmatic framework used to define the models and run the experiments was python’s Keras (Chollet, 2015) using Theano (Theano Development Team, 2016) as backend.

Table 3
Strategy label definitions. The labels indicate various ways in which students find or maintain the proper ratio between touch positions, thus keeping the screen green.

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Drag one hand</td>
<td>Keeping one hand still, exploring with the other.</td>
</tr>
<tr>
<td>AB</td>
<td>A per B</td>
<td>Step-wise cursor movement</td>
</tr>
<tr>
<td>SP</td>
<td>Speed</td>
<td>Simultaneous cursor movement</td>
</tr>
</tbody>
</table>

Extended labels
<table>
<thead>
<tr>
<th>Non-strategy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT</td>
<td>No touch</td>
</tr>
<tr>
<td>IT, RT</td>
<td>Initial touch, release touch</td>
</tr>
<tr>
<td>T, NV</td>
<td>Tuning, not visible</td>
</tr>
<tr>
<td>H, O</td>
<td>Horizontal movement, other</td>
</tr>
</tbody>
</table>

Table 4
Descriptive statistics of the 5-labeled students showing the number of time slices coded with each of the three strategy labels.

<table>
<thead>
<tr>
<th></th>
<th>AB</th>
<th>D</th>
<th>SP</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL</td>
<td>1740</td>
<td>15780</td>
<td>5940</td>
<td>23460</td>
</tr>
<tr>
<td>ER</td>
<td>5100</td>
<td>3960</td>
<td>0</td>
<td>9060</td>
</tr>
<tr>
<td>KN</td>
<td>6180</td>
<td>2700</td>
<td>8040</td>
<td>16920</td>
</tr>
<tr>
<td>MS</td>
<td>4680</td>
<td>6120</td>
<td>4140</td>
<td>14940</td>
</tr>
<tr>
<td>ND</td>
<td>14640</td>
<td>13500</td>
<td>0</td>
<td>28140</td>
</tr>
</tbody>
</table>
4.3. Behavior visualization using classic RNNs

Having defined a model to predict student strategy labels based on their touchscreen telemetry data, our second modeling objective was to better understand the touchscreen behavior patterns through which students transitioned. Here, we model behavior using a classic RNN instead of an LSTM augmentation. Whereas the LSTM distributes its representation across several states (the hidden and cell states), we wish to view the models’ complete description of the hidden state from a single layer, as is possible with a classic RNN. We also switch objectives from predicting the students’ next strategy label to predicting their next set of cursor coordinates. Without the requirement for a label ground truth – available for only 5 students – we can train our model on all 49 students in our dataset.

Classic RNNs are used in many tasks related to sequence prediction, including speech recognition (Graves, r. Mohamed, & Hinton, 2013), handwriting recognition (Graves & Schmidhuber, 2009), and sentence completion (smart keyboards) (Mirowski & Vlachos, 2015). Their architecture is similar to Fig. 3 except that the cell state and related gates are not included. The recurrent hidden state $S(t)$ is updated to:

$$S(t) = \sigma(WS_{t-1}) + Ux_t + b_s$$ (3)

$$O(t) = \text{Softmax}(VS_t + b_o)$$ (4)

where $x_t$ is the input vector (e.g. x and y touch position, screen color, time) of current time step, $S_t$ is the hidden layer vector that represents past observations in order to predict future actions, and $O_t$ is the output vector of the next x and y hand position. $W$, $U$ and $b$ are weight coefficient matrices and bias vectors. The $U$ matrix describes how the input will be transformed into a hidden state representation. The $W$ matrix describes how the hidden state representation will transform from one time slice to the next. The hidden state at any given time slice is the combination of the past transformed hidden state and the new transformed input observation. As above, $\sigma()$ and Softmax() are activation functions that dictate the range of the final output (e.g., sigmoids squashes the output range to a probability between 0 and 1 for each label, softmax distributes the probabilities across each label), and $t$ is the time step.

We trained the RNN by using the y-coordinates of both touch points, left and right, in previous time steps as input to predict the y-coordinates of both touch points in the next time steps (in Fig. 3, $y_1$ and $y_2$ of time $t$ is the input and $y_1$, $y_2$ of time $t+1$ is the output). The activation function applied to the hidden layer is the hyperbolic tangent (tanh). A dropout layer with rate 0.5 was set between the hidden layer and the output layer, a regularization technique frequently applied to reduce the ability of the model to become overfit to the training data. No activation function was implemented to the output layer because the original, non-normalized, touch positions were used. To suit the coordinate output, the loss function used for training the model was mean squared error (MSE). The maximum epoch, or number of times the fitting process iterates through the data, was set to 100.

After training on all 49 students, the 5 labeled students were run through the model. Their hidden states at every time step were saved and reduced to two dimensions using t-Stochastic Neighborhood Embedding (t-SNE) (van der Maaten, Hinton, & Bengio, 2008). The outcome is a visualization of the state space, synthesizing our 5 labeled students’ behaviors as regularized by all 49. Since there is no established rule as to the appropriate RNN hyper-parameters (e.g., number of layers and number of nodes per layer) to use for bringing out visually salient patterns, part of our methodology involved finding such values using a limited range of hyper-parameters to create a set of 96 models (time lags: 5, 10, 20; batch sizes: 32, 128; optimizers: Adagrad, RMSprop; hidden sizes: 64, 256; sequence chop: by prompt, no chop). This produced 96 respective lower dimensional visualizations which were rated for their usefulness on a five point scale.
in similar fashion to Geryk (Géryk, 2015). We chose the model that produced the highest plot score and visualized it using our d3 tool, d3-scatterplot to observe behavior patterns. An important feature of t-SNE is that it prioritizes retention of data points close together in the high dimensional space to also be close in the lower dimensional space. Of lower priority is that points far away in the high dimensional space also be far in the lower dimensional space.

5. Evaluation

We designed a set of experiments to evaluate different generalizing properties of the LSTM-augmented model. All experiments were tested on both the expanded label set of all 49 students as well as the subset of 5 students with strategy labels. In addition to training the models on each student’s entire telemetry sequence, we also partitioned the telemetry sequences into different segments (or chops) of shorter lengths. We sought to explore whether the model would better generalize when training on a greater number of shorter sequences instead of a smaller number of longer sequences. We also varied the frequency of the predictions between (1) predicting the label at every time slice (60hz) and (2) predicting the presence of the label in the sequence at the very end of the sequence, as defined by the segmenting. Accuracy was calculated based on the aggregate performance of all binary predictions. The setup for this sequence prediction model was the same as the time slice LSTM except that instead of a softmax function applied over all the labels at every time slice, an independent prediction of the presence of each label was made at the very last time slice of the sequence. Five-fold cross-validation was conducted at both the student level and segment level for all experiments. The data were separated into 5 folds, each serving as the test set once, with the remaining folds serving as the training set. The results from the 5 phases were averaged to produce a single accuracy evaluation metric.

5.1. Sequence segmentation

In addition to using the original whole sequence, we segment the student telemetry data in two ways.

5.1.1. Segment by prompt

During the tutoring sessions, students are given prompts by the tutor directing them to adjust their movements. Accordingly, there is a column named ‘prompt’ in each student’s data file that indicates the specific instructions given at that time period (see Table 1). A segment-by-prompt approach defines a segment from the start of a prompt to the start of the next prompt, normally including some time after the prompt during which the student is interacting with the system and the tutor is silent. Segmenting by prompt produced 162 total segments of varying lengths.

5.1.2. Segment by label

Using video recordings of the tutoring sessions, domain experts labeled student interaction strategies based on observable movement patterns. The segment-by-label approach divides the sequences using these expert-ascribed labels and produced 1,239 sequences of various lengths. This level of segmentation would not be available in a real-time scenario as it requires knowledge of the label beforehand. Nevertheless, this technique serves as a test of smaller sequence length for classification.

5.2. Hypotheses

We enumerate the following expectations for the results based on our evaluation approaches:

- **H1**: Compared to logistic regression, the LSTM-augmented RNN has the ability to learn chronological information in the sequence. Therefore, we expect that the LSTM-augmented RNN models will perform better on average than logistic regression models.

- **H2**: Different students may exhibit the same strategies in different ways. Therefore, we anticipate that models that have trained on data from the student they are predicting (sequence level cross-validation) will perform better than their respective student level cross-validation experiments.

- **H3**: We assume that predicting the label at the frequency of every sequence will be more accurate than predicting the label at every time step. At the sequence level, the model need only predict whether or not a label occurred rather than the temporal sequencing in which it occurred.

- **H4**: After restricting to the strategy label set, the classifications should be easier to learn compared to the expanded label set since there will be less opportunity for similar labels to be confused for one another.

6. Results

6.1. Strategy label classification

We briefly review the results of the student label evaluation. For more detail, please see (Pardos, Fan, & Jiang, 2019). In all but two of the experiments, the LSTM outperformed the logistic regression model, mostly confirming H1 (see Table 5). Contrary to H2, the sequence level cross-validation (bottom half of Table 5) outperformed its student level counterpart (top half of Table 5) in only a few experiments (e.g. Sequence > sequence > prompt of 89.8 is more accurate than Student > sequence > prompt of 79.5). This result suggests that perhaps identifying each student’s “signature” is not critical to predicting their behaviors based observations within the training set. As expected, classifying labels at the frequency of sequence (H3) was the easier classification task than at the time step frequency (e.g., Student > sequence and Sequence > sequence have higher accuracy than Student > time step and Sequence > time step), only performing worse than time-step frequency in one experiments (e.g., Sequence > sequence > label of 33.0 is lower than Sequence > time step > label of 39.0). Finally, in comparing the 10 LSTM results of the strategy label set to the expanded label set, the average predictive accuracy in the extended label set was 41.66% whereas the average predictive accuracy in the strategy label set was 54.65%. The advantage in predicting strategy labels suggests a more consistent set of behaviors when performing these strategies, mostly confirming H4.

6.2. Behavioral analysis visualization

A benefit of modeling students at the 60hz telemetry level is the ability to discover patterns of movement captured in an unsupervised fashion and reflect on the sufficiency of our expert labels and glean insights into why a particular label prediction might have been challenging for the model. For this analysis, we used a model trained on all 49 students to forecast their left and right hand positions. Figs. 4 and 5 represent the hidden state space, or hand movement pattern space, visualized using t-SNE of each time slice for each of the 5 labeled students. If we were to train on only the 5 labeled students, there would be a
Table 5

<table>
<thead>
<tr>
<th>CV_by</th>
<th>Frequency</th>
<th>Sequence</th>
<th>chop</th>
<th>B</th>
<th>LR</th>
<th>LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>Time step</td>
<td>None</td>
<td>31.2</td>
<td>39.29</td>
<td>47.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>31.2</td>
<td>39.29</td>
<td>45.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prompt</td>
<td>39.29</td>
<td>42.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence</td>
<td>None</td>
<td>66.6</td>
<td>NA</td>
<td>86.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>47.1</td>
<td>NA</td>
<td>46.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prompt</td>
<td>71.7</td>
<td>NA</td>
<td>79.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence</td>
<td>Time step</td>
<td>None</td>
<td>31.2</td>
<td>40.6</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>15.9</td>
<td>24.88</td>
<td>39.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prompt</td>
<td>43.4</td>
<td>43.5</td>
<td>37.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence</td>
<td>None</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>47.5</td>
<td>NA</td>
<td>33.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prompt</td>
<td>73.7</td>
<td>NA</td>
<td>89.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4. Visualization of the RNN states at every time slice of the five students from the classification analysis, colored by student. Regions of the figure labeled with a letter were identified manually as visually salient and referenced in the text. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A greater chance that the hidden state visualizations were a product of overfit. By training on all 49, we have greater confidence that the regularities being depicted are more normative, as they are learned from training set nearly ten times larger than the classification set. This model is intended to predict where on the screen a participant will place their fingers next (“In the next second, where will the student touch given where they touched in the past?”). Interestingly, this model was not trained based on strategy label information but was instead trained on location information.

In Figs. 4 and 5, each dot gives a 1-second snapshot of a participant’s touch behavior. Overlapping or snaking dots show a progression of the movement a student is making. The axes give a 2-dimensional simplification of the higher-dimensional space of the RNN model. Dots appear next to each other in this visualization if they were also close to each other in the higher dimensional state space as determined by the model.

The state space has been manually annotated with clusters for which we observed to contain both salient visual features and plausible interpretation. These clusters are circled in the figures and labeled a through h for reference. These clusters fall into three categories. Category (1) clusters contain different students who exhibit similar behaviors in the space corresponding to similar strategies (clusters d, e, f and g). Examples of this are students ND and KN, purple and green, respectively, in Fig. 4. They perform the exact same strategy AB, which is colored as orange in Fig. 5 (clusters d, e, f, and g). Category (2) clusters include the same student performing different strategies. For example, student ND (in cluster a of Fig. 4) presents two strategies in cluster a of Fig. 5, D and AB, which are very close to one another but still
distinguishable. Category (3) clusters such as b, c, and h contain different students presenting different strategies across the two plots. In cluster h, those students clustered together in the label-colored plot (Fig. 5), showing that they exhibit similar patterns when performing the three goal movements.

7. Discussion

Accurate labeling of students’ dynamic solution strategies is critical for the efficacy of interactive learning design, because these labels inform the automatic selection of optimal tutor responses. The pedagogical principle is that tutor responses keep the student as close as possible within their current parameter space of (bimanual) action by steering them to explore new values along existing parameters or orienting their attention toward a new parameter. This pedagogical regimen has been found to accord with constructivist principles (e.g., Shvarts and Abrahamson (2019)). As such, for an AI system to respond attentively to students within their micro-zone of proximal development, the system needs validated algorithms for ongoing modeling of how the students are attempting to solve the problem at hand. Such was the objective or the LSTM model.

Methodologically, the application of LSTM-augmented RNNs to our dataset of 60hz touchscreen sensor data was successful in realizing a moderate gain in classification performance. It achieved 47.1% accuracy in predicting the moment-by-moment strategy being employed by the student, the accuracy climbs to 86.7% with an RNN, compared to 66.6% using the majority observation (or non-observation) of each label across students in the training set. Inspection of the state space of these behaviors sheds some light on these moderate generalization results by summarizing behaviors, some of which exhibited regularities corresponding to the label, while others manifested as idiosyncratic and could not be generalized.

Even with the improvements achieved by the LSTM model, predicting interaction strategy remains a challenging task. In our view, this difficulty somewhat explains the messiness of the behavior visualizations shared in Figs. 4 and 5. Were the label classification task easier, we would expect more category (1) clusters (e.g. d and e in Figs. 4 and 5) where multiple students perform the same behavior and are identifiable as such by the model. We interpret category (1) clusters as suggesting that different students exhibit similar behaviors for a given coded strategy. The patterns in cluster category (1) most lend themselves to generalizing the prediction of strategies. Category (2) type clusters, in which one student displays two or more strategies proximal in the model space, may suggest that a student is on the verge of adopting a new strategy. Clusters and transitions of this type bear pedagogical significance and may merit special attention within educational protocols, if they can be identified in real-time. Finally, type (3) clusters represents cases that would be most difficult to generalize. Category (3) clusters contain behaviors that look similar to the model, that is, they are adjacent or overlapping within the model space but which are coded with different labels (colors). We note that the definition of strategies D and AB are somewhat overlapping (you must drag a finger to demonstrate A-per-B), somewhat explaining areas where those two strategies overlap (e.g. blue and orange in clusters a and c of
5. Such overlapping of colors in Fig. 5 suggests shared behaviors across those strategy classes. Still other behavior patterns, such as the ones in the upper left, suggest a degree of idiosyncratic behavior not shared among students. More work remains to develop models that build off such behaviors to predict subsequent touch and strategy.

Intriguingly, students’ behavioral patterns detected by the model yet unfamiliar to the researchers need not indicate the model’s failure to match the researchers’ observations. Rather, the model may be picking up interaction strategies yet-undetected by the researchers, such as intermediary strategies that, while falling short of solving the problem, are necessary as exploratory stepping stones toward a systemic reconfiguration (Tancredi, Abdu, Abrahamson, & Balasubramaniam, 2021). As such, the model’s predictive “lemons” may turn out to be research “lemonade”, demonstrating a methodological tool rarely if ever used in educational interaction design: a design process that enlists LSTM models to work reciprocally with human analysts, informing their ongoing—rather than only a posteriori—detection of interaction strategies.

8. Conclusions and future work

Students devise a broad variety of sensorimotor schemes for enacting a target movement (Abrahamson & Bakker, 2016; Abrahamson et al., 2014, 2016; Howison et al., 2011), in some cases with different underlying organizational schemes for seemingly the same observable movement patterns. Knowing what schemes students are employing is critical for supporting their learning, and yet determining these schemes within virtual tutoring environments has been a challenging engineering task. The difficulty that we encountered in modeling students’ schemes thus corroborates expectations coming from constructivist and enactivist theories, viz. that the mind and thus learning is highly individualistic and thus difficult to model in terms of generalizable qualities. Nonetheless, progress was made in establishing the improved performance of LSTM-augmented RNNs, which produced predictions that would be actionable in the MITp application. Most notably, the accuracy of A-per-B label provides a signal that the tutor can use to determine if the child is ready to advance to the more complicated strategies. And as the behavior visualization results of cluster type (1) indicate, these models can identify consistencies in certain strategy enactment across students.

Several future directions for this work may prove profitable. The MITp tutoring protocol includes several phases: guided exploration, using the A-per-B strategy, and then using the speed strategy. Including information about the current phase of the tutoring process in the analysis may improve labeling prediction, as different labels are more likely in different phases. The importance of accuracy for different labels also varies depending on the phase, so results could be more effectively interpreted if this information is included. A different approach would be to include additional multimodal input data. Eye-tracking analyses may offer one promising direction (Abrahamson et al., 2016), if applied in real-time and in concert with the touchscreen stream. Incorporating eye-tracking could also take us beyond the hand-coded ‘strategy’ labels, which represent what a student is performing, objectively, to the interpretation of how a student is orienting toward the enactment of a movement. Furthermore, it is expected that a larger set of labeled students and the ability to limit the imbalance of the speed label would improve classification accuracy, justifying deeper integration of these models in the tutoring system. Finally, we are excited by the new conjecture coming from this study, that the dark space of the model’s prediction failure might, in fact, shed light on students’ behavioral patterns yet unknown to the human researchers. That is, the model’s alleged failure is not bad reading of good strategy but, rather, good reading of bad strategy—the model is surfacing typical behavioral patterns that do not yet accomplish the task performance yet are vital for the learning process. If this conjecture bears out, it would suggest integrating the models themselves into the software’s real-time interaction regimen to illuminate for instructors—whether human or artificial—the dark matter of learning (Lindwall & Lymer, 2008).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Financial support for this work was partially provided by the National Science Foundation, United States of America through grants IIS 1320029, IIS 1321042, and IIS 1547055.

Open data

Under this study’s Internal Review Board protocol, even anonymized data may be shared only with members of the research team, unless expressly permitted by Media Release documentation. To access the data used for this study, contact Dor Abrahamson (dor@berkeley.edu). The interested researcher would need to be added to the research protocol as a member of the research team.

Ethics

This study was conducted in accordance with the guidelines set forth by the University of California, Davis Internal Review Board, as executed by the UC Davis Office of Research (IRBNet# 472639-11).

References


