Title
Analysis of Student Behaviour in Habitable Worlds Using Continuous Representation Visualization

Journal
Journal of Learning Analytics, 6(1)

ISSN
1929-7750

Authors
Pardos, Zachary A
Horodyskyj, Lev

Publication Date
2019

DOI
10.18608/jla.2019.61.1

Peer reviewed

Analysis of Student Behaviour in *Habitable Worlds* Using Continuous Representation Visualization

Zachary A. Pardos
University of California at Berkeley
zp@berkeley.edu

Lev Horodyskyj
Arizona State University

We introduce a novel approach to visualizing temporal clickstream behaviour in the context of a degree-satisfying online course, *Habitable Worlds,* offered through Arizona State University. The current practice for visualizing behaviour within a digital learning environment has been to utilize state space graphs and other plots of descriptive statistics on resource transitions. While these forms can be visually engaging, they rely on conditional frequency tabulations which lack contextual depth and require assumptions about the patterns being sought. Skip-grams and other representation learning techniques position elements into a vector space which can capture a wide scope of regularities in the data. These regularities can then be projected onto a two-dimensional perceptual space using dimensionality reduction techniques designed to retain relationships information encoded in the learned representations. While these visualization techniques have been used before in the broader machine learning community to better understand the makeup of a neural network hidden layer or the relationship between word vectors, we apply them to online behavioral learner data and go a step further; exploring the impact of the parameters of the model on producing tangible, non-trivial observations of behaviour that are illuminating and suggestive of pedagogical improvement to the course designers and instructors. The methodology introduced in this paper led to an improved understanding of passing and non-passing student behavior in the course and is widely applicable to other datasets of clickstream activity where investigators and stakeholders wish to organically surface principal behavioral patterns.

Keywords: behaviour visualization, representations learning, dimensionality reduction, continuous representation visualization, skip-gram, t-SNE, Habitable Worlds, ASU, higher-ed.

INTRODUCTION

A highly-touted benefit of a completely online or otherwise digital course has been that a detailed record of the interactions of learners with course materials is kept that can then be mined for potential actionable insights. It is therefore no surprise that descriptive statistics of the interactions between learners and pedagogical materials were among the first type of information sought by online course instructors. Among their questions of curiosity were: How did students utilize the course resources? How did struggling students differ in their usage patterns compared to passing students? Where did students need help and what can the data provide that can help make improvements to future offerings? While prior work has made headway, these questions largely remain. The current behaviour visualization practice of viewing state space diagrams of student transitions from one resource to another can be unsatisfying as it is ultimately a descriptive statistic summarization of behaviour that can struggle to surface less anticipated patterns of behaviour that require pattern recognition over a longer context window. Classification models of varying complexity can learn such complex behavioural patterns, and while they can at times convey enough corroborating evidence to convince researchers and educators that their predictions are robust, they rarely provide enough novel and interpretable information to affect an instructor's existing understanding of their own course. Representation learning approaches strike a happy medium between simple descriptions and the obscurity of more complex statistical models. As opposed to support vector machines and the cadre of neural network approaches labeled as "deep learning," representation learning algorithms, like skip-grams, are in the class of simple linear feed-forward neural network models. While they are trained by optimizing a predictive outcome, it is the structure of the data found in the model's learned parameters (the representation) that is the artifact of value. In this paper, we show that when used in combination with dimensionality reduction techniques, robust patterns in the data can be surfaced with an order of nuance not attainable with existing descriptive approaches.

BACKGROUND

The practice of training a machine-learned classifier in predictive learning analytics has involved the customary process of feature engineering, a step of transforming data from its original form into a set of more abstract, hand-crafted features benefitted from the domain knowledge of the researcher. This is often a process of aggregation, normalization, or combining of multiple attributes from the original dataset. The motivation for this process is often two-fold: (i) to bring the prior knowledge of the researcher to bear on the problem and (ii) to transform the original data into a form that is syntactically compatible with the chosen classifier(s). This standard practice can be seen in learning analytics work predicting student affect (Baker et al., 2012), drop-out (Boyer, Veeramachaneni, 2015), and question correctness (Stamper & Pardos, 2016), among many other applications of data mining in education (Koedinger et al., 2015). The premise of this approach is that by funnelling the data through the prism of the researcher's domain knowledge, the engineered set of features will be a better representation for the prediction task than the original untreated data. Better, in this case, is defined by superior model fit or generalization and guided by the intuition that there are certain hand-engineered transformations of the original attributes that closely relate to what's being predicted but would be difficult for a classifier to learn from the original data in the process of training. This approach has been quite effective in terms of producing classifiers of reasonable accuracy. When working with smaller datasets, the feature-engineering process can be seen as a type of non-linear human specified functional transformation between data and an intermediate representation which is closer to the target of prediction than the original data. In scenarios with smaller data, bringing the researcher's generalizations to bear to create these representations often leads to more robust models than attempting to statistically learn them.

Visualization, too, can be seen as a manual feature engineering process whereby the researcher brings her prior domain knowledge and hypotheses to bear on the transformation of the data from its original form. The difference between this process in the context of visualization and machine classification is that, in visualization, this feature representation is being presented to a human learner to understand and reason about instead of a machine learner. In the area of visualizing behavior in an online course environment, which is the case study in this paper, common engineered features have been descriptive statistics on MOOC certification (Breslow et al., 2013), dwell time by resource category (Seaton et al., 2013), and counts of click-stream actions in quantized windows over time combined with summaries of forum activity (Crossley et al., 2016). Other work has taken an approach of visualizing behavior using conditional probabilities (e.g. Markov models), which express the most common transitions from one resource to another (Köck & Paramythis, 2011; Caprotti, 2017) or a subsequence of commonly occurring transitions called motifs (Davis, Chen, Hauff & Houben, 2016). In the case of the descriptive approach, a bar chart or scatter plot is used with the X-axis typically being a categorical (types of resources) or timescale (week 1 through 10) and the Y-axis, the attribute being summed (e.g. dwell time, # of actions). In Xu et al. (2014), for example, the Y-axis of their three scatter plots was the learners' course grade, with the X-axis being quiz clicks, lecture page views, and discussion page views. An overview of the past art on visualizing behaviors within Massive Open Online courses can be found in Emmons, Light, & Börner (in-press). The transition frequency approach can be seen as an attempt to disaggregate the descriptive approach and study the relationships between individual resources or resource types. In this approach, a graph is visualized with nodes representing resources (or resource types) and edges representing transitions. The frequency of the transition can be expressed with the thickness or length of the edge. In the case of a report on the first four years of MITx MOOCs (Chuang & Ho, 2016), a prominent figure had each node representing a course and the size of the node depicting the number of enrolments. In graphs, the angular orientation between two vertices with respect to a frame of reference is irrelevant. Algorithms like graph-viz (Ellson et al., 2001) use variants on force-directed algorithms that optimize angles and orientations for visual appeal but do not necessary encode any additional information. The only constraints are the set of edges and vertices and, optionally, the length of those vertices. By visualizing machine-learned representations instead of hand-specified features, we hypothesize that patterns of greater novelty and significance can be revealed. Work has begun to bridge visualization with modeling, using regression with pre-hypothesized features and outcomes of the environment (Fratamico, Perez & Roll, 2017; Park et al., 2017). Bergner, Shu & von Davier (2014) discuss the difficulties of visualization with variable length sequences and the inadequacies of current visualization and clustering methods. The representation approach introduced in this paper provides a method to address these issues.

The practice of feature engineering is often an informationally lossy one. For one, because much of feature engineering is based on aggregation and summarization, and because features created by hand are limited to the scope of the researcher's intuition and domain knowledge. When an ample degree of domain knowledge is present, feature engineering can be an effective way of explicitly defining relevant relationships in the data that a statistical learning approach might struggle to identify. For domain areas in which there is little theory or expertise, such as many behavioural contexts, feature engineering may not be a viable option. The general paradigm of using connectionist models (neural networks) to generate these features is called representation learning. There has been no field where the lossyness of feature engineering has been made more apparent than in computer vision. The dominant intuition for the task of classifying what object is featured in an image was to create a set of features which described the edges present in an image and to then present this set of edge descriptions to a classifier for training. The accuracy results of this approach were eclipsed by Convolutional Neural Networks (Krizhevsky, Sutskever, & Hinton, 2012), which learn representations of the image from the original pixel data. We hypothesize that by using representation learning applied to student behavioural data in an online course, as it has been applied to problem interaction sequences in a math tutoring system (Pardos & Dadu, 2017) and to course enrolment data (Pardos & Nam, 2017), the most significant features of behaviour can be revealed where they may have never been detected if first filtered through the prism of one's domain knowledge in the form of hand engineered features expected to correlate with hypothesized phenomena. The visualization of the learned representation in our approach can also be seen as a visualization of the patterns, or regularities, which are learned in the model.

METHODOLOGY

This paper introduces a novel technical methodology involving representation learning and visualization as well as a qualitative methodology in which the visualization is tuned and interpreted in close collaboration with the course instructor.

The technical methodology began with a dataset containing the chronological sequences of interactions of students with an online for-credit course offered by Arizona State University. A representation learning model common in computational linguistics, called a skip-gram (Mikolov, Yih, & Zweig, 2013), was applied to this behavioural sequence. This is a novel application of the algorithm as it is customarily applied to sequences of words, not behaviours. We hypothesize that the analogy to language representation will hold. This process learns a high dimensionality vector representation of every element interacted with by the students. This vector was reduced to two dimensions for visualization using a non-linear dimensionality reduction technique called t-Stochastic Neighbourhood Embedding (t-SNE), designed to visualize hidden layers in a neural network while retaining significant structure (Maaten & Hinton, 2008).

The tuning and interpretation methodology involved close communication with a subject matter expert, the co-creator and instructor of the course. Since different hyperparameter values of the representation learning method can produce dramatically different vectors, and consequently dramatically different reduced dimensionality visualizations, different values for vector size and window size were used to produce 21 different visualizations of course behaviour that the course instructor was asked to then rate for the amount of novel information they contained. Additional guidance was given that favourable visualizations might be ones that both contained behavioural patterns the instructor was, a priori, confident existed but also contained patterns that were not anticipated but were plausible. The purpose of the visualization was to surface information not already known by the expert. The rationale was that if the visualization depicted behavioural patterns known to be true by the expert, other aspects of the visualization may also be true but not yet known. After identifying useful parameters for the base models, visualizations of student discussion posts and differences between passing and failing students were designed in concert with the instructor. An interactive d3-based visualization was developed simultaneously with this research to enable the subject matter expert to inspect the visualization by hovering the mouse cursor over plot points to reveal semantic meta-information about the element, such as the name of the question and which lesson it belonged to. A coloring feature was also made available whereby the data points could be coloured by lesson or any other categorical feature of the element. The features of the in-house d3 visualization were very similar to features offered by a commercial software package called Tableau but had the benefit of being easily linked to by URL in shared web documents and being made open source[1] to the community by this research effort.

[1] https://github.com/CAHLR/d3-scatterplot

Representation Learning with Skip-Grams

When applied to natural language, a skip-gram model will learn a vector representation of a word based on the many contexts it appears in across a large corpus of text. Simply, the prediction objective of the model is that given an input word, predict the probability of every individual word in the vocabulary, calculating error based on the probabilities of words appearing in context. The vector representation of the word is the output of the single hidden layer network that comprises the model. When two words share similar contexts, they will likely have similar learned vector representations. Often, synonyms of words will have vectors located close to one another in this vector space, also referred to as an embedding. The novel intuition of our application of this to student sequences is that, instead of learning the structure of language by training on sequences of words, we are learning the structure of learner behavior from sequences of problem attempts. It was previously found that clickstream behaviors within MOOCs could be predicted using an RNN with 70% accuracy, compared to the 45% accuracy provided by the expected path through the course when following the existing course structure (Tang, Peterson, & Pardos, 2017). This work builds on that observation that signal exists in learner clickstream behaviors, but instead of focusing on prediction, we seek to scrutinize the behavior to qualitatively learn what these patterns are and their relevance to the pedagogical design of the course. While RNNs are useful for prediction, skip-gram models are better suited for interpretation since they are linear models which create representations embedded in a vector space, subject to arithmetic and scalar manipulation.

Formally, a skip-gram is a simple feed-forward, three-layer neural network with one input layer, one hidden layer, and one output layer (Figure 1). The input, in our context, is a one-hot representation of the course element and the output can be described as a multi-hot representation of the specified number of elements in context. The number of elements in context is two times the window size, which is a hyperparameter of the model. The objective of the model is to predict the elements in context given the input element. Since multiple elements are being predicted, the loss function (categorical cross-entropy) is calculated for each element in context (comprising the multi-hots). The number of weights trained in the model does not increase with the window size since the same weights are used to make predictions of every element in context. The second major hyper parameter is the size of the hidden layer, which ultimately is equivalent to the number of dimensions of the resultant element vector since it is the weights associated with the edges stemming from the one-hot position of the element to all the nodes in the hidden layer. In the case of sequences of words (natural language), the inputs are words in a vocabulary, processed by sweeping sequentially through words in a large corpus, where the model objective is to predict words in context given an input word. In our online course interactions context, the inputs are elements a student interacts with and the model sweeps across chronological sequences of these elements. In our particular dataset of a course created in the Smart Sparrow tutoring system, the logged actions are students’ interactions with screens (or pages) in the course's lessons containing simulations, practice problems, and graded assessments. Specifically, the inputs are screen IDs and the output being predicted are the screen IDs before and after the current screen ID. No correctness information is used in this model, as it’s the student’s navigational behaviour that is the focus, not their performance.

In a skip-gram, the vector representation of an input screen is defined as:

$$v_{w_I} = W^T \delta(w_I) \tag{1}$$

Where W^T is the left side weight matrix in Figure 1, indexed by the one-hot of the input screen, $\delta(w_I)$. A softmax layer, typical in classification tasks, is used to produce a probability distribution over screens to predict screens in context:

$$p(w_O|w_I) = \frac{\exp(W'\delta(w_o)v_{w_I})}{\sum_{j=1}^{V}\exp(W'\delta(w_j)v_{w_I})} \tag{2}$$

For a given output screen, w_O, in the vocabulary, its probability is the exponential normalization defined by the exponentiation of the input screen's vector, v_{w_I}, multiplied by the output screen's vector, $W'\delta(w_O)$, divided by the sum of all screens' exponentiation of their output vector multiplied by the input vector. An output vector is the multiplication of the right side weight matrix, W', with a one-hot of the output screen, w_O.

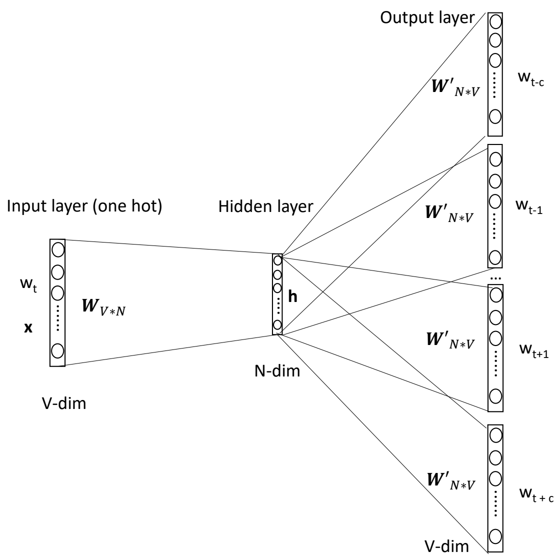

Figure 1. The skip-gram model architecture

The cross-entropy loss (log loss) across all students' sequence of screens, which is backpropogated, is:

$$\mathrm{C} = -\sum_{s \in S}\frac{1}{T}\sum_{t=1}^{T}\sum_{-c \leq i \leq c, i \neq 0}\log p(w_{t+j}|w_t) \tag{3}$$

Where, for each student, s, the average loss is calculated over the input screens at each time slice, t. The loss for a single time slice of a student is the sum of the log of the model's probability of observing the screens within a time slice window size, c, of the current time slice. While the model is trained to minimize error in predicting the screens in context, the intended extract from the model after training is not its predictions of screens but rather the learned representations of the screens in the form of the weight vectors associated with each screen. These weight vectors, a product of the single hidden layer of the model, are the automatic featurization of the screen. Screens which have similar contexts become mapped to vectors of similar magnitude and direction in order to minimize the loss.

CASE STUDY – *HABITABLE WORLDS*

Habitable Worlds is an introductory level 4-credit online lab science course developed by Dr. Ariel Anbar and Dr. Lev Horodyskyj. The course was developed from 2011-2015 in Smart Sparrow's Adaptive eLearning Platform (AeLP), a Powerpoint-esque development environment that gives instructors full control over content, layout, adaptivity, and learning pathways. The platform collects snapshot data while students interact with the educational content, and the data is reported to the instructor both in aggregate and as downloadable CSV sheets should the instructor desire to conduct more detailed analyses on a particular question or activity. Because of these capabilities, continual refinement of *Habitable Worlds* through data analyses, student reactions, and intuition has been made possible. A full account of the pedagogy and design philosophy of the course can be found in Horodyskyj *et al.* (in-press). The course reached a stable version by the fall semester of 2015 with minimal subsequent changes. Despite improvements in course content through changes based on descriptive statistics like time-on-question, number of attempts, and the frequency of certain incorrect answers, general behavioural insights were difficult to come by due to the sheer amount of data that is available and the lack of algorithms for parsing the data into usable information.

Course content is divided into three types of activities: *training*, *application*, and *project*. Most content in the course is linear in nature, with occasional adaptive pathways to remediate misconceptions or bypass mastered content. *Training* activities are activities that introduce students to new concepts and give them the opportunity to explore and experiment without penalty, usually through the use of surveys, short written text, images, short videos (<8 minutes), simulators, equations, problem sets, and virtual field trips (Figure 2). Points are accumulated as students pass milestones. However, students cannot progress until they complete the required activity correctly. *Applications* are equivalent to quizzes, in that no new content is introduced, students are expected to demonstrate competency with topics studied in the associated training activities, and points are deducted for lack of competency. The *project*, essentially a final exam, is a comprehensive activity requiring students to locate rare habitable worlds in a field of 500 randomly generated stars. Course content is released on a weekly basis during the course's 7.5-week deployment period, and students have a week to complete newly released exercises. Training activities remain open and accessible for full credit for the entire term. Applications close off week by week. The project opens on the second week of the course and remains available the rest of the term (Figure 3).

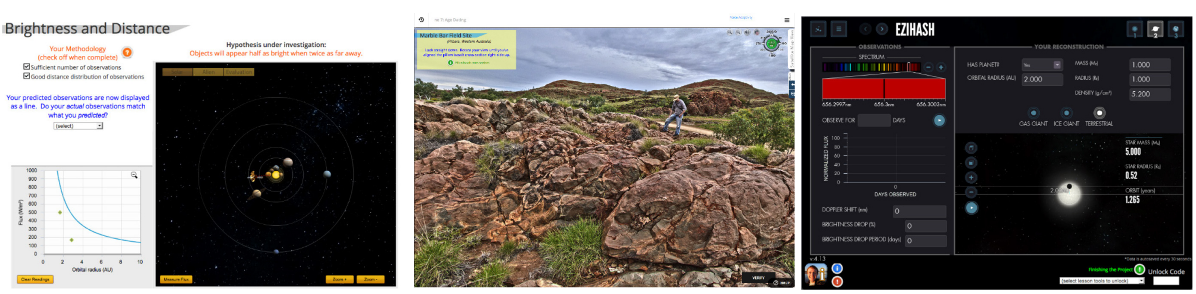

Figure 2: (left) An experimental activity with a simulator involving a hypothesis, a check on the execution of their methodology, and an evaluation of their hypothesis. (center) An observational activity at a virtual field trip. Here students are instructed to rotate their view to observe basaltic rocks in the field in a particular orientation. (right) The course project, where students are required to find rare habitable worlds in a field of 500 randomly generated stars.

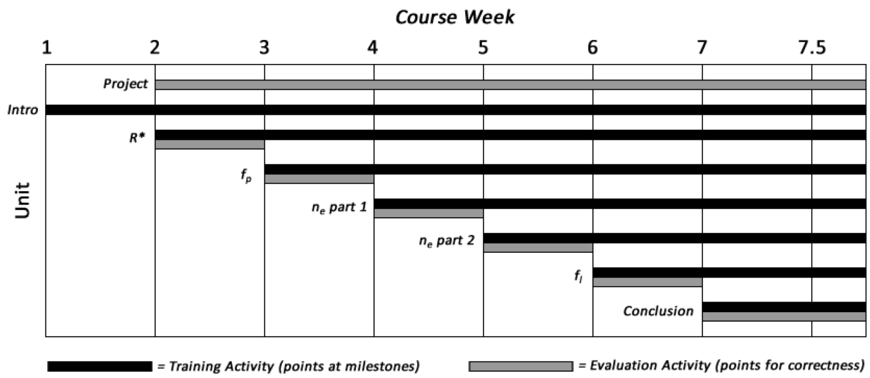

Figure 3: Schedule of activities for *Habitable Worlds*. Training activities open week by week and remain open throughout the term. Applications open for one-week windows. The project opens in week 2 and remains open throughout the term.

The intended completion strategy for *Habitable Worlds* is to alternate between a training activity and its paired application, while accessing the project on a weekly basis as units are completed. However, students are not restricted to the prescribed path. Based on aggregate data generated by the AeLP, student behaviour as observed on a paired discussion board, and general intuition, it was hypothesized that students who failed the course were taking non-optimal approaches, but the nature of those approaches has been opaque to the course instructors to this point.

Data

We used the time-stamped interactions (Table 1) of 778 anonymized students from two offerings of *Habitable Worlds* (Fall 2015 and Spring 2016). A skip-gram model was used to learn continuous representations of course materials from 1.4 million temporal interactions of students with 1,644 pages of the course, called screens, within 134 different lessons. This model had several tuneable hyperparameters that changed the learned representations and thus the resultant visualizations. These were the context window sizes and the number of nodes in the hidden layer (vector size).

Field	Description
userID	Unique identifier for each student
screenID	Unique identifier for each "screen" in a lesson
interactionID	Unique chronologically ordered identifier for an event recorded by the AeLP (typically triggered when students click a "Check" button on the screen)

userID	Example sequence of screenIDs
Sam53	s:1 s:2 s:3 s:2 s:3 s:4 s:5 **s:6** s:7 …
Erica90	s:1 **s:6** s:5 **s:6** s:3 **s:6** s:7 **s:15** s:14 **s:15** s:12 **s:15** s:12 **s:15** s:16 **s:21** …
Paulo1	s:1 s:2 s:3 s:4 s:5 **s:6** s:7 s:8 s:9 s:10 s:11 s:12 s:13 s:14 **s:15** s:7 s:8 s:9 s:10 s:11: s:12 s:13 s:14 **s:15** s:14 **s:15** s:16 …

Table 1: Fields used to construct the dataset (left). Example sequences of screenIDs (right). These sequences, specified one row per student, comprised the dataset used to train the skip-gram model. Application (graded) screen are in bold.

Parameter Tuning & Visualization Evaluation Results

There is no established rule of thumb as to the appropriate hyper-parameters for skip-grams to use for bringing out visually salient patterns in a dataset. As a result, part of our methodology involved finding good values using a limited range of hyper-parameters to create a set of 21 representations and respective t-SNE visualizations which were rated for their usefulness by the co-creator of the course[2] on a five point scale (Figure 4) in similar fashion to Géryk, J. (2015).

[2] Lev Horodyskyj (the 2nd author)

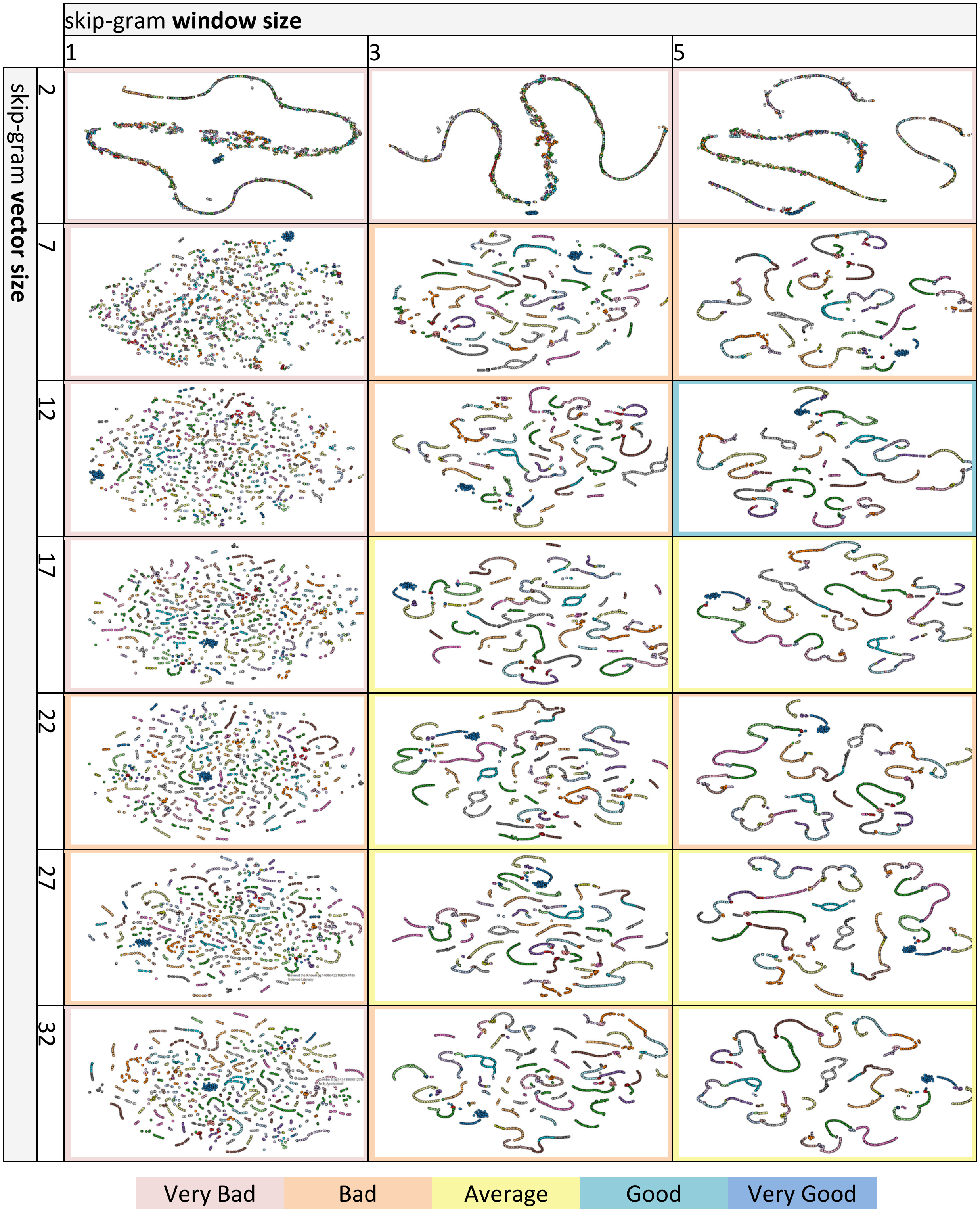

Figure 4: Skip-gram t-SNE visualizations with a variety of window sizes (1, 3 ,5) and vector sizes (2, 7, 12, 17, 22, 27, 32), rated by usefulness (border color). Each of the 21 scatter plots depict the complete set of elements in the course and their relationship to one another.

As mentioned earlier, the guidance given to the expert rater was to favor visualizations that both confirmed existing intuitions and provided additional insight. After initially scanning all visuals to identify common patterns, three confirmatory patterns quickly emerged: (i) clustering of week 2 quizzes (expected, due to observations showing students racing to finish expiring activities before a hard deadline, which first happens in week 2); (ii) clustering of the beginnings of new units onto the endings of old units (expected, based on observations of high-performing students beginning new activities as soon as they were released); and (iii) the splitting of a particular week 5 activity (expected, as the activity is excessively long). A visualization was rated initially (favorably) based on the presence of these three patterns. A visualization would receive extra points if, in addition to possessing the three patterns, the particular plot highlighted novel but explainable patterns.

Analyzing the results from an algorithmic standpoint, the smaller vector size hyper-parameters generally lead to simpler, more singular and linearly connected data points, representing the underlying intended sequencing of the course. There is an analogy here to principal component analysis. When only allowed to represent course elements with two continuous values, these values capture the most predominant structure, which is the courseware sequencing, followed by many students. Large vector sizes can bring out second- and third-order patterns of significance, which will be looked at closer in the next section. When the vector size was two, window sizes of one and three produced more linear, connected plots than a window size of five. With a smaller window with which to learn representations, patterns consisting of many behaviours cannot be easily considered and the representation takes on a form more common when constructing transition plots based only on the frequency of transition from one element to the next (an effective context window of 1). At higher vector sizes, a higher context window size struck a balance between providing a paucity of sequence patterns and retaining full, identifiable and anticipated patterns of behavior.

Results

In summary, the highest rated visualization was produced by a skip-gram with a window size of five and a vector size of 12. Other high vector sizes (paired with high window sizes) also yielded visualizations with useful information. Low vector sizes yielded excessively sequential plots, while low window sizes yielded plots with almost no connections between screens.

With a model of satisfactory representational quality in-hand, the following sections further interrogate this informational artifact to look into more pointed questions regarding student behaviour and its connection to the course pedagogy. Each iteration of the analysis adds an extra layer of complexity to the analysis in order to answer research questions of increasing specificity and utility.

Iteration 1 – Inspecting the Highest Rated Visualization

The annotated visualization (most highly rated from Figure 4) is show in Figure 5, with the prescribed course sequence marked with black arrows and the beginnings of new units starred. Essentially, the plot shows that students tend to approach a course that was designed to be linear in a mostly linear fashion. The visualization reveals behaviours that were previously intuited from aggregate data and discussion board observations. Although helpful in visualizing and confirming student behaviours that had been previously assumed, the skip-gram revealed only limited additional information for course improvements (mostly related to which lessons could be split for being too lengthy).

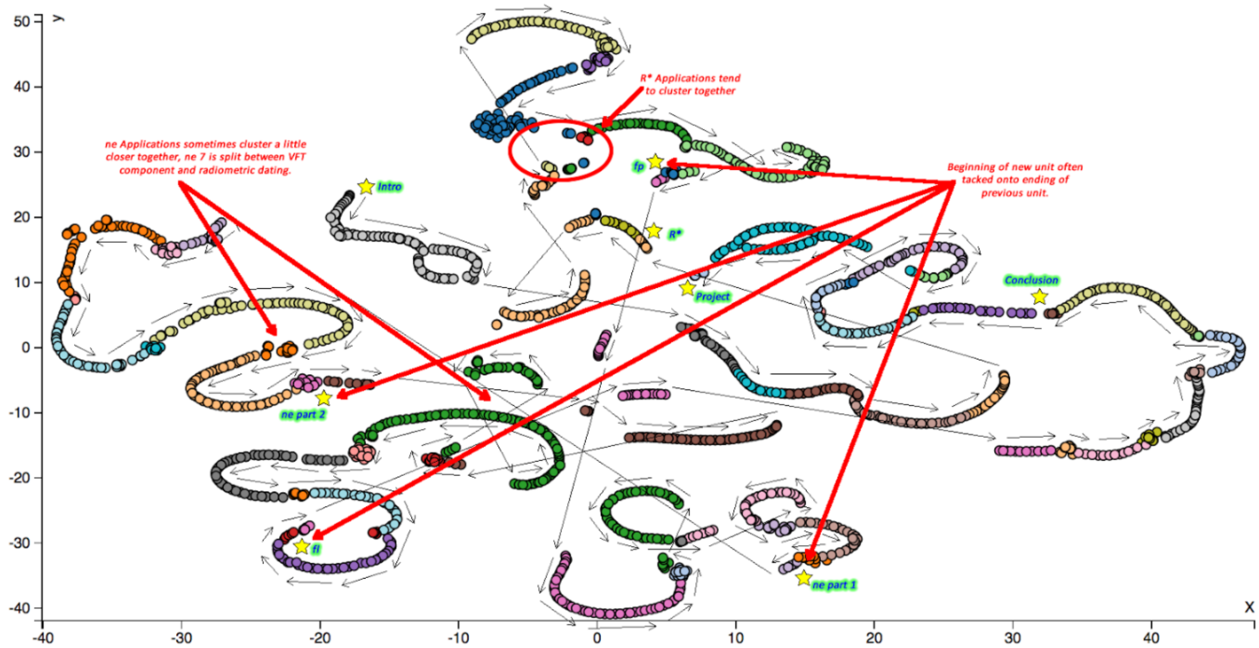

Figure 5: Best visualization (vector length 12, windows size 5) from qualitative rating, with annotations. Gray arrows indicate the sequence of content in the course, stars indicate the beginnings of a new unit, and red notes indicate patterns of interest. The typical pattern consists of a long linear sequence representing the training activity followed by a more clustered application activity. Course material progresses from *Intro* to *R to f_p to n_e *(part 1)* to n_e *(part 2)* to f_l to *Conclusion*. The titles refer to terms of the Drake Equation, a construct used to organize thinking about searching for alien life.**

Iteration 2 – Differentiating Behaviours Exhibited by Passing and Failing Students

Failure in courses is often attributed to student difficulties with content or disengagement. Instructors often have little insight into why a student has failed a course, aside from either observing that a student has failed to attend class or performed poorly on assigned tasks. Previous informal analyses of *Habitable Worlds* offerings showed differences in engagement between passing, failing, and withdrawing students (in addition to level of content mastery and "attendance" based on whether course content was accessed). Passing students fully engaged with a majority of the content, while students who withdrew engaged with little, if any content. Failing students, surprisingly, showed persistence in the course, often engaging with content week after week, despite their inability to complete the content successfully.

The second iteration of the skip-gram featured the addition of grade information to parse behaviours (Figure 6). Although content in *Habitable Worlds* is linear, individual training (10-60 screens), application (~5 screens), and project (~5 screens) lessons can be accessed at random, if desired, during any given week. The optimal course pathway is to complete a training activity followed by its associated application activity, repeated for each topic (between 4 and 6 for each week). Once the unit is complete, the associated component of the project can be completed. Units of the course are time-gated, but lessons within the units, once released, are not gated. Although designed to be completed in sequence, a student can start and work on any lesson for that week that they choose.

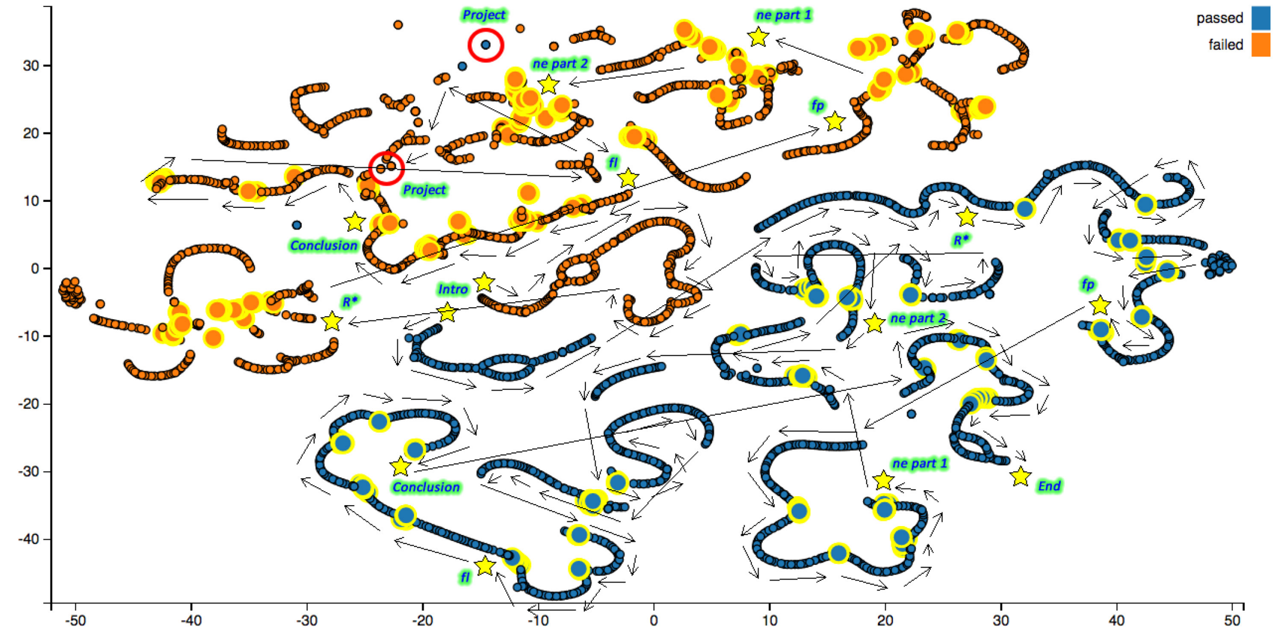

Figure 6: Behaviour of passing (blue) and failing (orange) students. Passing students show a linear progression through content, with applications appearing sequentially after their associated training activities (highlighted in yellow). Failing students show a hub-and-spoke approach for most units, with training activities converging onto a cluster of application activities for that unit. Project screens are highlighted in red. The project clusters far from activities for passing students (indicating random access during the semester), and clusters close to the concluding unit for failing students.

The introductory unit of the course, completed in week 1, does not have any associated applications (points for competency), only training activities (points for completion). Paired applications begin with week 2 content. For both passing and failing students, week 2 (R*) applications cluster together, indicating that students are accessing them closely together in time. This may indicate that students are racing the deadline and attempting to complete the hard-deadline content before that material locks off (training activities do not lock off). Passing students seem to realize that this is not an optimal strategy and switch to a more sequential approach in subsequent units, where they complete applications in tandem with their associated training activities. Failing students, however, make this switch much later in the term. This results in a hub-and-spoke pattern for each unit on the skip-gram for failing students, where applications cluster together, with training activities radiating off of them. This indicates that students may be attempting to complete the applications first, and only proceeding to the training activities when they cannot complete the applications.

In addition, there are differences between passing and failing students in how they approach the project. The course project is released in week 2 and is a fairly complex endeavour, requiring students to utilize skills learned in almost every unit of the course and assembling those skills into a methodology to find a handful of habitable worlds in a field of hundreds of stars. The optimal strategy is to engage with the project early and complete components of the project as the concepts are learned. For passing students, the project clusters quite far away from the rest of the course, indicating that they are accessing the project throughout the course, hence the lack of association with any particular week's activities. For failing students, however, the project clusters very close to the Conclusion unit, indicating that failing students do not engage with the project until all other course material has been completed.

Overall, this visualization, where students are differentiated based on their grade, revealed that although both groups of students take the non-optimal strategy early in the course, passing students subsequently adopt the optimal strategy while failing students do not, continuing to struggle week by week.

Iteration 3 – Discussion Forums

Habitable Worlds is paired with a discussion help forum in the Piazza platform (www.piazza.com). Students who are stuck or require a more detailed explanation on any course concept can post their questions to the forum, where either an instructor or a fellow student replies. Average response time is 5-10 minutes during most of the day. Instructors typically offer assistance for training activities only, while fellow students are allowed to offer assistance on training, applications, and the project. Because most of the evaluations are generated randomly using equations and algorithms in the AeLP, students can exchange techniques for solving problems, but not answers as no two students have the same problem sets. This results in a collaborative environment in which students and instructors work together to learn and master skills and concepts, mimicking a real scientific environment.

Piazza posts (but not views) are time-stamped, and so they were added to the analysis to understand how passing and failing students are utilizing the discussion board (Figure 7). With very few exceptions, passing students' forum postings do not cluster with any particular lesson. This indicates that passing students are posting on the discussion board only when they need help, and this differs from student to student, resulting in clustering of posts away from any particular content (with exceptions for a couple of applications that are known to be difficult). Failing students' forum postings tend towards this pattern as well, but there are a larger number of posts that cluster close to their associated applications. This indicates that when running into problems, failing students reach out for assistance. However, because this tends to happen during the applications activities, this indicates that these students have not mastered the content in the training activities, and so when asked to apply their knowledge, they struggle and reach out for help.

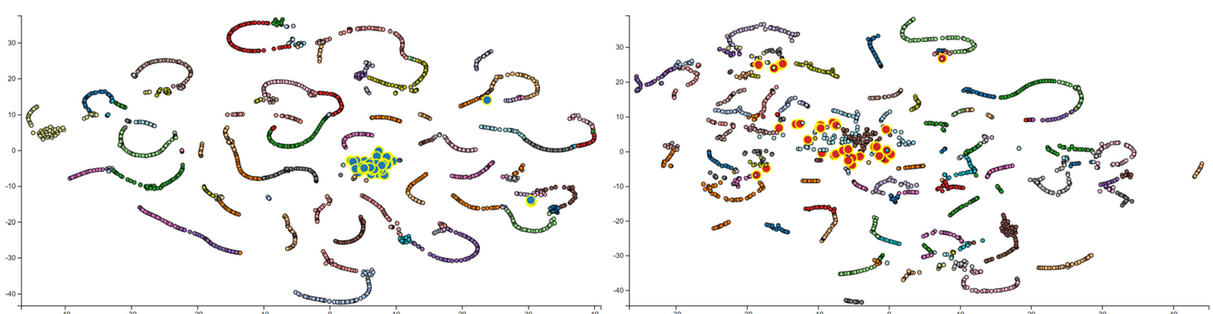

Figure 7. (left) Passing student forum behaviour (spring 2016), showing little association between lessons and the discussion board. (right) Failing student forum behavior (spring 2016), showing a closer association between application activities and discussion board posting.

The reason for the lack of content mastery is unclear. Piazza records the number of posts and threads "read" by students, but does not timestamp this particular activity. Many students have self-reported that they do not post on the board often because they start activities late and find that when they become stuck, their questions have already been answered on the forum days earlier. Training activities are necessarily limited because they are teaching conceptual frameworks, and these frameworks often have singular explanations (i.e., there are only so many ways to teach "higher temperatures = more evaporation"). If a student is using the discussion board as a crutch to complete difficult training content by doing what previous students had reported as resulting in success (without understanding *why* it led to success), it is likely that they are not internalizing the concepts of the course. Hence, when they reach the application, which creates randomized activities based on the underlying concepts, it is likely that they struggle and reach out for help because they never gained an understanding of the underlying concept that is being tested in the application.

Habitable Worlds Case Study Conclusions

The skip-grams expanded on a variety of intuitions resulting from interacting with students on the discussion boards, via e-mail, and in person over the years that *Habitable Worlds* has run. On a cursory level, the skip-grams have helped to identify changes that could benefit the class, such as splitting lengthy content.

More significantly, the skip-grams have revealed significant behavioural differences between passing and failing students. This information is critical to the next phase of development for *Habitable Worlds*. Additional trap-states and remediation pathways are helpful in providing students with additional activities for concepts with which they are having difficulty. But these kinds of remediation, while helpful in refining existing content, do not assist in more fundamental problems in learning behaviours.

Habitable Worlds was designed to be approached in whatever approach a student finds most comfortable. A student can pace their work across an entire week or complete it all in a short burst. Students can work alone or together, either online or in-person. The course project can be completed during the course of the entire semester or all the way at the end. This design was intentional to allow maximum flexibility for students, many of whom are non-traditional and have significantly more responsibilities than a typical on-campus student.

However, for many failing students, this design may be detrimental as it depends on a certain amount of self-discipline and awareness of personal limitations. The representation visualization confirmed an existing assumption about failing students' unsophisticated strategy towards engaging with the course project, but also revealed that this strategy was applied not just to the project, extending to the rest of the course content and likely the usage of the discussion board. Students who fall into this category may require a more structured approach in order to successfully complete the course. Future versions of the course will focus on building better supports, such as an early module on successful course strategies, and course structure enhancements, such as adding flexible deadlines and enforced content ordering for students who need it. These planned modifications are expected to better support those who are not just struggling with the subject matter of the course but with learning to learn in an online environment.

CONCLUSIONS

We have developed a methodology around the nascent field of representation learning and interpretation and applied it to the domain general topic of understanding the patterns of temporal behaviors of learners. Philosophically, this approach makes as few pre-conceptions about behaviour as possible, instead allowing prominent features of behavior to surface organically with the aid of careful model and visualization tuning that takes place between the researcher and the practitioner closest to the domain. The course element vectors produced from the aggregate behaviors of students are themselves the collection of edge weights from a neural network model. This notion of information being contained within the edges of a network, distributed across a space, as opposed to single nodes, or pre-determined constructs, is harkened to in theories of Connectivism (Siemens, 2014). We believe that the novel visualization and representation methodologies introduced in this paper can usher in a new brand of measurement in digital learning environments that aligns with what have been the core values of the Learning Analytics community.

FUTURE WORK

An obstacle to adoption of this method by a broader audience of instructors, without researcher support, is the rather careful process by which the parameters of the visualization were initially tuned. Since this was a collaborative research endeavour, guided by the educational theory informed goals of the instructor (Hillaire, Rappolt-Schlichtmann, & Ducharme, 2016), it was justified for this laborious effort to be undertaken by the instructional staff; however, this should not be a pre-requisite, in practice, for a typical instructor to take on before interfacing with these analytics. The technique needs to be applied to more courses and more settings that vary by the type of students, course material, and platform to ascertain if common useful settings emerge in general contexts which would reduce or eliminate the upfront manual tuning effort. It is also an open question whether a threshold level of analytical data fluency or light professional development is required in order for a domain expert to independently begin to make sense of these visualizations.

ACKNOWLEDGEMENTS

This work was supported by the Bill & Melinda Gates Foundation.

REFERENCES

Baker, R. S., Gowda, S., Wixon, M., Kalka, J., Wagner, A., Salvi, A., ... & Rossi, L. (2012). Sensor-free automated detection of affect in a Cognitive Tutor for Algebra. In Educational Data Mining 2012. Pp. 276-281.

Bergner, Y., Shu, Z., & von Davier, A. (2014). Visualization and Confirmatory Clustering of Sequence Data from a Simulation-Based Assessment Task. *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*, 177-184.

Boyer, S., & Veeramachaneni, K. (2015). Transfer Learning for Predictive Models in Massive Open Online Courses. *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, 54-63.

Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., & Seaton, D. T. (2013). Studying Learning in the Worldwide Classroom: Research into edX's First MOOC. *Research & Practice in Assessment*, *8*, 13-25.

Caprotti, O. (2017). Shapes of educational data in an online calculus course. Journal of Learning Analytics, 4(2), 76–90.

Chuang, I., & Ho, A. D. (2016). HarvardX and MITx: Four Years of Open Online Courses (Fall 2012-Summer 2016). Available at SSRN: http://dx.doi.org/10.2139/ssrn.2889436.

Crossley, S., Paquette, L., Dascalu, M., McNamara, D. S., & Baker, R. S. (2016). Combining Click-Stream Data with NLP Tools to Better Understand MOOC Completion. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, 6-14.

Davis, D., Chen, G., Hauff, C., & Houben, G. J. (2016). Gauging MOOC Learners' Adherence to the Designed Learning Path. *Proceedings of the 9th International Conference on Educational Data Mining (EDM)*, 54-61.

Ellson, J., Gansner, E., Koutsofios, L., North, S. C., & Woodhull, G. (2001). Graphviz — Open Source Graph Drawing Tools. *International Symposium on Graph Drawing*, 483-484.

Emmons, S., Light, R., & Börner, K. (in-press) MOOC Visual Analytics: Empowering Teachers, Students, Researchers, and Developers of Massively Open Online Courses. *Journal of the Association for Information Science and Technology*. arXiv:1702 (01090v1).

Fratamico, L., Perez, S., & Roll, I. (2017). A Visual Approach towards Knowledge Engineering and Understanding How Students Learn in Complex Environments. *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, 13-22.

Siemens, G. (2014). Connectivism: A learning theory for the digital age.

Géryk, J. (2015). Using Visual Analytics Tool for Improving Data Comprehension. *Proceedings of the 8th International Conference on Educational Data Mining (EDM)*, 327-334.

Hillaire, G., Rappolt-Schlichtmann, G., & Ducharme, K. (2016). Prototyping Visual Learning Analytics Guided by an Educational Theory Informed Goal. *Journal of Learning Analytics, 3*(3), 115-142.

Horodyskyj, L. B., Mead, C., Belinson, Z., Buxner, S., Semken, S., & Anbar, A. D., (in-press). *Habitable Worlds*: Delivering on the Promises of Online Education. *Astrobiology, 17*(11).

Köck, M., & Paramythis, A. (2011). Activity Sequence Modelling and Dynamic Clustering for Personalized E-Learning. *User Modeling and User-Adapted Interaction*, *21*(1), 51-97.

Koedinger, K. R., D'Mello, S., McLaughlin, E. A., Pardos, Z. A., & Rosé, C. P. (2015). Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(4), 333-353.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems, 25*, 1097-1105.

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579-2605.

Mikolov, T., Yih, W. T., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL* (Vol. 13, pp. 746-751).

Pardos, Z.A., Dadu, A. (2017) Imputing KCs with Representations of Problem Content and Context. In *Proceedings of the 25th Conferece on User Modeling, Adaptation and Personalization* (UMAP'17). Bratislava, Slovakia. ACM. Pages 148-155.

Pardos, Z.A. & Nam, A. (2017) The School of Information and its Relationship to Computer Science at UC Berkeley. In *Proceedings of the 2017 iConference*. Wuhan, China.

Park, J., Denaro, K., Rodriguez, F., Smyth, P., & , M. (2017). . In Proceeding of *Learning Analytics and Knowledge* (pp. 21-30).

Tang, S., Peterson, J., and Pardos, Z. (2017). Predictive Modelling of Student Behaviour Using Granular Large-Scale Action Data. In Lang, C., Siemens, G., Wise, A. F., and Gaevic, D., editors, *The Handbook of Learning Analytics*, pages 223–233. Society for Learning Analytics Research (SoLAR), Alberta, Canada, 1st edition.

Seaton, D. T., Bergner, Y., Chuang, I., Mitros, P., & Pritchard, D. E. (2013). Who does what in a massive open online course? *Communications of the ACM, 57*(4), 58-65.

Stamper, J., & Pardos, Z. A. (2016). The 2010 KDD Cup Competition Dataset: Engaging the machine learning community in predictive learning analytics. *Journal of Learning Analytics*, *3*(2), 312-316.

Xu, Z., Goldwasser, D., Bederson, B. B., & Lin, J. (2014). Visual Analytics of MOOCs at Maryland. *Proceedings of the First ACM Conference on Learning @ Scale*, 195-196.