Learner Behavioral Feature Refinement and Augmentation using GANs

Da Cao[1], Andrew S. Lan[2], Weiyu Chen[1],
Christopher G. Brinton[1 2], and Mung Chiang[3]

[1] Zoomi, Inc.
[2] Princeton University
[3] Purdue University

Abstract. Learner behavioral data (e.g., clickstream activity logs) collected by online education platforms contains rich information about learners and content, but is often excessive and difficult to interpret. In this paper, we study the problem of learning low-dimensional, interpretable features from this type of raw, high-dimensional behavioral data. Based on the premise of generative adversarial networks (GANs), our method refines a small set of human-crafted features while also generating a set of additional, complementary features that better summarize the raw data. Existing methods, by contrast, either define heuristic features that cannot capture all nuances in the raw data or generate uninterpretable features. Through experimental validation on a real-world dataset that we collected from a corporate training setting, we demonstrate that our method leads to features with high predictability and interpretability. In particular, they obtain log-likelihoods of up to -0.917 in predicting learner quiz scores, and not only can fool the discriminator against the understandable human-crafted features but are also strongly associated with raw learner behaviors.

1 Introduction

Online learning platforms, such massive open online courses (MOOCs), have the capability of collecting large-scale learner behavioral data at low costs. Examples of such data include content usage patterns [10, 12], social interactions [6, 29], and keystroke/clickstream events [1, 7]. The existence of behavioral data has motivated research on identifying non-assessment-related factors that contribute to learner performance, e.g., engagement [15,30], confusion [32], and emotion [8].

These factors in turn have the potential of providing effective learning and content analytics to instructors, with research having shown that learner behavior is highly predictive of learning outcomes. [21], for example, found that working on more assignments in MOOCs improves knowledge transfer, while [10, 16] found that learner activity patterns are predictive of certification status and early dropout, respectively. Further, [1, 25, 31] found that learner discussion forum, assignment submission, and keystroke pattern activities are predictive of test performance, exam scores, and essay quality.

Despite its predictive power, behavioral data is itself often massive and difficult to interpret; [9, 23], for example, showed that a single learner can generate thousands of clickstream events even in short courses. When extracted carefully, however, even small sets of features have been seen to sufficiently characterize learner behavior in a manner that is predictive of learning outcomes; [7, 33] showed that a set of nine behavioral features computed from clickstream events could obtain above 70% quality in predicting learner performance on particular quizzes, and an average improvement of 60% over baselines in predicting learner grades, with additional features not yielding significant improvements. Such findings in turn motivate work in developing systematic methods for identifying low-dimensional representations of learner behavior.

Existing approaches to finding such representations can be divided into two categories: model-driven feature extraction and human-crafted features, each of which suffers from some drawbacks. Model-driven features will by definition capture even the most subtle nuisances in the raw learner data, and thus lose the least amount of variance in the process; examples of such approaches applied to educational data include principal component analysis (PCA) [4], matrix factorization [5], and variational autoencoders [20]. The features resulting from these approaches, however, exhibit little to no interpretability, and do not offer strong learning and content analytics for instructors. Human-crafted features, on the other hand, are based on human knowledge of education and are highly interpretable as a result [2, 7, 19, 23]. Compared with model-driven approaches, though, these features often have significantly lower predictability [24].

There are a few model-driven approaches that have sought to achieve both high predictability and high interpretability [3, 11, 22], but these have focused exclusively on learner quiz response data. Our goal in this paper, as a result, is to develop the first model-driven approach for analyzing behavioral data that yields features with both of these qualities.

1.1 Our Method and Contributions

In this work, we address the challenge of learning interpretable, low-dimensional representations of high-dimensional learner behavioral data. Our method, detailed in Section 2, does this using a small number of interpretable, hand-craft features derived from the data. First, it uses generative adversarial networks (GANs) [14] to learn a set of refined features that closely correspond to the hand-crafted features, yet better summarize the original raw, high-dimensional learner data. Second, it learns a set of additional, complementary features that capture the nuisances in the data that are not explained by the refined features. Our resulting GAN approach consists of three components: a *generator* that turns raw learner data into refined and complementary features, a *discriminator* that tries to distinguish between hand-crafted features and generated features, and a *reconstructor* that ensures the refined and additional features can reconstruct the original, raw data with high fidelity.

In Section 3, we experimentally validate our proposed method using a dataset we collected from a short online course hosted on Zoomi's learning platform.[4] In doing so, we demonstrate that our refined and complementary features both (i) outperform other behavioral data featurization methods in predicting learner quiz performance, obtaining log-likelihoods of up to -0.917, and (ii) exhibit strong interpretability, with the ability to fool the discriminator against human-crafted features and a strong association with raw learner behaviors. These results emphasize that our model-driven feature generation process preserves human interpretability.

2 Feature Generation Method

In this section, we detail our approach for feature refinement and augmentation using GANs. We first formalize the problem (Section 2.1), then present the GAN models (Section 2.2-2.4), and finally derive our parameter optimization procedure (Section 2.5).

2.1 Problem formulation

Let U denote the number of learners, indexed by $u \in \{1, 2, \ldots, U\}$, and let D denote the number of raw data features, indexed by $d \in \{1, 2, \ldots, D\}$. We represent learner u's data as the feature vector $\mathbf{x}_u^r \in \mathbb{R}^D$. In doing so, we assume that there are no missing values in these raw learner feature vectors, meaning that the raw feature value of every learner on every feature is observed.

We also leverage a set of $G \ll D$ given, human-crafted "gold standard" features. Letting $\mathbf{x}_u^g \in \mathbb{R}^G$ denote the vector containing learner u's gold standard feature values, our goal is to produce a set of refined features and an additional set of $A \ll D$ complementary features that satisfy the following conditions:

a) The refined features are similar to the gold standard features, but better resemble the raw data.
b) The additional complementary features, together with the refined features, form a low-dimensional representation of learner behavior that is able to reconstruct the raw data with high fidelity.
c) Both the refined and complementary features are human-interpretable.

We denote learner u's refined and complementary feature vectors as $\mathbf{x}_u^f \in \mathbb{R}^G$ and $\mathbf{x}_u^a \in \mathbb{R}^A$, respectively.

In order to satisfy these three conditions, we make use of the GAN framework. Our GAN model consists of three parts: (i) a **generator** that outputs a vector $\mathbf{x}_u^{gen}$ of each learner's values, consisting of each refined feature $\mathbf{x}_u^f$ and each complementary feature $\mathbf{x}_u^a$, given the learner's raw data features $\mathbf{x}_u^r$ as input; (ii) a **discriminator**, which seeks to classify whether a learner's gold standard feature vector $\mathbf{x}_u^g$ or refined feature $\mathbf{x}_u^f$ vector is human-crafted or produced by

[4] `http://zoomiinc.com/`

the generator, and (iii) a **reconstructor**, which takes each learner's refined and complementary features and attempts to reconstruct the raw data features from them. We formalize these three parts in detail next; note that we will experiment with both shallow and deep neural networks in each case.

2.2 Generator

The purpose of the generator is to transform high-dimensional, raw data vectors into low-dimensional feature vectors containing both the refined features and the complementary features. Formally, the generator is denoted $G(\cdot) : \mathbb{R}^D \to \mathbb{R}^{G+A}$, and

$$\mathbf{x}_u^{gen} = [(\mathbf{x}_u^f)^T, (\mathbf{x}_u^a)^T]^T = G_{\mathcal{W}_G}(\mathbf{x}_u^r),$$

where $\mathcal{W}_G$ denotes all parameters of the generator. We use a one layer fully-connected neural network as the non-linearity G of the generator, given by

$$\mathbf{y} = \tanh(\mathbf{W}\mathbf{x} + \mathbf{b}),$$

so the set of parameters $\mathcal{W}_G$ consists of the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$. In experimenting with deeper neural networks consisting of several hidden layers, we found a tendency of overfitting.

2.3 Discriminator

The purpose of the discriminator is to classify whether a learner's feature values are human-crafted (labeled as $y_u = 1$ for $\mathbf{x}_u^g$) or generated by the generator (labeled as $y_u = 0$ for $\mathbf{x}_u^f$). Formally, the discriminator is denoted $D(\cdot) : \mathbb{R}^G \to [0, 1]$, and

$$p(y_u = 1) = D_{\mathcal{W}_D}(\mathbf{x}_u^f \text{ or } \mathbf{x}_u^g),$$

where $\mathcal{W}_D$ denotes all parameters of the discriminator. Here, we use a deep fully-connected neural network for the nonlinearity:

$$\begin{aligned}
\mathbf{h}_1 &= \tanh(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \\
\mathbf{h}_2 &= \tanh(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2) \\
\mathbf{y} &= \tanh(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3),
\end{aligned}$$

where h_1 and h_2 are hidden layers with 50 hidden units. In this case, then, $\mathcal{W}_D$ consists of $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$.

2.4 Reconstructor

The purpose of the reconstructor is to reconstruct a learner's raw data feature values using their refined gold standard feature values $\mathbf{x}_u^f$ and their additional

complementary feature values $\mathbf{x}_u^a$. Formally, the reconstructor is denoted $R(\cdot) : \mathbb{R}^{G+A} \rightarrow \mathbb{R}^D$, and

$$\widehat{\mathbf{x}}_u^r = R_{\mathcal{W}_R}([(\mathbf{x}_u^f)^T, (\mathbf{x}_u^a)^T]^T),$$

where $\mathcal{W}_R$ denotes all parameters of the reconstructor. Similar to the network defined in Section 2.2, we also use an one layer fully-connected neural network as our reconstructor. Note that since we normalize our features to lie within $[-1, 1]$, tanh is chosen for both the generator nonlinearity (i.e., $G(\cdot)$) and the reconstructor nonlinearity (i.e., $R(\cdot)$) since it is the only one that has an output range $[-1, 1]$.

2.5 Optimizing GAN

We iteratively update the parameters of the generator, discriminator, and reconstructor in round-robin fashion. Similar to the general steps described in [14], in each training iteration, we perform the following three steps:

(1) *Minimize Discriminator Loss*: First, we minimize the discriminator loss over its parameters, i.e.,

$$l_D(\mathcal{W}_D) = -\sum_{u=1}^{N} \log D_{\mathcal{W}_D}(\mathbf{x}_u^g) - \sum_{u=1}^{N} \log(1 - D_{\mathcal{W}_D}(\mathbf{x}_u^f)). \tag{1}$$

In other words, we aim at improving the discriminator's ability to distinguish between human-crafted gold standard features and generated features, as defined in Sec. 2.2. Note that in this step we do not optimize over the parameters of the generator.

(2) *Minimize Reconstructor Loss*: Then, we minimize the l-2 reconstructor loss over its parameters, i.e.,

$$l_R(\mathcal{W}_R) = \sum_{u=1}^{N} \frac{1}{2} \|\mathbf{x}_u^r - \widehat{\mathbf{x}}_u^r\|_2^2 = \sum_{u=1}^{N} \frac{1}{2} \|\mathbf{x}_u^r - R([(\mathbf{x}_u^f)^T, (\mathbf{x}_u^a)^T]^T)\|_2^2. \tag{2}$$

In other words, we aim at improving the reconstructor's ability to reconstruct each learner's raw feature values from the generator's output (their refined gold standard feature values and their additional complementary feature values). In this step, we do not optimize over the parameters of the generator either.

(3) *Minimize Generator Loss*: Finally, we minimize the generator loss over its parameters. In doing so, we seek to (i) minimize the reconstruction loss while (ii) improving the generator's ability to generate refined gold standard features that are similar to human-crafted ones, thus fooling the discriminator. These two are combined in the loss function

$$l_G(\mathcal{W}_G) = \alpha \sum_{u=1}^{N} \frac{1}{2} \|\mathbf{x}_u^r - \widehat{\mathbf{x}}_u^r\|_2^2 - (1 - \alpha) \sum_{u=1}^{N} \log(D_{\mathcal{W}_D}(\mathbf{x}_u^f))$$

$$= \alpha \sum_{u=1}^{N} \frac{1}{2} \|\mathbf{x}_u^r - R(G(\mathbf{x}_u^r))\|_2^2 - (1-\alpha) \sum_{u=1}^{N} \log(D_{\mathcal{W}_D} G(\mathbf{x}_u^r)), \quad (3)$$

where $\alpha \in [0, 1]$ strikes balance between these two objectives. In this step, we do not optimize over the parameters of the discriminator and the reconstructor.

3 Experiments

In this section, we evaluate our feature generation method using real-world data. We start by describing our dataset in Section 3.1 and detailing our algorithm implementation procedures and metrics in Section 3.2. Then, we present and discuss results of the predictability and interpretability aspects of our features: in Sec. 3.3, we compare their ability to predict learner quiz performance against other feature extraction methods, and in Sec. 3.4, we investigate the degree to which their components associate with raw learner behaviors.

3.1 Course Dataset

The datasets we use come from an online corporate training courses that we hosted on Zoomi's platform. The course content is a slideshow presentation consisting of 35 slides; we consider each slide to be a "segment" of content. We consider users the 2,914 learners who enrolled in this course over a seven month period in 2017.

For each learner, we summarize their clickstream data in terms of two sets of features: raw data features $\mathbf{x}_u^r$, and human-crafted gold standard features $\mathbf{x}_u^g$. The raw data features consists of four quantities measured at the individual segment level: time spent, expected time spent, number of views, and engagement score [9]. With 35 different segments, there are 140 total raw features. The set of gold standard features, on the other hand, consists of (i) four features constructed using the same set of measurements as the raw features, but aggregated at the course level, and (ii) two additional hand-crafted features: the number of times a learner switches away from the course platform, and the number of visits to the course. These features are summarized in Table 1.

Feature	Description	Feature Type
Time Spent	the amount of (real) time that a learner spent	Raw, Gold Standard
Views	the number of times that a learner has viewed	Raw, Gold Standard
Engagement	the amount of effort a learner is putting into studying	Raw, Gold Standard
Expected Time Spent	the amount of (real) time that a learner is expected to spend	Raw, Gold Standard
Off-task Behavior	the amount of time that a learner spent with the application idle	Gold Standard
# Course Visits	the number of times the learner visited the course	Gold Standard

Table 1: Definitions of the raw and human-crafted gold standard behavioral features.

3.2 Training procedures and metrics

We now detail our procedures for training and tuning model parameters, as well as the metrics used in our evaluation.

Training GAN. We train the generator and discriminator in round-robin fashion, following [14]. We use the batch approach, where in each epoch, we randomly select 50 users in our dataset to train the parameters in our model. In particular, we employ the Adam Optimizer [18] in minimizing the generator, discriminator, and reconstructor losses. We found that the discriminator usually outperforms the generator if they are trained equally frequently. Therefore, to balance these two, we train the generator multiple times before training the discriminator once, i.e., a gen-to-dis ratio that is larger than 1. We experimented with several values and observed that a gen-to-dis ratio of 2 worked best. The reconstructor network, however, is trained in every epoch.

Parameter tuning. To optimize the performance of our model, we consider two main parameters for tuning. We first tune α in the generator loss in Equation 2, which balances between minimizing the reconstruction loss and maximizing the generator's ability to fool the discriminator. With $\alpha = 0$, the generator weights are optimized solely towards generating features similar to the original human-crafted ones, thus fooling the discriminator. On the contrary, with $\alpha = 1.0$, the generator weights are optimized solely towards minimizing the reconstruction loss. We sweep over the values of α as $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$. The second parameter we consider is A, the number of additional complementary features as defined in Section 2.1. We sweep over the values of A as $A \in \{0, 1, \ldots, 15\}$.

Performance metrics. We quantify the ability of the generator and discriminator to classify hand-crafted and generated features using the standard cross entropy loss metric [13]. Lower loss values imply better performance. For the reconstructor, we quantify its ability on reconstructing raw learner data using the R^2 score as our performance metric. In our context, R^2 is defined as

$$R^2 = 1 - \frac{\sum_{u=1}^{N} \frac{1}{2} \|\mathbf{x}_u^r - \widehat{\mathbf{x}}_u^r\|_2^2}{\sum_{u=1}^{N} \frac{1}{2} \|\mathbf{x}_u^r - 1/N \sum_{u=1}^{N} \mathbf{x}_u^r\|_2^2},$$

with larger values corresponding to better reconstruction quality.

Training progress. Figure 1 plots the training progress of our model, with choices of $A = 8$ and $\alpha = 0.9$. The discriminator and reconstructor curves show loss values in each epoch as calculated in Equations 1 and 2, respectively. As the first item defined in Equation 3 has already been reflected in the reconstructor loss curve, the generator loss curve only plots the second item in the equation, which also corresponds to the cross entropy loss of the generator's success in confusing the discriminator.

The reconstructor loss drops significantly over the first 5,000 epochs, before stabilizing at 0.02, showing that our generated features can reconstruct the raw data with high fidelity. The generator and discriminator losses tend to fluctuate in opposite directions, conforming to the intuition that they are competing

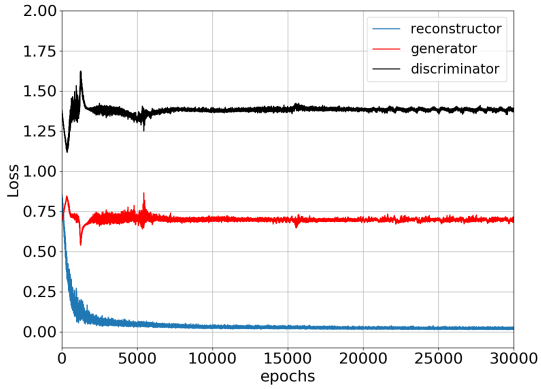


Fig. 1: Visualization of the training progress of our GAN. The reconstruction loss decreases steadily while the generator and the discriminator losses experience some fluctuations before converging.

against each other. Both losses display relatively dramatic fluctuations in the first 3,000 epochs, after which they fluctuate to a much smaller degree, and eventually stabilize after around 5,000 epochs ending up as matching rivals in an equilibrium.

Figure 2 shows the R^2 score of the reconstructor over the number of complementary features, for various α values. When the number of complementary features is small, higher α values tend to perform slightly better. As the number of complementary features increases, the performance improves, and the impact of α diminishes.

3.3 Predicting quiz performance

Overall quiz score is the most direct and effective measure of a learner's outcome available for this dataset. Thus, we evaluate our model by comparing the generated features ($\mathbf{x}_u^{gen}$) to features generated by baseline methods in terms of their ability to predicting learner quiz scores. We consider the following baselines: (i) high-dimensional raw features ($\mathbf{x}_u^r$), (ii) low-dimensional, human-crafted gold standard features ($\mathbf{x}_u^g$), (iii) low-dimensional features constructed by principal component analysis (PCA) on raw data features, denoted as $\mathbf{x}_u^{PCA}$, and (iv) low-dimensional features constructed by training a one-layer autoencoder [17], denoted as $\mathbf{x}_u^{ae}$. Similar to our GAN, we use tanh for the encoder nonlinearity, while letting the decoder have no nonlinearity. In the latter two baselines, we ensure that the number of features is equal to the total number of features in our GAN (*i.e.,* $A + G$), for a fair comparison.

Labels. The overall quiz score for each learner is discrete, taking values in $[0, 0.8, 0.9, 1.0]$. Thus we formulate this as a multi-class classification problem.

Method and metrics. We use several classifiers to predict the quiz performance class from features. These classifiers include Logistic Regression (LR) [26], Multilayer Perceptron (MLP) [28], and Linear Discriminant Analysis (LDA) [27]. We

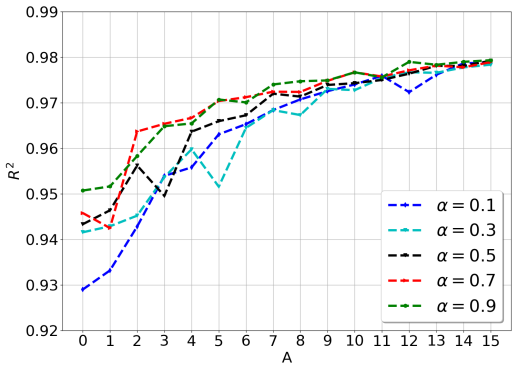


Fig. 2: R^2 metric on the reconstructor versus the number of additional complementary features A, for different values of α. The reconstructor's ability to reconstruct raw learner data improves as A increases, and the effect of α diminishes.

report the performance of our model generated features and baseline features using two standard evaluation metrics: (i) the log likelihood, and (ii) accuracy, which is simply the percent of learners for whom the quiz score class is predicted correctly. For both metrics, higher values imply higher prediction quality. In each case, we perform 5-fold cross validation and report the average metric values.

Classification performance. In Table 2, we show the prediction results for each algorithm on the features constructed by proposed GAN model compared with the baseline features ($\mathbf{x}_u^r$, $\mathbf{x}_u^{ae}$, $\mathbf{x}_u^g$, and $\mathbf{x}_u^{PCA}$). We have highlighted the entries in the table that have achieved the best performances when compared to the other feature types. We make a few observations:

Features constructed from our proposed GAN model, $\mathbf{x}_u^{gen}$, outperform other feature types, regardless the choice of the classifier. Prediction classifiers using $\mathbf{x}_u^{gen}$ as input tend to reach an accuracy of 0.63, beating classifiers built on all baseline features. PCA compressed features $\mathbf{x}_u^{PCA}$ have achieved matching performances, with an accuracy around 0.62 and a log likelihood value around -0.972. However, PCA compress features by projecting high dimensional input to lower dimensional data points via explained variance; thus these features are difficult to explain in real-world learning environment, sacrificing the other objective of interpretability that we aim to achieve.

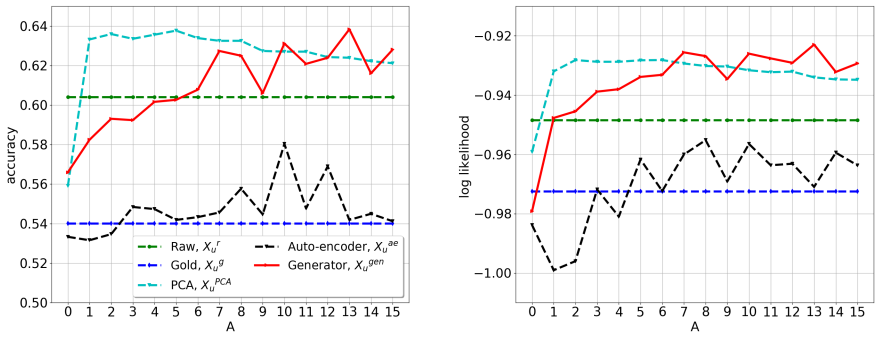
Prediction quality is reasonably invariant to the choice of classifier. Considering each feature type separately, the performances for all choices of algorithms are reasonably similar. The only variable changing here is the choice of input feature, further confirming the ability of $\mathbf{x}_u^{gen}$ to represent lower-dimensional information.

Varying the number of complementary features A. We also analyze the effect of the number of complementary features on prediction quality. With $A = 0$, we do not generate any additional complementary features and solely rely on $\mathbf{x}_u^f$ for reconstructor. On the other hand, with A chosen to be a larger number, we generate additional complimentary features in additional to $\mathbf{x}_u^f$, and use both to feed into the reconstructor.

Features	LR		MLP		LDA	
	Accuracy	Log likelihood	Accuracy	Log likelihood	Accuracy	Log likelihood
Raw, $\mathbf{x}_u^r$	0.604	-0.948	0.596	-0.937	0.585	-1.203
Gold, $\mathbf{x}_u^g$	0.540	-0.972	0.539	-0.971	0.543	***-1.121***
PCA, $\mathbf{x}_u^{PCA}$	0.624	-0.934	0.618	-0.988	0.617	-1.163
Auto-encoder, $\mathbf{x}_u^{ae}$	0.579	-0.949	0.552	-0.964	0.563	-1.157
GAN, $\mathbf{x}_u^{gen}$	***0.627***	***-0.923***	***0.631***	***-0.917***	***0.624***	-1.157

Table 2: Accuracy and log likelihood values for quiz response prediction with $A = 15$ and $\alpha = 0.1$ using different classifiers. The refined and complementary features from the generator achieves the highest performance.

Figure 3 shows the results. The accuracy tends to range between 0.52 and 0.64, with the PCA and generator features generally better. While PCA compressed features outperform all other feature types when A is small, $\mathbf{x}_u^{gen}$ shows a continuous trend to increase while A increases. When A is larger than 10, the classifier built on $\mathbf{x}_u^{gen}$ finally reaches an accuracy of around 0.63 and a log likelihood value around -0.93, beating $\mathbf{x}_u^r$, $\mathbf{x}_u^{ae}$, $\mathbf{x}_u^g$, and $\mathbf{x}_u^{PCA}$ in most cases.

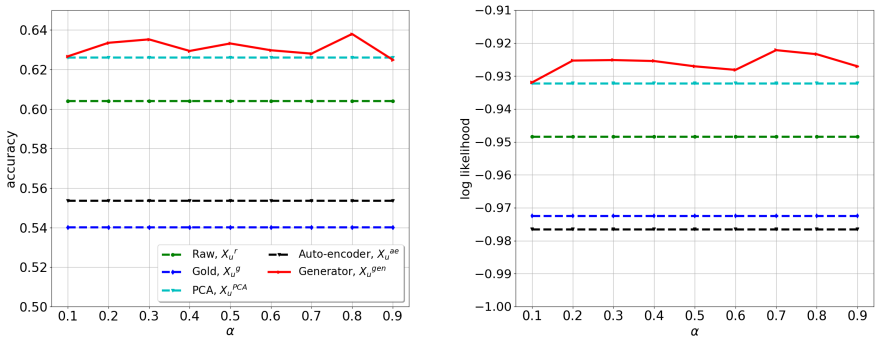


(a) Accuracy (b) Log likelihood

Fig. 3: Comparison of accuracy and log likelihood in predicting quiz score versus the number of complementary features (A) with $\alpha = 0.9$. The refined features and additional features from the generator outperform other compressed features at large A, while PCA features achieve the best performance at small A.

Varying α. Recall that the goal of the generator network is to learn features that strike a balance between two objectives: reconstructing raw learner data and fooling the discriminator network. We now investigate the impact of α, which controls the relative weight placed on each objective. In this experiment, we sweep the values of α as $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$, with A chosen to be 12 for optimal classification performance.

Figure 4 shows the accuracy and log likelihood values of quiz score prediction using the model-generated features and baseline features as we change α. Overall, we see that varying α does not result in significant changes in the performance of generated features; the generated features performs similarly to the PCA features and outperform other baseline features.



(a) Accuracy. (b) Log Likelihood.

Fig. 4: Comparison of accuracy and log likelihood in predicting quiz scores versus α. We see that the performance is relatively robust to the choice of α.

3.4 Feature interpretability

With the refined features closely resembling the human-crafted gold features, the additional complementary features can capture nuisances in raw learner behavioral data that are not fully captured by the human-crafted gold standard features. In order to promote the interpretability of our refined gold standard features and complementary features, we use the same generator defined in Section 2.2 with a small change to the generator loss. Specifically, we add an ℓ_1-norm penalty to the generator loss as $\lambda\|\mathcal{W}_G\|_1$ to promote sparsity of the connections between the raw features and the refined and complementary features. Here, $\lambda > 0$ is a tuning parameter and $\|\cdot\|_1$ denotes the ℓ_1-norm, i.e., the sum of the absolute values of each element.

To analyze interpretability, we investigate the learned weight matrix $\mathcal{W}_G$, finding the most representative raw features for each of the generated features. Specifically, for the n^{th} generated feature, we look into the n^{th} column of $\mathbf{W}$ and find the raw features whose corresponding weights have the largest absolute values.

Table 3 shows top 8 most representative raw features corresponding to 3 complementary features chosen. The raw features here are specified by two attributes: the segment and the quantity from Table 1. We observe that raw features from 9 specific segments are very representative of the generated features. On these segments, raw features corresponding to engagement and time spent measurements are more representative than those corresponding to views and expected time spent measurements. For many generated features, the top representative raw features consist of a mixture of various segments and measurements. Interestingly, while some features are associated with all positive weights (e.g., complementary feature 8), other features are associated with a mix of positive and negative weights with various segments and measurements (e.g., complementary features 1 and 4). This observation suggests that there are some important relationships between segments that are not captured by the human-crafted gold

standard features; the generator network thus enables us to gain further insight into the course content by analyzing raw learner behavioral data, in addition to analyzing the content itself.

Rank	Complementary feature 1		Complementary feature 4		Complementary feature 8	
	Weight	Raw feature	Weight	Raw feature	Weight	Raw feature
1	0.1740	02 ENG	-0.0802528	32 ENG	0.1041	14 TS
2	0.1200	03 ENG	0.0695	29 ETS	0.0522	12 Views
3	-0.0739	06 ENG	0.0588	26 ETS	0.0389	02 ETS
4	-0.0496	07 ENG	-0.0472	31 ENG	0.0297	12 TS
5	-0.0250	06 ETS	-0.0445	35 ENG	0.0201	01 TS
6	0.0219	08 ENG	-0.0425	19 ENG	0.0181	32 TS
7	-0.0190	07 ETS	-0.0259	27 TS	0.0121	06 Views
8	-0.0113	05 ENG	-0.0257	05 TS	0.0110	19 TS

Table 3: Selected complementary features with their top 8 representative raw features. Segment numbers are included, with ENG, TS and ETS denoting engagement, time spent and expected time spent, respectively. The complementary features reveal relationships between content that the gold standard features do not.

4 Conclusion and Future Work

In this work, we have developed a method for generating low-dimensional features to summarize learner behavioral data that promote both interpretability and predictability. Our method uses the generative adversarial network (GAN) framework to turn high-dimensional, raw learner behavioral data into a set of refined features that resemble the characteristics of a small number of human-crafted gold standard features. Additionally, our framework produces complementary features that capture the nuisances in raw data that are not captured by the gold standard features. By evaluating on data collected from an online corporate training course, we showed that the generated features were able to reconstruct raw data with high fidelity. We further demonstrated that the generated features were more predictive of learner quiz scores than features constructed by several baseline methods. Last but not least, we showed that the generated features were able to capture detailed learner-content interactions not available in the gold standard features.

There are several avenues of future work. First, we are attempting to incorporating other sources of data to describe the learners and the course, such as features that are specific to different content types (e.g., videos, slideshows, and PDFs), into the raw features. These data sources enable us to extract learner interactions with specific types of content, while generic features do not. We are also investigating other neural network architectures, such as recurrent neural networks (RNNs) to learn time-varying representations; this information will enable us to model the dynamics of learner behavior.

References

1. L. Allen, M. Jacovina, M. Dascalu, R. Roscoe, K. Kent, A. Likens, and D. McNamara. {ENTER}ing the time series {SPACE}: Uncovering the writing process through keystroke analyses. In *Proc. Intl. Conf. Educ. Data Min.*, pages 22–29, June 2016.
2. A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Engaging with massive online courses. In *Proc. Intl. Conf. World Wide Web*, pages 687–698, Apr. 2014.
3. T. Barnes. The Q-matrix method: Mining student response data for knowledge. In *Proc. AAAI Workshop Educ. Data Min.*, pages 1–8, July 2005.
4. B. Beheshti, M. Desmarais, and R. Naceur. Methods to find the number of latent skills. In *Proc. Intl. Conf. Educ. Data Min.*, pages 81–86, June 2012.
5. Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton, and D. Pritchard. Model-based collaborative filtering analysis of student response data: Machine-learning item response theory. In *Proc. Intl. Conf. Educ. Data Min.*, pages 95–102, June 2012.
6. C. Brinton, S. Buccapatnam, F. Wong, M. Chiang, and H. Poor. Social learning networks: Efficiency optimization for MOOC forums. In *Proc. IEEE Conf. Comput. Commun.*, pages 1–9, Apr. 2016.
7. C. Brinton and M. Chiang. MOOC performance prediction via clickstream data and social learning networks. In *Proc. IEEE Conf. Comput. Commun.*, pages 2299–2307, April 2015.
8. L. Chen, X. Li, Z. Xia, Z. Song, L. Morency, and A. Dubrawski. Riding an emotional roller-coaster: A multimodal study of young child's math problem solving activities. In *Proc. Intl. Conf. Educ. Data Min.*, pages 38–45, June 2016.
9. W. Chen, C. Brinton, D. Cao, and M. Chiang. Behavior in social learning networks: Early detection for online short-courses. In *Proc. IEEE Conf. Comput. Commun.*, pages 1–9, May 2017.
10. C. Coleman, D. Seaton, and I. Chuang. Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Proc. ACM Conf. Learn at Scale*, pages 141–148, Mar. 2015.
11. A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-adapt. Interact.*, 4(4):253–278, Dec. 1994.
12. B. Gelman, M. Revelle, C. Domeniconi, A. Johri, and K. Veeramachaneni. Acting the same differently: A cross-course comparison of user behavior in MOOCs. In *Proc. Intl. Conf. Educ. Data Min.*, pages 376–381, June 2016.
13. I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep Learning*. MIT Press, 2016.
14. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
15. P. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proc. ACM Conf. Learn at Scale*, pages 41–50, Mar. 2014.
16. S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in MOOCs using learner activity features. In *Proc. European MOOCs Stakeholders Summit*, pages 58–65, Feb. 2014.

17. G. Hinton and T. Sejnowski. *Unsupervised Learning: Foundations of Neural Computation*. MIT press, 1999.
18. D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Intl. Conf. Learn. Represent.*, May 2015.
19. R. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proc. Intl. Conf. Learn. Analyt. Knowl.*, pages 170–179, Apr. 2013.
20. S. Klingler, R. Wampfler, T. Kaser, B. Solenthaler, and M. Gross. Efficient feature embeddings for student classification with variational auto-encoders. In *Proc. Intl. Conf. Educ. Data Min.*, pages 72–79, June 2017.
21. K. Koedinger, J. Kim, J. Jia, E. McLaughlin, and N. Bier. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In *Proc. ACM Conf. Learn at Scale*, pages 111–120, Mar. 2015.
22. A. Lan, A. Waters, C. Studer, and R. Baraniuk. Sparse factor analysis for learning and content analytics. *J. Mach. Learn. Res.*, 15:1959–2008, June 2014.
23. A. S. Lan, C. G. Brinton, T. Yang, and M. Chiang. Behavior-based latent variable model for learner engagement. In *Proc. Intl. Conf. Educ. Data Min.*, pages 64–71, June 2017.
24. K. Lee, J. Chung, Y. Cha, and C. Suh. ML approaches for learning analytics: Collaborative filtering or regression with experts? *online: http://ml4ed.cc/attachments/LeeLCCS.pdf*, Dec. 2016.
25. J. McBroom, B. Jeffries, I. Koprinska, and K. Yacef. Mining behaviours of students in autograding submission system logs. In *Proc. Intl. Conf. Educ. Data Min.*, pages 159–166, June 2016.
26. S. Menard. *Applied Logistic Regression Analysis*. SAGE Publications, 2018.
27. S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Mullers. Fisher discriminant analysis with kernels. In *Proc. IEEE Signal Process. Workshop*, pages 41–48, 1999.
28. S. Pal and S. Mitra. Multilayer Perceptron, Fuzzy Sets, and Classification. *IEEE Trans. Neural Netw.*, 3(5):683–697, Sep. 1992.
29. J. Reich, B. Stewart, K. Mavon, and D. Tingley. The civic mission of MOOCs: Measuring engagement across political differences in forums. In *Proc. ACM Conf. Learn at Scale*, pages 1–10, Apr. 2016.
30. S. Slater, R. Baker, J. Ocumpaugh, P. Inventado, P. Scupelli, and N. Heffernan. Semantic features of Math problems: Relationships to student learning and engagement. In *Proc. Intl. Conf. Educ. Data Min.*, pages 223–230, June 2016.
31. S. Tomkins, A. Ramesh, and L. Getoor. Predicting post-test performance from online student behavior: A high school MOOC case study. In *Proc. Intl. Conf. Educ. Data Min.*, pages 239–246, June 2016.
32. D. Yang, R. Kraut, and C. Rosé. Exploring the effect of student confusion in massive open online courses. *J. Educ. Data Min.*, 8(1):52–83, 2016.
33. Tsung-Yen Yang, Christopher G Brinton, Carlee Joe-Wong, and Mung Chiang. Behavior-Based Grade Prediction for MOOCs Via Time Series Neural Networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(5):716–728, 2017.