Predictive Learning Analytics for Video-Watching Behavior in MOOCs

Madhumitha Shridharan[1], Ashley Willingham[1], Jonathan Spencer[2], Tsung-Yen Yang[2], and Christopher Brinton[2]
Departments of [1]Operations Research and Financial Engineering and [2]Electrical Engineering
Princeton University

***Abstract*—In this paper, we develop Predictive Learning Analytics (PLA) methodology for learner video-watching behavior in Massive Open Online Courses (MOOCs). After defining features to summarize such behavior from clickstream measurements, we perform a statistical analysis of a real-world MOOC dataset and uncover several interesting relationships between the different features. Motivated by this analysis, we propose three algorithms for predicting future video-watching behavior, which incorporate biases for learners and videos, collaborative filtering across videos, and regularization to reduce overfitting. Through evaluation on our dataset, we find that the predictors obtain low RMSE overall, and that augmenting the bias predictor with either collaborative filtering or regularization improves prediction quality in eight out of nine cases.**

I. Introduction

Massive Open Online Courses (MOOCs) have spread access to education worldwide, allowing anyone with an Internet connection to take courses from providers such as Coursera, Udacity, and Udemy. Despite the conveniences they offer, the lower completion rates and engagement levels of MOOCs as compared with traditional courses have been the subject of criticism [1]. This has in turn motivated research aimed at improving the quality of learning in these courses, such as through content analytics and automated personalization [2].

Predictive Learning Analytics (PLA) is a particular subset of this research concerned with deriving analytics from data collected about learners as they take courses. This data ranges from performance-based measures like quiz question responses to behavior-based measures such as clickstream actions on lecture videos. The goal of PLA is to provide instructors with intelligence on how learning experiences in a course may be improved, both for individual learners and for overall courses. A common example is the use of prior quiz performance data to predict how a new group of learners may perform on a quiz. If the predicted scores are deemed inadequate, the course can be modified.

Beyond quiz performance, PLA for how learners may interact with course *content* can provide fine-granular insights into what particular content may benefit from modification or adaptation [2]. For example, if it is estimated that a learner will pause and rewind in a lecture video significantly more than average, the learner may benefit from viewing a more rudimentary version first. On the other hand, if a high average playback rate and several fast forwards are predicted, then the learner could be recommended/directed to the key parts.

Motivated by this, we ask the following two questions in this paper: Do statistical patterns exist in the ways learners watch lecture videos? If so, can a learner's behavior in a future course video be predicted from prior behavior?

A. Related Work

Prior work on PLA for MOOC has focused mainly on modeling learning outcomes, e.g., quiz scores or knowledge transfer. Algorithms have been developed to predict how learners will perform on particular assessments [3], [4], how average grades will change over time [1], [5], whether learners will drop out of the course [6], [7], and whether they will obtain certificates [8], [9]. Different from these, our work is focused on more fine-granular predictive analytics for how learners behave in learning modules (course videos).

In this regard, a few works have studied learner behavioral data in MOOCs, including video-watching clickstream measurements and Social Learning Network (SLN) interactions. Some have focused on an exploratory analysis of such data and its associated analytics, e.g., [10] modeled transitions between learning activities and [11] quantified parameters to summarize learner behavior in an SLN. Others have developed predictors for learning outcomes in MOOCs based on behavioral data [4], [7], [6], [12], [1]. More specifically, [7], [12] mined clickstream actions for recurring subsequences of learner behavior (e.g., reviewing the same content several times) associated with quiz performance and/or course completion, while [4], [6], [1] defined intuitive sets of features that summarize behavior (e.g., fraction of time playing a video) and used those to predict outcomes. Our work is perhaps most similar to [4], [1] in that we use similar sets of human-crafted features, but we are focused on predicting future behavior (i.e., the feature values themselves) rather than eventual outcomes.

B. Our Methodology and Contributions

In this paper, we develop novel PLA methodology for predicting future learner video-watching behavior. To do so, we use a dataset of roughly one million clickstream events generated by learners watching 92 videos in a MOOC.

We begin in Section II by describing our dataset and the machine learning features that summarize each learner's video-watching behavior. In doing so, we perform a statistical analysis of the features which leads to several insights both for the development of our behavioral prediction algorithm and for instructor analytics. For example, we identify a significant negative association between the fraction of a video a learner completes and most of their other behaviors exhibited on the video, i.e., those who complete more content also tend to spend less time on it overall, which can inform content changes.

Then, in Section III, we propose the predictors for learner video-watching behavior. Given the significant findings from our linear regression in Section II, we start with least-squares bias predictors that includes biases for each learner and each video with respect to each feature. We then introduce nearest-neighbor-based collaborative filtering on the errors produced by the bias predictor on the training set, and finally a regularization method to prevent overfitting. In evaluating the predictors in terms of RMSE on the evaluation set, we find that regularization and leveraging similarities between videos improves our initial least-squares bias prediction model on four out of nine features.

II. Features and Dataset

After introducing our dataset (Sec. II-A) and features (Sec. II-B), we perform a statistical analysis (Sec. II-C) to derive instructor analytics and motivate our predictors in Sec. III.

A. Course Dataset

The dataset we use was collected during our fall 2012 MOOC offering of "Networks: Friends, Money, and Bytes" (NFMB) on Coursera [13]. This course consists of 92 lecture videos across 20 lectures, with a quiz ending each video.

A total of 3,976 learners appeared in the dataset. However, the number of videos that learners watched varied greatly due to the inherent dropoff rates of MOOCs [14]. In total, we ended up with 29,304 unique learner-video pairs (i.e., instances of a learner watching a video).

Coursera (and other MOOC platforms) records user interactions with lecture videos as clickstream events. Each time a learner makes an action on the video – play, pause, skip, playback ratechange, exit, and so on – the action is recorded along with its timestamp, playback position, and user/video identification information. Roughly one million events were generated across the user-video pairs in this dataset.

B. Feature Specification

For each learner-video pair, we compute nine machine learning features based on the clickstream events. We omit the exact formulas for brevity, referring the interested reader to [1] for similar derivations. The features, and the intuitions/interpretations behind their values, are as follows:

1) Fraction completed (fracComp): The percentage of the video that the learner watched. Repeated segments are not counted more than once, so it must be between 0 and 1. FracComp is a rudimentary indicator of a learner's engagement level with a video.

2) Fraction spent (fracSpent): The amount of (real) time the learner spent on the video (i.e., while playing or paused) divided by its total playback time. It compares the total time the learner devoted to watching the video to the length of the video itself, so it can take values greater than 1. FracSpent also conveys how much time it took a learner to digest the material presented, and is therefore a reflection of the clarity of the presentation and content difficulty. An abnormally high fracSpent value could also indicate that the learner engaged in off-task behavior.

TABLE I
Average and standard deviation of the video-watching behavioral features, computed over all learner-video pairs.

Feature X	Mean X	St. dev. s_X
fracSpent	0.895	0.461
fracComp	0.767	0.340
fracPlayed	0.985	3.723
fracPaused	0.370	0.756
numPauses	2.825	59.100
avgPBR	1.104	0.315
stdPBR	0.014	0.050
numRWs	2.238	15.564
numFFs	1.567	6.369

3) Fraction played (fracPlayed): The amount of the video that the learner played, with repetitions, divided by its total playback time. Unlike fracComp, fracPlayed counts repeated segments more than once, and thus can take values ≥ 1.

4) Fraction paused (fracPaused): The amount of time the learner spent paused on the video, relative to its total playback time. It is possible for fracPaused to take values greater than 1. Larger values of this feature can convey that the learner exhibited additional effort to internalize the material, although an unusually high fracPaused probably indicates the learner was engaged in off-task behavior.

5) Number of pauses (numPauses): The number of times the learner paused the video. Similar to fracPaused, this feature is an indicator of the additional effort the learner exhibited to internalize the material. Further, an abnormally high numPauses could indicate that the presentation of the material is unclear, or that the content is too difficult.

6) Average playback rate (avgPBR): The time-average of the playback rate that the learner had selected while in the playing state.[1] Analysis of avgPBR could indicate whether the material was presented too fast or too slow for learners.

7) Standard deviation of playback rate (stdPBR): The standard deviation of playback rates selected by the learner over time.

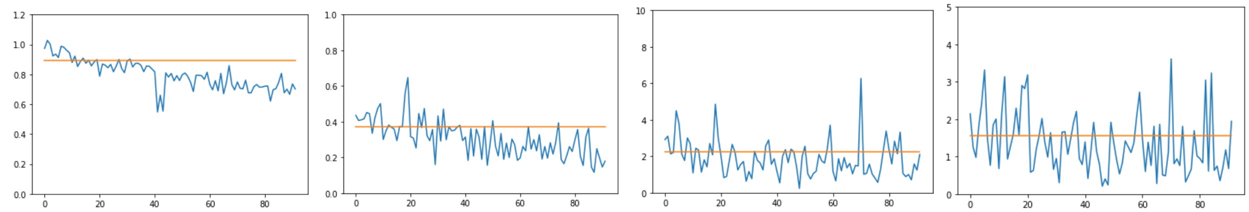
8) Number of rewinds (numRWs): The number of times the learner skipped backwards in the video. A higher value of this feature indicates the learner exerted more effort to understand the material. Easier material may have smaller numRWs.

9) Number of fast forwards (numFFs): The number of times the learner skipped forward in the video. A higher value in the presence of low fracComp indicates the learner felt this content unnecessary. Material that learners were already familiar with would tend to have higher numFFs.

C. Statistical Analysis

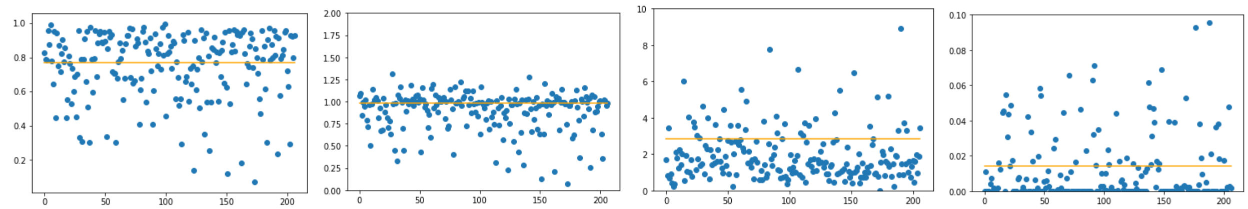
With the features defined, we now perform a statistical analysis to gain insight into learner video-watching behavior.

[1]The player on Coursera allows rates between 0.5x and 2.0x the default speed, in increments of 0.25x.



(a) fracComp (b) fracPaused (c) numRWs (d) numFFs

Fig. 1. Evolution of average feature values $\bar{X}(v)$ over videos v. Overall, several videos v deviate substantially from $\bar{X}$, implying that videos have unique biases with respect to behavior that we can leverage in the development of our predictors in Section III.



(a) fracComp (b) fracPlayed (c) numPauses (d) stdPBR

Fig. 2. Variation of average feature values $\bar{X}(u)$ over learners u. Overall, several learners u deviate substantially from $\bar{X}$, implying that learners have unique biases with respect to behavior that we can leverage in the development of our predictors in Section III.

Statistics across learners and videos. Table I gives the mean $\bar{X}$ and standard deviation s_X of each feature X, computed over all learner-video pairs excluding outliers.[2] A mean fracComp of 0.767 indicates that on average, learners visited about 75% of the content in videos and were thus well engaged overall. The mean fracSpent and fracPlayed confirm this, with values close to 1. At the same time, fracPaused, numPauses, and numRWs indicate that learners tended to spend a significant amount of time reflecting on material: they paused for about 40% of the length of the videos, while pausing and rewinding about 3 and 2 times respectively on average per video. This may indicate that the material presented was challenging overall, further supported by numRWs being greater than numFFs on average. Videos with higher-than-average values for the numRWs and numPauses features could be flagged as challenging and instructors could be directed to simplifying or better explaining them.

On the other hand, avgPBR indicates that if learners changed the speed of video playback at all, it tended to be increased. Learners may be recommended to slow their speed if necessary.

Statistics by video. We are interested to see how the feature values change by video, as we expect that different videos induce different behavior. Figure 1 plots the evolution of the average $\bar{X}(v)$ over the videos v for four representative features, compared with the global mean $\bar{X}$. Overall, we see that several videos v deviate substantially from $\bar{X}$, indicating that videos may have unique behavioral biases. We will leverage this in the development of our predictors in Section III.

[2]Outliers were particularly present in fracSpent and fracPaused: when learners become distracted, pause, and return to the video much later, they will naturally register much larger-than-normal time spent in the paused state.

More specifically, we make a number of interesting findings from Figure 1. First, as the course progresses, many of the features associated with learner engagement like (a) fracComp, (b) fracPaused, (c) numRWs, and (not shown) fracSpent each tend to decrease. This is surprising as we would expect learners who have not yet dropped out of the course to be more motivated, and thus to exhibit higher engagement. One possible conclusion, then, is that the content itself tends to become less engaging, or more difficult to grasp, as the course progresses. Instructors could use such information to identify the set of videos that do not meet learner expectations and work to improve their content and delivery.

In particular, the fracComp feature experiences an abruptly large drop in videos 42-44. Learners may be choosing to not complete or skip over a substantial amount of content here for reasons that the instructor can address. On the other hand, fracPaused exhibits an abruptly large increase for videos 18-19, which may imply that this material is particularly challenging and requires additional explanation from the instructor.

For numRWs and numFFs, there is no distinct trend or abrupt deviations for particular videos. The variance around $\bar{X}$ is large, however, a point which also manifested itself in Table I. The fact that each video tends to be far from the global mean indicates that bias values may work particularly well in predicting these features. This point will be reinforced in Section III when we see that collaborative-filtering-based prediction does not improve performance as much as expected due to these large biases.

Statistics by learner. Similarly, we are interested in how the feature values change across learners. To investigate this, we take the 208 learners who watched at least 30% of the videos and calculate their average feature values $\bar{X}(u)$. Four representative features are plotted in Figure 2. Overall, as with

TABLE II

MULTIVARIATE LINEAR REGRESSION COEFFICIENTS FROM FITTING EACH FEATURE TO THE OTHERS (ROWS ARE TARGET VARIABLES, COLUMNS ARE INDEPENDENT VARIABLES). COEFFICIENTS WITH A STATISTICALLY SIGNIFICANT $p \leq 0.05$ ARE MARKED IN BOLD.

Feature	fracSpent	fracComp	fracPlayed	fracPaused	numPauses	avgPBR	stdPBR	numRWs	numFFs
fracSpent	-	**-0.03674**	-0.01748	0.00400	0.00019	-0.02688	-0.00581	-0.00116	-0.00982
fracComp	**-0.01943**	-	**0.63652**	**-0.02982**	**-0.06065**	**0.12973**	**-0.04763**	**-0.21779**	**-0.06057**
fracPlayed	-0.00870	**0.59933**	-	0.02567	**0.15506**	**0.11627**	0.01726	**0.28166**	**-0.07545**
fracPaused	0.00400	**-0.05641**	0.05157	-	**0.03605**	-0.00208	**0.01040**	-0.02419	0.01810
numPauses	0.00018	**-0.10206**	**0.27709**	**0.03208**	-	**-0.06885**	**0.03783**	**0.18274**	**0.00670**
avgPBR	-0.02199	**0.20078**	0.19111	**-0.00170**	**-0.06333**	-	**0.23827**	-0.05171	**0.08849**
stdPBR	-0.00542	**-0.08403**	**0.03235**	**0.00970**	**0.03966**	0.27156	-	-0.00349	-0.00808
numRWs	-0.00076	**-0.26960**	0.37027	**-0.01583**	0.13443	-0.04135	**-0.00245**	-	**0.42411**
numFFs	-0.00736	**-0.08585**	-0.11356	0.01356	**0.00564**	0.08103	**-0.00648**	**0.48559**	-

the videos, we see that many of the learners deviate from $\bar{X}$ substantially, motivating learner biases in the predictors.

Linear regression. Finally, we investigate whether general relationships between the features exist. We do this through a series of multivariate linear regressions, one per feature. In each regression, one feature is treated as the target variable and all others as independent variables, using data from the 535 learners who watched at least 15% of the videos. Table II gives the learned coefficients, with each row corresponding to a different regression. Coefficients that were statistically significant (p-value below 0.05) are emphasized in bold.

These coefficients have several interesting implications. For one, note that all features except fracPlayed and avgPBR have a negative impact on fracComp. This implies that learners who skip, speed, and/or pause are targeting specific points in the videos at the expense of completing the entirety of the content. A learner predicted to have high numFFs could be given an abridged version of the video, while those anticipated to have high fracPaused, numPauses, and/or numRWs could be presented with remediation videos to preemptively address potential confusions. The negative coefficient of stdPBR could also indicate that the instructor spent too much time on rudimentary material (prompting learners to increase the playback rate), but too little time on challenging content (prompting them to slow it down). The instructor may consider better aligning the speed of content delivery with its difficulty.

Surprisingly, unlike fracComp, fracPlayed is positively correlated with numPauses and numRWs. This difference may be explained by video difficulty: when learners pause and skip back, they may repeat the same content multiple times, which will increase fracPlayed but not fracComp.

III. BEHAVIORAL PREDICTION

We now use the statistical analysis to formulate (Sec. III-A) and evaluate (Sec. III-C) video-watching behavior predictors.

A. Formulating Predictors

Training and testing sets. Due to the high drop-off rates we saw in Sec. II-C, we will build and evaluate our predictors using the set of 535 learners u who watched at least 15% of the videos v. Letting $\mathcal{S}$ be the set of learner-video pairs in this category, we divide $\mathcal{S}$ into training $\mathcal{C}$ and test $\mathcal{E} = \mathcal{S} \setminus \mathcal{C}$ sets. We will explain these partitions further in Section III-B.

Least-squares bias predictors. Given that our statistical analysis in Sec. II revealed the presence of bias values, we start with predictors that learn feature-specific biases for each learner and each video. Formally, letting b_u^X and b_v^X be the biases for u and v with respect to feature X, the prediction $\hat{x}_{uv}$ for the value of X taken by learner u on video v is

$$\hat{x}_{uv} = \bar{X} + b_u^X + b_v^X$$

where $\bar{X}$ is the global mean of the feature. We determine the model variables b_u^X and b_v^X by solving the following least-squares optimization problem over the training set $\mathcal{C}$:

$$\text{minimize}_{\{b_u^X, b_v^X\}} \sum_{(u,v) \in C} (x_{uv} - \hat{x}_{uv})^2$$

where x_{uv} is the actual value of feature X taken by $(u, v) \in \mathcal{C}$.

Collaborative filtering predictors. Next, we perform nearest-neighbor-based collaborative filtering on the errors produced by the least-squares bias predictor $\hat{x}_{uv}$ on the training set. We take neighbors across videos. Formally, letting v and i be two videos, we first quantify the distance d_{vi} between them as:

$$d_{vi} = \frac{\sum_{u \in \mathcal{U}_{vi}} \tilde{x}_{uv} \tilde{x}_{ui}}{\sqrt{\sum_{u \in \mathcal{U}_{vi}} (\tilde{x}_{uv})^2} \sqrt{\sum_{u \in \mathcal{U}_{vi}} (\tilde{x}_{ui})^2}}$$

where $\mathcal{U}_{vi}$ is the set of learners u who watched both videos v and i in the training set, and $\tilde{x}_{uv} = x_{uv} - \hat{x}_{uv}$ is the bias predictor error. The more similar the bias predictor errors $\tilde{x}_{uv}$ and $\tilde{x}_{ui}$ of the videos, then, the smaller their distance metric.

Given a query video v, we then select its top-K nearest neighbor videos based on the maximum values of $|d_{vi}|$ across i, i.e., those with the highest similarity. Denoting $\mathcal{K}_v$ as the resulting neighborhood of v, we then form the collaborative filtering-based prediction $\hat{x}_{uv}^N$ of the value taken by feature X for the learner-video pair (u, v) as

$$\hat{x}_{uv}^N = (\bar{X} + b_u^X + b_v^X) + \frac{\sum_{i \in \mathcal{K}_v, (u,i) \in C} d_{vi} \tilde{x}_{ui}}{\sum_{i \in \mathcal{K}_v, (u,i) \in C} |d_{vj}|}$$

i.e., the summation is taken only over videos i for which learner u has watched i in the training set. The specific weighting scheme here is chosen based on [15]. The number of neighbors K is a parameter we will vary in our evaluation.

Regularized predictors. Finally, we introduce regularization to the collaborative filtering predictors to analyze and prevent overfitting on the training set. This is done by adjusting the least-squares minimization for the bias terms to include an L2 regularization penalty term, weighted by a tradeoff parameter λ:

$$\text{minimize}_{\{b_u^X, b_v^X\}} \sum_{(u,v)\in C} (x_{uv} - \hat{x}_{uv})^2 + \lambda (\sum_u \left(b_u^X\right)^2 + \sum_v \left(b_v^X\right)^2)$$

We then proceed to calculate the collaborative-filtering predictions $\hat{x}_{uv}^N$ using the regularized versions of the $\hat{x}_{uv}$. The effect of λ will be considered in our evaluation.

B. Evaluation Implementation and Metric

We now briefly discuss our implementation. We solve the minimizations for the regularized and non-regularized biases b^X via the standard normal equations for least squares. The nearest neighbor algorithm was then implemented de novo.

To partition the dataset $\mathcal{S}$ into training $\mathcal{C}$ and test $\mathcal{E}$ sets, we first remove any learner-video pairs from $\mathcal{S}$ that were statistical outliers either for the fracSpent or fracPaused features. Then, in each evaluation iteration, we randomly split 90% of $\mathcal{S}$ into $\mathcal{C}$, and hold out the remaining 10% for $\mathcal{E}$. After training the predictors for each feature on $\mathcal{C}$, we evaluate model performance on $\mathcal{E}$ according to the Root Mean Squared Error (RMSE) metric [4]:

$$\text{RMSE} = \sqrt{\sum_{(u,v)\in\mathcal{E}} \frac{(x_{ui} - \hat{x}_{ui}^p)^2}{|\mathcal{E}|}}$$

where $\hat{x}_{ui}^p$ is the predicted value of $(u, v) \in \mathcal{E}$ for the model under consideration. The lower the RMSE, the higher the prediction quality. For each model and each choice of parameter K and λ, we repeat this random partition, training, and RMSE evaluation ten times and report the average result.

As stated, we sweep over several values of the number of neighbors K and the regularization parameter λ in our evaluation. For each feature, we consider $K \in \{2, 5, 10, 20, 50\}$ and $\lambda \in \{0, 0.01, 0.05, 0.1, 0.5, 2, 5, 10\}$.

C. Results and Discussion

We divide the evaluation into two parts: (i) optimizing K compared with the biases alone, and (ii) optimizing λ compared with the other methods.

Optimizing K. Table III gives the RMSE values obtained by the bias and collaborative filtering predictors, with the best observed neighborhood size K given in each case. These results indicate reasonable quality of the predictions overall: the RMSE of 0.2865 obtained by collaborative filtering on fracComp, for example, represents a 43% improvement over a random predictor (which is expected to have an RMSE of 0.5 since fracComp ranges between 0 and 1). This demonstrates the overall ability to predict a learner's future behavior.

Further, we see that using a video's nearest neighbors improved the quality of our predictions for all features except numRWs and numFFs. This confirms our belief that individual learners tend to have some similarities in the way they watch videos, irrespective of the particular content being covered.

TABLE III
RMSE VALUES OBTAINED BY THE BIAS AND COLLABORATIVE FILTERING PREDICTORS FOR THE BEST CHOICE OF K. COLLABORATIVE FILTERING IMPROVES MODEL QUALITY IN SOME CASES BUT NOT OTHERS.

Feature	Bias RMSE	Collaborative RMSE	Best K
fracSpent	0.37911	0.36836	5
fracComp	0.29513	0.28650	10
fracPlayed	0.48308	0.37945	10
fracPaused	0.71526	0.70373	10
numPauses	3.44089	3.15003	5
avgPBR	0.22700	0.21478	5
stdPBR	0.04183	0.04116	50
numRWs	5.81223	5.99765	2
numFFs	4.80595	5.08605	2

Note that the features vary in the best observed choice of K. In particular, $K = 50$ for stdPBR is much larger than that of other features. It is possible that this is due to fewer choices of the speed at which videos can be watched: a given query video therefore may have a stdPBR value similar to that of a large number of videos, which can be leveraged to predict the query video's stdPBR. On the other extreme, numRWs and numFFs obtained the highest quality on the bias predictor, i.e., $K = 0$. The fact that collaborative filtering does not improve quality in these cases implies that individual learners may exhibit substantial variations in the skips they register on different videos, e.g., due to variations in content difficulty.

The majority of the features had their best observed values of K at either 5 or 10. In general, the models improve as the number of neighbors taken increases, but deteriorates at a point when neighbors become less relevant; adding too many non-similar features can blunt the effect of those with strong similarity (or dissimilarity). There are several reasons too as to why a learner may not behave according to the similarity values between videos. For one, the environment in which the learner watches a video might impact the number of times they pause, fast forward, or rewind. If a learner watches one video in a noisy, distracted environment, they may lose focus and be forced to skip around more, while if they watch a different one in a quiet environment this may happen less.

Optimizing λ. In Figure 3, we give the results on each feature for (i) the least-squares bias predictor, (ii) two different versions of collaborative filtering, $K = 5$ and $K = 50$, and (iii) regularization for the best choice of λ with $K = 2$.

Overall, we see that collaborative filtering with or without regularization is able to improve performance over the biases-only predictors in all cases except for numFFs. Each accounts for about half of the cases. The most improvement gleaned by the regularized collaborative filtering model was in numRWs, with the second most successful being fracPlayed; this implies that these features were the most prone to overfitting in the initial least-squares bias predictor.

Prediction quality for the numFFs feature was maximized by the unaugmented least-squares bias predictor. Referring back to Figure 1, we see that this feature exhibits the largest variation from video-to-video, which is consistent with there being a lack of similarity between videos. Intuitively, each

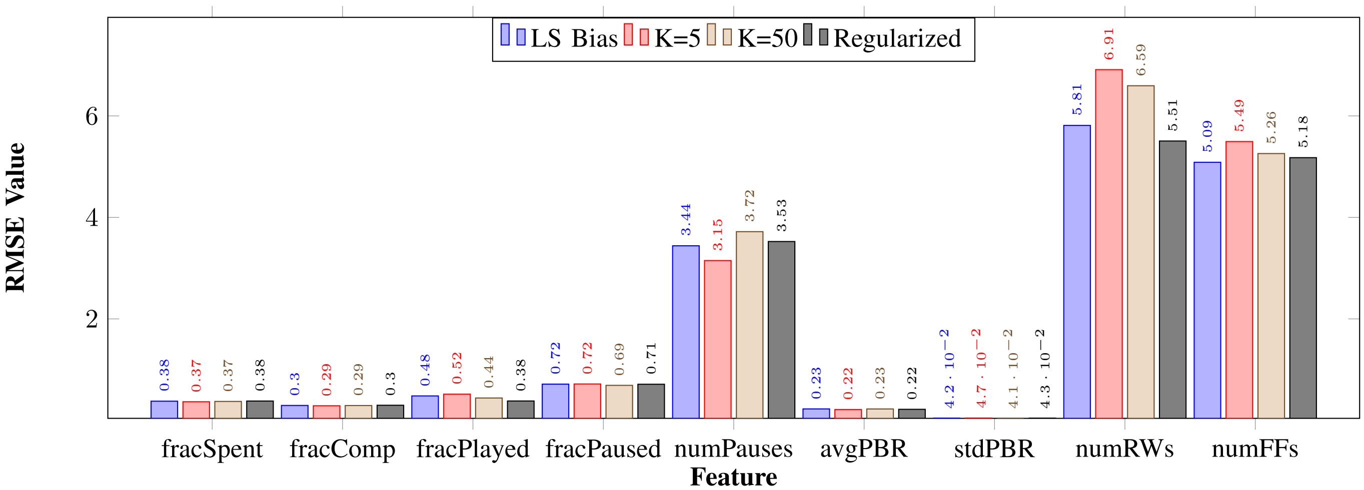


Fig. 3. RMSE values obtained on each feature for four different predictors: least-squares biases, collaborative filtering with $K = 5$ and $K = 50$, and regularization with $K = 2$ and the best choice of λ. Regularization obtains the highest quality in $4/9$ cases.

learner will possess a different level of background knowledge and interest before watching each video, so the decision of how much content to skip over will be independent of what the learner has done in other videos.

IV. Conclusion and Future Work

In this paper, we developed novel Predictive Learning Analytics (PLA) methodology for behavior-based prediction of a learner's future video-watching behavior. After defining several features to summarize a learner's behavior while watching a video, we performed a statistical analysis on a Massive Open Online Course (MOOC) dataset that led to several insights both for instructor analytics and for the development of behavioral prediction algorithms. We started our predictive modeling with a least-squares bias predictor that included biases for each learner and each video with respect to each feature. We then introduced nearest-neighbor-based collaborative filtering on the errors produced by the bias predictor, and finally a regularization method to prevent overfitting. In evaluating the predictors on our dataset, we obtained overall low errors in predicting the feature values, and found that augmenting the bias predictor with collaborative filtering and/or regularization led to better performance for all features except one. In presenting these results, we also were able to derive several analytics that could be used by instructors to improve course delivery.

Future work may consider an experiment in which the number of neighbors is varied alongside the regularization parameter to jointly model the effect of these parameters. Additionally, it would be interesting to compute the neighborhood predictor using similarities between learners and compare its error with our current neighborhood predictor that uses similarities between videos; it is possible that a combination of both types would perform the best. Other types of machine learning and deep learning algorithms could be considered as well, perhaps in an ensemble predictor.

References

[1] T.-Y. Yang, C. G. Brinton, C. Joe-Wong, and M. Chiang, "Behavior-based grade prediction for moocs via time series neural networks," *IEEE JSTSP*, 2017.
[2] C. G. Brinton and M. Chiang, "Social Learning Networks: A Brief Survey," in *IEEE CISS*, 2014, pp. 1–6.
[3] A. S. Lan, C. Studer, and R. G. Baraniuk, "Time-varying learning and content analytics via sparse factor analysis," in *SIGKDD*. ACM, 2014, pp. 452–461.
[4] C. G. Brinton and M. Chiang, "MOOC Performance Prediction via Clickstream Data and Social Learning Networks," in *IEEE INFOCOM*, 2015, pp. 2299–2307.
[5] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep Knowledge Tracing," in *NIPS*, 2015, pp. 505–513.
[6] J. Qiu, J. Tang, T. X. Liu, J. Gong, C. Zhang, Q. Zhang, and Y. Xue, "Modeling and Predicting Learning Behavior in MOOCs," in *ACM WSDM*, 2016, pp. 93–102.
[7] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg, "Your Click Decides Your Fate," in *ACL EMNLP*, 2014, pp. 3–14.
[8] H. Lakkaraju, E. Aguiar, C. Shan, D. Miller, N. Bhanpuri, R. Ghani, and K. Addison, "A Machine Learning Framework to Identify Students at Risk of Adverse Academic Outcomes," in *ACM SIGKDD*, 2015, pp. 1909–1918.
[9] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Engaging With Massive Online Courses," in *WWW 2014*. ACM, 2014, pp. 687–698.
[10] C. Geigle and C. Zhai, "Modeling mooc student behavior with two-layer hidden markov models," in *L@S*. ACM, 2017, pp. 205–208.
[11] C. G. Brinton, S. Buccapatnam, F. M. F. Wong, M. Chiang, and H. V. Poor, "Social Learning Networks: Efficiency Optimization for MOOC Forums," in *IEEE INFOCOM*, 2016.
[12] C. G. Brinton, S. Buccapatnam, M. Chiang, and H. V. Poor, "Mining MOOC Clickstreams: Video-Watching Behavior vs. In-Video Quiz Performance," *IEEE TSP.*, vol. 64, no. 14, pp. 3677–3692, 2016.
[13] M. Chiang. Networks: Friends, money, and bytes. [Online]. Available: https://www.coursera.org/course/friendsmoneybytes
[14] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong, "Learning about social learning in moocs: From statistical analysis to generative model," *IEEE Trans. Learning Technol.*, vol. 7, pp. 346–359, 2014.
[15] A. Toscher and M. Jahrer, "Collaborative filtering applied to educational data mining," *KDD cup*, 2010.