A Bandit You Can Trust

Ethan Prihar
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
ebprihar@wpi.edu

Adam Sales
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
asales@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
nth@wpi.edu

ABSTRACT

This work proposes Dynamic Linear Epsilon-Greedy, a novel contextual multi-armed bandit algorithm that can adaptively assign personalized content to users while enabling unbiased statistical analysis. Traditional A/B testing and reinforcement learning approaches have trade-offs between empirical investigation and maximal impact on users. Our algorithm seeks to balance these objectives, allowing platforms to personalize content effectively while still gathering valuable data. Dynamic Linear Epsilon-Greedy was evaluated via simulation and an empirical study in the ASSISTments online learning platform. In simulation, Dynamic Linear Epsilon-Greedy performed comparably to existing algorithms and in ASSISTments, slightly increased students' learning compared to A/B testing. Data collected from its recommendations allowed for the identification of qualitative interactions, which showed high and low knowledge students benefited from different content. Dynamic Linear Epsilon-Greedy holds promise as a method to balance personalization with unbiased statistical analysis. All the data collected during the simulation and empirical study are publicly available at https://osf.io/zuwf7/.

CCS CONCEPTS

• **Applied computing** → **Education**; **Distance learning**; **Computer-assisted instruction**.

KEYWORDS

Contextual Bandit Algorithms, Online Learning, Empirical Studies

ACM Reference Format:
Ethan Prihar, Adam Sales, and Neil Heffernan. 2023. A Bandit You Can Trust. In *UMAP '23: Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization (UMAP '23), June 26–29, 2023, Limassol, Cyprus.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3565472.3592955

1 INTRODUCTION

Online learning platforms have become significantly more popular in recent years due to the prevalence of technology in the classroom and the transition to remote learning due to the global pandemic [15]. This has allowed students that would have otherwise been unable to attend class to receive instruction and enabled researchers to perform large-scale investigations into various instructional methods. However, these opportunities have come with challenges.

There are countless choices to be made when structuring online instruction. Should lessons be student-pace or teacher-paced? Should the assignments have multiple-choice or open-ended questions? What criteria should be used to determine when a student has mastered the material? When students are struggling, what kind of assistance should be provided?

Researchers have attempted to answer many of these questions using randomized experiments (A/B testing) integrated into online learning platforms [20, 25], but these learning platforms must balance scientific inquiry with social responsibility. If researchers are experimenting with new and potentially beneficial instructional interventions, then the control students who do not receive the beneficial intervention are being treated unfairly. In an attempt to counteract this unfair treatment of students, researchers have proposed using multi-armed bandit algorithms (MABs) to mediate which interventions are given to students [18, 21, 26]. MABs learn over time which interventions are most effective, and transition from assigning interventions uniformly to recommending the most effective interventions.

Using MABs has the potential to remedy the unfair treatment of students, but doing so causes other problems. MABs adjust which interventions they assign based on prior assignments. Therefore, assignments are not independent of each other, which prevents statistical methods such as t-tests or ANOVAs from being used because they require samples to be independent of each other. Some researchers have proposed modifications to MABs that make the data they collect more similar to an experiment [30, 31], but these modifications only help to identify the most effective treatment for students on average.

To personalize students' learning, the algorithm used to assign treatments must be able to learn qualitative interactions between students and interventions. A qualitative interaction exists when different groups of students each benefit from different interventions [19]. Qualitative interactions can exist for individual students and interventions, e.g., Student A benefits most from Intervention 1, or on a student and intervention feature basis, e.g., Students that take longer than average to answer questions benefit more from multiple-choice problems. Researchers are particularly interested in these feature-based qualitative interactions because they can generalize beyond a specific experiment and have a much greater impact on the pedagogy of online learning.

In order to find qualitative interactions while still gaining the advantages of using MABs, contextual MABs (CMABs) can be used. Unlike MABs, which learn the average effectiveness of each intervention, CMABs learn how to estimate the effectiveness of an intervention given information on a student, their learning environment, and the intervention itself. CMABs are capable of personalizing students' experiences, but, like MABs, bias common statistical methods by creating dependence between samples.

UMAP '23, June 26–29, 2023, Limassol, Cyprus

ACM ISBN 978-1-4503-9932-6/23/06.
https://doi.org/10.1145/3565472.3592955

In this paper, we propose Dynamic Linear ϵ-Greedy (DLEG), a novel adaptation of established CMAB methods that allows for students to receive personalized interventions while identifying valid, unbiased, generalizable qualitative interactions between features of students and the interventions available to them. We first demonstrate in simulation the effects of using DLEG compared to the most widely used CMABs. Then, we evaluate DLEGs ability to improve student learning while discovering generalizable qualitative interactions in a three month long empirical study on 3,602 real students during regular instruction within an online learning platform.

In this work, we make the following contributions.

(1) We propose Dynamic Linear ϵ-Greedy (DLEG), a novel contextual multi-armed bandit algorithm (CMAB) designed to balance the needs of students and researchers.
(2) We compared DLEG to the most well established existing CMABs in simulation.
(3) We empirically evaluated DLEG's ability to help students in a large-scale study.
(4) We empirically evaluated DLEG's ability to discover opportunities to personalize students' learning at-scale within this study.

2 BACKGROUND

2.1 Multi-Armed Bandit Algorithms

Multi-Armed bandit algorithms (MABs) are a class of reinforcement learning algorithm [27] in which the algorithm, or agent, is presented with multiple actions it can take. The agent takes one of the possible actions, and is given a numeric reward based on criteria defined by the researcher. The agent learns over time the relationship between the actions it can take and the reward it receives, and uses this knowledge to try and maximize the reward it receives by taking actions it thinks will lead to a high reward [23]. MABs differ from other more complicated reinforcement learning algorithms because they assume that the reward received for an action is independent of the sequence of actions taken.

In previous work, researchers have shown that MABs were able to increase students' learning during randomized experiments performed within an online learning platform, but that MABs added bias and increased the false positive rate of the following experiment analyses [21]. Some researchers have developed methods of bounding the behavior of MABs [30, 31] in order to make them behave more like a randomized experiment. However, this prior work focused on making MABs more interpretable, but not on identifying opportunities to personalize students' learning.

2.1.1 Contextual Multi-Armed Bandit Algorithms. In this work we focus on contextual multi-armed bandit algorithms (CMABs). CMABs expand upon MABs by incorporating information about the agent's environment, or context, into its decision of what action to take. This context allows users' recommendations to be personalized [4] by learning the relationship between users context and the expected reward.

One challenge when designing a CMAB is to choose a model that can accurately identify relationships between features of the context, the actions, and the reward. Some models, like neural networks, can be very powerful but difficult to interpret. A detailed look at various neural-network based CMABs can be found in [22]. Other models, like linear regressions, are easier to interpret but must have non-linear interactions explicitly engineered into the model. Two of the most well known CMABs, LinUCB [11] and Linear Thompson Sampling [1], both use a ridge regression. A major advantage of using a ridge regression is that it can be updated from a stream of data, i.e., these CMABs do not need a complete history of all the contexts, actions, and rewards they have observed to update their models.

Another challenge is to balance learning about the relationships between the context, actions, and reward with taking the actions that the CMAB expects will lead to the highest reward. This balance is often referred to as the exploration-exploitation trade-off [2]. A naive approach to addressing this balance is to take a random action a pre-determined percent of the time, and otherwise take the action with the highest expected reward. This method is called ϵ-greedy, where ϵ is the percent of time a random action is taken, and the greedy action is the action with the highest expected reward. The ϵ-greedy method is not optimal because theoretically, the CMAB will eventually collect enough data to know with certainty which actions will lead to the highest reward at which point it is unnecessary to take any more random actions. Often, the exploration-exploitation trade-off is addressed using a variant of an Upper Confidence Bound (UCB) [10], or Thompson Sampling (TS) [24, 28] algorithm.

Both UCB and TS use the estimated reward for each possible action as well as a measure of the uncertainty of the estimate to determine which action to take. UCB adds to the estimated reward of an action inversely proportional to how many times previously the action was taken, and calls this value the upper confidence bound of that action. UCB then takes the action with the highest upper confidence bound [10]. TS uses the estimated reward and the variance of this estimate for each action to randomly sample from each possible action's prior reward distribution. TS then takes the action corresponding to the highest-valued random sample [24]. Both UCB and TS start by making mostly random decisions, but as the error of their estimates decreases, they converge to selecting the action with the highest estimated reward.

The downside of using UCB or TS is that actions are always taken based on prior observations, which biases the data collected during these algorithms use, making it unsuitable for typical statistical analyses to compare the effects of the actions. For this reason, in this work we modify the ϵ-greedy method such that it behaves similarly to UCB and TS while still collecting some independently sampled data for statistical analysis.

2.2 ASSISTments

In this work, both studies were performed using data from, or within the ASSISTments online learning platform [1]. ASSISTments is an online learning platform with over 100,000 active student users that focuses on middle-school mathematics. In ASSISTments, teachers assign problem sets from open source mathematics curricula. Students then complete the assignments in the ASSISTments Tutor [8]. When students are struggling they can request to view a video relevant to the skills required to solve the problem, or they

[1] https://new.assistments.org/

can request a hint or explanation directly relevant to the specific problem.

2.2.1 Skill-Level Videos. When a student requests a skill-level video, they are shown a YouTube video related to the skills required to solve their problem. In ASSISTments, each problem is tagged with its most relevant Common Core State Standards for Mathematics Skill Code [13], and five videos are available for each skill code. The student will receive the same video for a specific problem even if they press the button multiple times, but can receive different videos on other problems of the same skill.

2.2.2 Problem-Level Support. Between two and four problem-level supports are available for most of the mathematics question in ASSISTments [16] in the form of sets of hints or explanations. Sets of hints are composed of multiple small pieces of advice that the student must request one at a time and do not reveal the answer. Explanations contain a complete solution to the problem and the correct answer. Based on what is available, the student can request hints or an explanation, but never both for the same problem. Sets of hints and explanations will impact a student's score when they are requested, but hints remove a fraction of a student's score for each hint requested, and explanations remove all of a student's score upon request [16].

2.2.3 The Automatic Personalized Learning Service. The ASSISTments platform has developed the Automatic Personalized Learning Service (APLS) in order to use MABs to recommend both skill-level videos and problem-level supports to students [18]. The APLS operates in real-time by responding to requests from the ASSISTments Tutor. In these requests, the tutor provides the APLS with unique identifiers for the student, the problem, and the available content. The APLS uses these identifiers to look up features of the student, problem, and content, compiles these features into context, and then uses a recommendation algorithm to select content for the student. The APLS randomly chooses from multiple recommendation algorithms each time it makes a recommendation, which enables randomized experiments between algorithms [18]. In this work, we used the APLS to compare random recommendations to recommendations made by Dynamic Linear ϵ-Greedy.

In the APLS, each recommendation algorithm receives a reward of 1 when the student gets the next problem correct without any additional support after viewing the algorithm's recommended content, and 0 when they do not. When no information on the student's next-problem correctness is available, the recommendation is not used to update the algorithm. The APLS calculates these rewards every day in the evening during low load periods in order to not interrupt users' experience. After updating each algorithm with the rewards it received for each recommendation it made since the last update, the APLS uses logs of students' actions within the ASSISTments Tutor to update the features of the students, problems, and content. A complete list and descriptions of all the context calculated by the APLS can be found at https://osf.io/zuwf7. The subset of this context used during the empirical study in this work is discussed later in Section 4.1.

3 DYNAMIC LINEAR EPSILON-GREEDY

This work presents the Dynamic Linear ϵ-Greedy (DLEG) algorithm, shown in Algorithm 1. The source code for DLEG can be found at https://osf.io/q298c. DLEG is a contextual multi-armed bandit algorithm that addresses the exploration-exploitation trade-off in a way that enables statistically reliable, generalizable insight to be gleaned from the data collected during its use. DLEG uses a modification of the ϵ-greedy method, because the data collected from random decisions is akin to data collected during a randomized experiment, and is thus unbiased, and available for use in common statistical analyses.

DLEG estimates the reward from context using a ridge regression, similarly to other linear CMABs [1, 11]. After a short period of random recommendations used to give the regression initial data to fit on, with probability ϵ, DLEG will randomly select from the possible actions it can take, observe a reward, and then update the ridge regression with this sample. After updating the ridge regression, the regression is used to estimate the reward of the random recommendation that was just made. The error in this estimate is used to track the mean squared error of the model's reward estimates for its random recommendations, mse_r.

After a short period of random recommendations used to give the regression initial data to fit on, with probability $1-\epsilon$, DLEG will use the ridge regression to estimate the reward for each possible action, and then take the action with the highest estimated reward, i.e., the greedy action. DLEG observes the reward for this greedy recommendation, but *does not* update the ridge regression after a greedy recommendation. The error of the greedy recommendation's reward estimate is used to track the mean squared error of the model's reward estimates for its greedy recommendations, mse_g.

The data collected from DLEG's random recommendations are independent of each other, and therefore can be used to analyze the qualitative interactions in the data without inducing any bias from dependence between samples. However, if ϵ never changes, then once the ridge regression has learned all it can from the data, DLEG will be wasting opportunities to exploit these qualitative interactions by continuing to make random recommendations. To avoid this, ϵ is updated dynamically on Line 31 of Algorithm 1 based on mse_r and mse_g, as long as a small amount of data exists for the calculation of mse_r and mse_g. If these two mean squared errors are equal, it means that the regression is just as good at estimating the reward given context it was not trained on as it is given context it was trained on, which implies that the model has captured the underlying trends in the data. If this is the case, then the model will stop making random recommendations. On the other hand, the worse the model is at estimating the reward given context it was not trained on compared to context it was trained on, the higher ϵ will be, resulting in more random recommendations. This allows the model to improve its predictive accuracy by collecting more training data. This method is also robust to changes in the relationship between context and reward, because if the accuracy of the reward estimates for greedy recommendations was very high, but started getting worse, DLEG would begin to make more random recommendations and continue to fit the regression. This simple trick of adjusting ϵ based on the ratio of the standard errors allows this variant of the ϵ-greedy method to be competitive with more

optimal methods, while allowing for unbiased statistical analysis on the random recommendations.

In Algorithm 1:

- λ is the L2 penalty of the ridge regression, used during the initialization of the regression.
- α is the number of random recommendations that must be made first before DLEG can begin to make greedy recommendations.
- ϵ is the probability that the model will make a random recommendation after α random recommendations.
- n_r and n_g track the number of random and greedy recommendations made by DLEG respectively.
- mse_r and mse_g track the mean squared error of the ridge regression's reward predictions for random and greedy recommendations respectively.
- $\mathbf{A}$ and $\mathbf{b}$ are the $X^\top X + \lambda I$ and $X^\top Y$ components of the ordinary least squares solution for ridge regressions: $\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top Y$ [29]. $\mathbf{A}$ and $\mathbf{b}$ can be updated iteratively as more samples are collected.
- $\hat{\boldsymbol{\beta}}$ is the vector of coefficients of the ridge regression, which can be calculated each time a prediction needs to be made from $\mathbf{A}$ and $\mathbf{b}$.
- $\mathcal{U}(0, 1)$ is a random value sampled from a uniform distribution in the range $[0, 1)$.
- $\mathbf{x}_{t,s,a}$ is all the context for state s and action a at time t.
- $p_{t,s,a}$ is the predicted reward received from taking action a in state s at time t.

Algorithm 1 Dynamic Linear ϵ-Greedy

Inputs: $\lambda \in \mathbb{R}_+$, $\alpha \in \mathbb{N}$, $\epsilon = 0.5$, $n_r = 0$, $n_g = 0$, $mse_r = 0$, $mse_g = 0$
$\mathbf{A} \leftarrow \lambda \mathbf{I}_d$ (d x d dimensional diagonal matrix where all values on the principle diagonal are λ)
$\mathbf{b} \leftarrow \mathbf{0}_{d \text{ x } 1}$ (d x 1 dimensional zero matrix)
$\hat{\boldsymbol{\beta}} \leftarrow \mathbf{A}^{-1}\mathbf{b}$
for t = 1, 2, 3, ..., T **do**
  $R \leftarrow \mathcal{U}(0, 1)$
  Observe features of state s and all actions $a \in A_t$: $\mathbf{x}_{t,s,a} \in \mathbb{R}^{1 \text{ x } d}$.
  for all $a \in A_t$ **do**
    if $R \leq \epsilon$ **or** $n_r < \alpha$ **then**
      $p_{t,s,a} \leftarrow \mathcal{U}(0, 1)$
    else
      $p_{t,s,a} \leftarrow \mathbf{x}_{t,s,a}\hat{\boldsymbol{\beta}}$
    end if
  end for
  Choose arm $a_t = \arg\max_{a \in A_t} p_{t,s,a}$ with ties broken arbitrarily.
  Observe reward $r_t \in \mathbb{R}$.
  if $R \leq \epsilon$ **or** $n_r < \alpha$ **then**
    $\mathbf{A} \leftarrow \mathbf{A} + \mathbf{x}_{t,s,a_t}^\top \mathbf{x}_{t,s,a_t}$
    $\mathbf{b} \leftarrow \mathbf{b} + \mathbf{x}_{t,s,a_t}^\top r_t$
    $\hat{\boldsymbol{\beta}} \leftarrow \mathbf{A}^{-1}\mathbf{b}$
  end if
  $e \leftarrow \mathbf{x}_{t,s,a_t}\hat{\boldsymbol{\beta}} - r_t$
  if $R \leq \epsilon$ **or** $n_r < \alpha$ **then**
    $n_r \leftarrow n_r + 1$
    $mse_r \leftarrow mse_r + \frac{e^2 - mse_r}{n_r}$
  else
    $n_g \leftarrow n_g + 1$
    $mse_g \leftarrow mse_g + \frac{e^2 - mse_g}{n_g}$
  end if
  if $n_r \geq \alpha$ **and** $n_g \geq \alpha$ **then**
    $\epsilon \leftarrow 1 - \sqrt{\frac{mse_r}{mse_g}}$
  end if
end for

3.1 Design Constraints

For DLEG to operate at-scale within the ASSISTments APLS, its model was required to 1) have a limited, fixed memory cost, i.e., DLEG could not grow in size over time, nor could it be too big to begin with, and 2) be able to train from one sample at a time, i.e., not require the entire history of recommendations to fit the model. Some CMABs like LinUCB [11] can be trained from one sample at a time, but fit one model for each action the CMAB can take. Within ASSISTments, new content is constantly being added to the system. If DLEG created an additional model each time new content was added, the system would quickly run out of memory. Additionally, separate models for different actions prevents the insight learned about the effectiveness of an action from being transferable to other actions.

Some CMABs use more complicated models like random forests [7] or deep neural networks [22] to learn the relationship between context and reward, but these models not only take up a large memory cost due to their structure, but they must also be re-fit using previous data as new data is collected, making these methods unsuitable for use within the APLS.

In order for DLEG to fit within the imposed constraints, a single ridge regression predicting reward using the context of the students, problems, and available content as input was used as DLEG's model. The ridge regression in DLEG is very similar to the model used in LinUCB [11], but instead of fitting a separate regression for each action, one regression that includes context of the actions was fit. This single regression allows DLEG to identify transferable insight into opportunities to personalize content provided to students based on features of the students, problems, and content in ASSISTments.

4 METHODOLOGY

4.1 Feature Selection

4.1.1 Simulation Study. Before conducting an empirical study of DLEG using the ASSISTments Automatic Personalized Learning Service (APLS), a simulation study was done comparing DLEG to similar variants of existing CMABs. The simulation study was performed using the ASSISTments Student Support Dataset (SSD) [18]. This dataset contains samples from thousands of experiments in which students were randomized between different problem-level supports. The features used from the SSD were chosen to be as similar as possible to the features chosen for the empirical study. For

students, the user_avg_correctness, user_avg_support_requested, and user_med_ln_first_response_time features were used. While these features are not calculated identically to the features in the APLS, they attempt to measure the same thing. The difference being that the features in the APLS are normalized versions of the features included in the SSD. For problems, the problem_avg_correctness, problem_avg_support_requested, problem_med_ln_first_response_time, problem_type_1, problem_subject_g, problem_subject_rp, problem_subject_ns, problem_subject_ee, problem_subject_f, and problem_subject_sp features were used. The problem_type_1 feature in the SSD is similar to the problem_type_choice feature in the APLS, which is an indication of whether the question is of any type that requires the user to choose from options, as opposed to problem_type_1, which is an indication of whether or not the question is a multiple-choice question. For the problem-level supports, the student_support_is_explanation, student_support_message_count, student_support_contains_image, and student_support_contains_video features were included. The student_support_is_explanation feature in the SSD is equivalent to the answer_given feature in the APLS. The SSD provides the next problem correctness for each sample, which the APLS uses as the CMAB reward, Therefore, the simulation also used this as the reward. A complete description of the features in the SSD is available through [18]. In total, 1 constant, i.e., the intercept, 17 features, and 52 interactions between features of the supports and features of the users and problems were included in DLEG's regression for the problem-level support simulation study.

4.1.2 Empirical Study. Prior to this work, no CMABs had been evaluated using ASSISTments' APLS. Prior research has shown the negative impact that including too many features in a CMAB has on the CMAB's ability to benefit users [12]. Therefore, for the study in this work, the CMAB used a smaller subset of the features available in the APLS, as well as the interactions between the features of the content and features of the student and problem. The interactions between features was a necessary inclusion because without interactions, the ridge regression used by DLEG to estimate reward would not be able to find opportunities for personalization. For students, the correctness, support_requested, and ln_first_response_time features were chosen. For problems, the correctness, support_requested, ln_first_response_time, type_choice, subject_g, subject_rp, subject_ns, subject_ee, subject_f, and subject_sp features were chosen. For the skill-level videos, only percent_likes, percent_dislikes, and percent_comments were included in the context provided to DLEG. The definitions for all the above features can be fount at https://osf.io/zuwf7/. In total, 1 constant, i.e., the intercept, 16 features, and 39 interactions between features were included in DLEG's regression for the skill-level video empirical study.

4.2 Study Design

4.2.1 Simulation Study Design. The simulation study was conducted identically to previous simulation studies done using medical and educational data from randomized studies [18, 21]. To simulate how effectively CMABs would have recommended support to students in the SSD, samples from the SSD were randomly selected with replacement using the following strategy [18].

(1) Initialize a CMAB.
(2) Randomly sample with replacement a single instance of a student receiving support from the SSD.
(3) Provide context from the sample to the CMAB algorithm for all possible supports the student could have received.
(4) Given this context, receive a support recommendation from the CMAB.
(5) If the support recommended by the CMAB matches the support that was actually given to the student, update the bandit algorithm using the next problem correctness value in the SSD, otherwise ignore the recommendation and go back to step 2.
(6) Repeat steps 2-5 to simulate the CMAB making a series of recommendations.

This study ran for 1,000,000 recommendations to observe the long-term effects of the different algorithms. In the simulation study, DLEG was compared to random selection, Linear Thomson Sampling [1], and Pooled-LinUCB, which is similar to LinUCB [11] but with only one regression that shares context across actions. These CMABs were selected for comparison because they are well established algorithms that meet the memory and time requirements of the ASSISTments APLS.

4.2.2 Empirical Study Design. Once the simulation study demonstrated the effectiveness of DLEG compared to existing CMAB algorithms (results discussed in Section 5.1) the next step was to evaluate DLEG in a real setting, at-scale, within an online learning platform. Both a random selection model and a DLEG model were created in the APLS for recommending skill-level videos. Each time a student requested a video, the student's request was randomly sent to either the random model or DLEG with equal probability. The random model randomly recommended one of the available videos with equal probability, and DLEG recommended a video using Algorithm 1. Essentially, this study is a randomized experiment between two conditions (Random vs. CMAB recommendations), and the random selection model performed a randomized experiment between the different videos. Only one model was compared to DLEG in order to collect as much data as possible for each model's analysis. The random model was chosen because it provided a control to measure both DLEG's performance and interpretability.

4.3 Study Analysis

4.3.1 Recommendation Algorithm Comparison. To compare the different recommendation algorithms to each other in both the simulation study and the empirical study, a logistic regression was used to predict the reward given the following inputs:

(1) A constant.
(2) Three covariates: student, problem, and next-problem prior correctness.
(3) A binary feature for each model except random selection indicating if that model made this recommendation.
(4) The number of recommendations made thus far by the algorithm that made this recommendation.
(5) A feature for the interactions between each of Input 3's features and Input 4.

If any of Input 5's features were positive and statistically significant, then the corresponding algorithm out-performed random selection, because over time, the chance of receiving a high reward increased for that algorithm more than it did for random selection. Additionally, if any of Input 5's features were statistically significantly different from each other, then one non-random model out performed another. This analysis was used instead of just comparing the distribution of reward between the algorithms because the distribution of reward is not expected to be different at the beginning of the algorithms' use, when mostly random recommendations are being made. However, once the non-random models have learned something, the reward distributions should be different.

4.3.2 Identifying Effective Content. To determine if DLEG was capable of identifying any significant relationships between features of the videos and students' performance at-scale, a logistic regression was fit to estimate students' next-problem correctness using all the video features available in the APLS as well as covariates for student, problem, and next-problem prior correctness. To ensure there was no bias in the estimates due to dependence between samples, only the data from DLEG's random recommendations during the empirical study was used to fit the model. This model was also fit using data from the random selection model used during the study to see how much difference there was between what DLEG's random recommendations revealed and what a randomized experiment revealed. The p-values of the models' coefficients were corrected for multiple hypothesis testing using the Benjamini-Hochberg procedure [3].

It is important to note that a lack of bias from dependent samples does not mean that the results of this regression can be interpreted as causal relationships. To identify causal relationships in the data, all but one feature of the content provided to students would have needed to be controlled [9]. However, the skill-level videos came from publicly available YouTube videos. No efforts were made to control for different features across videos, nor to make sure each skill had a similar distribution of features in the videos available for it. As such, the coefficients of this regression can only be interpreted as correlations in the data. However, there is nothing preventing the use of DLEG in a causal setting, as long as the content is appropriate for causal inference.

4.3.3 Identifying Qualitative Interactions. The greatest value of DLEG is in its ability to identify opportunities to personalize students' learning. For these opportunities to exist, qualitative interactions must be present in the data. Using the data collected from DLEG's random recommendations, the same method used in [18] to identify statistically significant qualitative interactions between users and the content available to them was used. In order to identify generalizable interactions, students were binned into high and low knowledge groups based on whether or not they had a higher than average correctness feature in the APLS. Each video feature was also binned into above and below average groups. The regression $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_1 \oplus x_2)$ was then fit, where x_1 is a binary variable for a binned video feature, x_2 is a binary variable for a student's binned prior correctness, and y is the student's next problem correctness. Using this model, a qualitative interaction exists if β_3^2 is greater than β_1^2, which is derived with more detail in [18]. p-values for the statistical significance of these qualitative interactions were calculated using a bootstrapping approach in which a regression for each video feature was fit 10,000 times on subsets of equal size to the original data sampled from the original data with replacement. The distribution of $\beta_3^2 - \beta_1^2$ was used to perform a one-sample t-test to determine the p-value of the null hypothesis: $\beta_3^2 - \beta_1^2 \leq 0$. p-values were corrected for multiple hypothesis testing using the Benjamini-Hochberg procedure [3].

5 RESULTS

5.1 Simulation Study

Figure 1 shows the cumulative reward received by the three CMABs compared to random selection during the simulation. In Figure 1, the total reward received through random selection was subtracted from the total reward received by each algorithm after the same number of recommendations were made. The random selection line is a horizontal line at $y = 0$ because the cumulative reward received through random selection was subtracted from itself. By comparing each CMAB to random selection, we can see more clearly how each CMAB compares to selecting at random from the available content.

The regression described in Section 4.3.1 found that DLEG and Linear Thompson Sampling statistically significantly out-performed random selection ($p < 0.001$ and $p = 0.006$ after correction respectively), but Pooled-LinUCB did not. Although Figure 1 indicates that DLEG and Pooled-LinUCB are the best, this is not the case after adjusting for prior-knowledge covariates. Additionally, DLEG statistically significantly out-performed Pooled-LinUCB ($p = 0.012$ after correction). Based on the simulation, we could expect DLEG to perform better than random selection and at least as well as existing CMABs, while enabling further statistical analysis of the data.

5.2 Empirical DLEG Performance Analysis

From October 3rd 2022 to December 30th 2022, 3,602 students participated in the skill-level video empirical study. Each time a student requested a video, they were randomized at a problem-level between receiving a randomly selected video, chosen from 5 skill-related videos, or receiving the video recommended by DLEG from the same set of 5 videos. 6,035 total recommendations were made, 2,982 of them made by DLEG, and 3,053 of them made by random selection. 817 videos were shown to students across 217 skills. On average, when DLEG was used to make recommendations, about 2.8 different videos were shown per skill, and each video was viewed an average of 5.5 times. When random selection was used to make a recommendation, about 3.3 different videos were shown per skill, and each video was viewed around 4.5 times.

Figure 2 shows the trends in the recommendations made by DLEG. As shown by left graph, DLEG made fewer random recommendations over time, which indicates that it was able to learn the relationship between context and reward. The right graph shows that after an initial learning period, DLEG began to consistently out-perform random selection. Using the regression described in Section 4.3.1, no statistically significant differences between DLEG and random selection were found. With a longer study, it is likely that DLEG's video recommendations would have a statistically

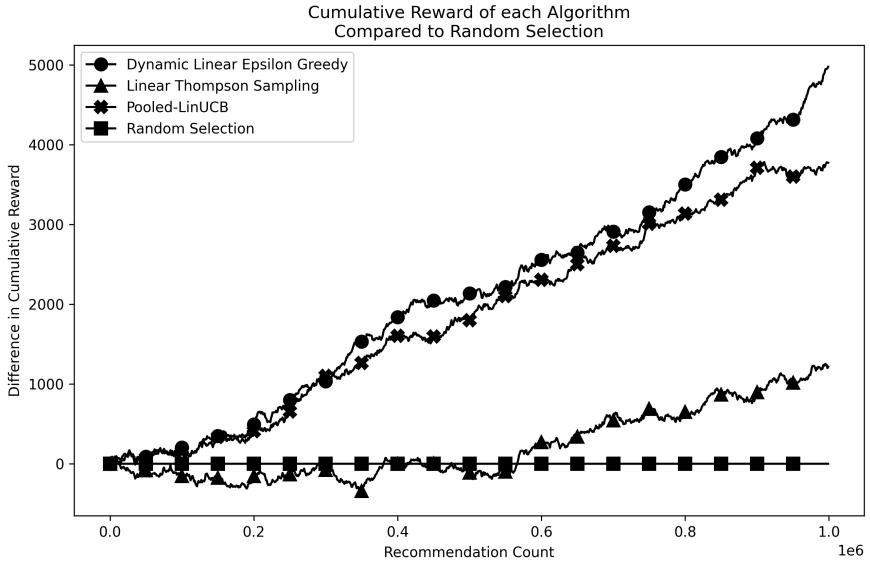

Figure 1: The cumulative reward of each algorithm as a function of how many recommendations they have made compared to the cumulative reward received through random selection.

significant positive effect on students' propensity to get the next problem correct.

Figure 3 shows how the state of DLEG changed as more recommendations were made during the study. The left graph shows how at the beginning of the study, the standard error of DLEG's predictions of the reward for the random recommendations was very low, this was because there were few recommendations made, and the random recommendation data was used to train the regression, which caused DLEG's regression to over-fit on the random recommendation data. Due to this over-fitting, the standard error of DLEG's predictions of the reward for the greedy recommendations was very high. This resulted in a high initial ϵ. This is ideal because a CMAB should explore more at the beginning of its use in order to learn the trends in the data.

As DLEG made more recommendations, the standard error of the random recommendation reward predictions climbed and the standard error of the greedy recommendation reward predictions fell. This is an indication that DLEG's regression was trending away from over-fitting. As a result of these shifts in standard error, ϵ decreased. This is preferred because as a CMAB learns more about the relationship between context and reward, it should explore less and make more exploitative choices. At the end of the study, DLEG was making random recommendations about 7% of the time. One can observe that a sudden drop in reward around recommendation 3,000 caused ϵ to slightly increase. This is desired because as trends in the data change, DLEG should explore more to learn about these new trends.

5.3 Empirically Identifying Effective Content

The purpose of using DLEG was not only to positively impact students' learning, but to also reveal statistically reliable relationships between features of the context and reward. Using the methodology discussed in Section 4.3.2, two logistic regressions were fit. One using DLEG's random recommendations and the other using the random selection algorithm's recommendations. The confidence intervals of the coefficients of the logistic regression fit using data from DLEG were about 43% larger on average than the confidence intervals of the coefficients of the logistic regressions fit using data from the random selection algorithm. The difference in confidence intervals is likely due to DLEG only making random recommendations about 20% of the time. However, even though the confidence intervals were larger, neither regression had any statistically significant coefficients, meaning that the lack of data did not result in DLEG missing any significant correlations.

5.4 Empirically Identifying Qualitative Interactions

Even though there were no features of the videos that were statistically significantly predictive of student performance, there could still be opportunities for personalization. The coefficients in the previous regressions only indicated how predictive each feature was of student performance on average, but it could be that higher knowledge students benefited from different things than lower knowledge students. To investigate for these qualitative interactions, the approach discussed in Section 4.3.3 was used to determine if there was a qualitative interaction between each feature of the videos and students' prior knowledge. The results of this analysis revealed 5 statistically significant qualitative interactions, shown in Table 1, all of which had p-values < 0.001 after correcting for multiple hypotheses.

These results indicate that despite little evidence that features of the content were predictive of students' average performance, many features had qualitative interactions with students' prior knowledge. These interactions can be used to personalize the videos provided

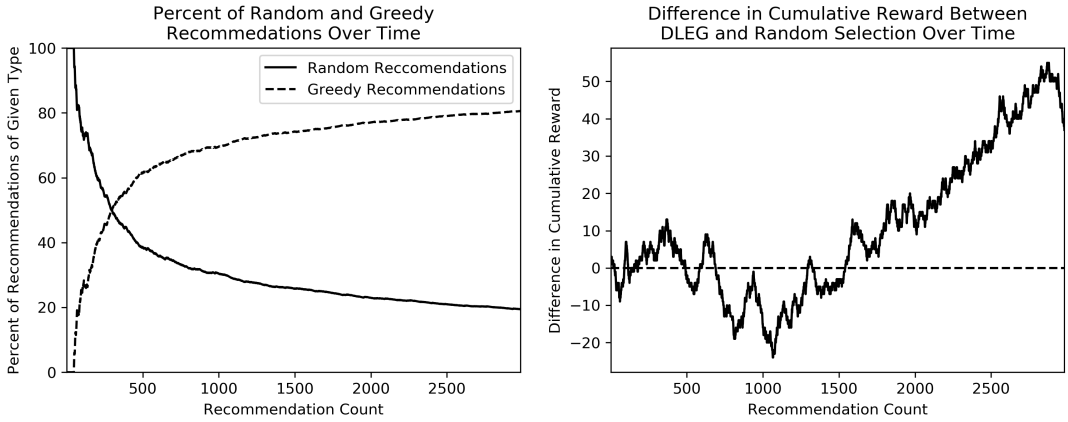

Figure 2: The total percent of random and greedy recommendations made by DLEG (left) and the cumulative reward received by DLEG compared to random selection (left) for the skill-level video empirical study.

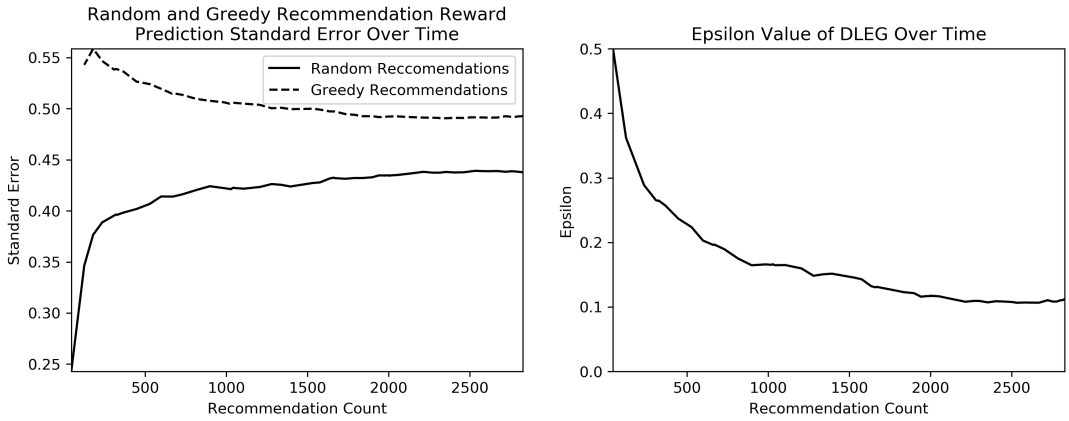

Figure 3: The standard error of the predictions made for random and greedy recommendations (left) and the resulting ϵ (right) calculated by DLEG for the skill-level video study.

Table 1: Some Typical Commands

Feature	High Knowledge Benefits From	Lower Knowledge Benefits From
Percent Likes	Above Average	Below Average
Length	Above Average	Below Average
Face Included	Yes	No
Reading Tone	Below Average	Above Average
Male Tone	Below Average	Above Average

to students to help each student achieve their maximum potential. This analysis is possible because of the independent random recommendations that DLEG made during the study, without which the coefficients and confidence intervals of these analyses would be biased.

6 DISCUSSION

Using DLEG to recommend content to students had promising results. DLEG performed slightly, but significantly better than random selection and another CMAB when recommending problem-level supports in simulation, and also slightly out-performed random selection at-scale within ASSISTments, though not significantly.

Overall it seems the CMABs explored in this work struggled to have a large benefit on students' learning. Most likely, this lack of significant improvement was caused by the constraints placed upon the algorithms. DLEG, Pooled-LinUCB, and Linear Thompson Sampling all used a single ridge regression to model the relationship between context and reward. Many models used in learning sciences to understand students' performance do not reduce all content to a set of features, and instead model students or problems as individuals [5, 6, 17]. Additionally, the relationship between context and reward changes over time. In the skill-level video study, after an initial learning period, DLEG appeared to steadily outperform random selection, but near the end of the study, DLEG's

performance dropped. Around this time, students were preparing for winter break, and may have felt rushed to finish their work. This could have changed what kind of videos were most effective. Perhaps longer videos, which were previously more informative, were now ignored because students were unwilling to spend time watching them. This is just one hypothetical example of a change in students' preferences over time, but any number of factors could have led to this shift. DLEG will eventually re-learn trends, but if the trends in the data are often shifting, temporal features should be included in the model so a CMAB can learn to anticipate these trends. Lastly, it could be that DLEG had a difficult time significantly out-performing random selection because all the content in ASSISTments was equally good. Even if there were slight differences in quality, all the content was written or validated by mathematics teachers. In domains where there are fewer consequences for low-quality material, DLEG would likely have a larger benefit. However, in education, there is a significant negative impact when students are shown low-quality material. Therefore, all the content DLEG could recommend was likely similarly high-quality.

Although DLEG had only a small benefit to students, its purpose was not solely to benefit students, but to also glean statistically reliable and unbiased insight into the relationship between the context and the reward. Although there were no features of educational videos that were significantly predictive of students' average performance, multiple qualitative interactions between students' prior knowledge and features of the videos were significantly predictive of students' performance. Although these are only correlations, we can look at the interactions, theorize why they occurred, and see if there are causal studies to support our theories. For example, this work found that higher knowledge students benefited more from videos that were above average in length. Studies have shown that students' attention span is a key factor in their academic success [14]. Therefore, it could be that students' attention spans help them to both achieve more academically and watch longer videos.

6.1 Limitations and Future Work

While the results of this work are promising, there are some limitations to the scope of our analysis. Currently, DLEG has only been tested on data from the ASSISTments platform. While this has provided the opportunity to evaluate the effectiveness of DLEG at scale in a real-world environment, it also put strong restrictions on the memory and time requirements of DLEG. A version of DLEG more akin to LinUCB, where each action has a separate regression, could be even more powerful while still allowing for some statistically reliable insight. In the future, exploring how DLEG performs in other domains could both reveal interesting insight into the relationship between the context and reward of those new domains, as well as provide further opportunities to refine DLEG.

In addition to the limited scope, the empirical study ran for only about three months, and DLEG was only able to make 2,982 recommendations. While this may seem like a lot, many CMAB studies allow the algorithm to make millions of recommendations before interpreting the results. The limited time available for this study likely impacted the discovery of more significant results. Longer versions of this study should be repeated to gather more data and evaluate if the results are consistent.

7 CONCLUSION

In this work, we introduced DLEG, a CMAB algorithm that enables personalized content recommendations by learning and leveraging the statistical relationships between context and reward. We demonstrated through simulation and empirical studies that DLEG can slightly improve student performance within an online learning platform. Additionally, we found that unbiased random samples from DLEG's recommendations can reveal interesting qualitative interactions between features of educational content and students' prior knowledge. These results have implications for both DLEG's ability to enhance student performance and for researchers seeking to design further studies or build upon existing pedagogy. In any domain where reliable, generalizable insights from recommendations are desired, DLEG can be employed to identify opportunities for personalization that benefit both researchers and recipients.

ACKNOWLEDGMENTS

We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), NHI (R44GM146483), and Schmidt Futures. None of the opinions expressed here are that of the funders.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*. PMLR, 127–135.
[2] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410, 19 (2009), 1876–1902.
[3] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.
[4] Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. 2020. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.
[5] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.
[6] Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.
[7] Raphaël Féraud, Robin Allesiardo, Tanguy Urvoy, and Fabrice Clérot. 2016. Random forest for the contextual bandit problem. In *Artificial intelligence and statistics*. PMLR, 93–101.
[8] Neil T Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.
[9] Guido W Imbens and Donald B Rubin. 2010. Rubin causal model. In *Microeconometrics*. Springer, 229–241.
[10] Tze Leung Lai, Herbert Robbins, et al. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.
[11] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.
[12] ZhaoBin Li, Luna Yee, Nathaniel Sauerberg, Irene Sakson, Joseph Jay Williams, and Anna N Rafferty. 2020. Getting Too Personal (ized): The Importance of Feature Choice in Online Adaptive Algorithms. *International Educational Data Mining Society* (2020).
[13] William McCallum. 2015. The common core state standards in mathematics. In *Selected regular lectures from the 12th international congress on mathematical education*. Springer, 547–560.
[14] Megan M McClelland, Alan C Acock, Andrea Piccinin, Sally Ann Rhea, and Michael C Stallings. 2013. Relations between preschool attention span-persistence and age 25 educational outcomes. *Early childhood research quarterly* 28, 2 (2013),

314–324.
[15] Kyndra V Middleton. 2020. The Longer-Term Impact of COVID-19 on K–12 Student Learning and Assessment. *Educational Measurement: Issues and Practice* (2020).
[16] Thanaporn Patikorn and Neil T Heffernan. 2020. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 115–124.
[17] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. Performance Factors Analysis–A New Alternative to Knowledge Tracing. *Online Submission* (2009).
[18] Ethan Prihar, Aaron Haim, Adam Sales, and Neil Heffernan. 2022. Automatic Interpretable Personalized Learning. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*. 1–11.
[19] Ethan Prihar, Thanaporn Patikorn, Anthony Botelho, Adam Sales, and Neil Heffernan. 2021. Toward Personalizing Students' Education with Crowdsourced Tutoring. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*. 37–45.
[20] Ethan Prihar, Manaal Syed, Korinn Ostrow, Stacy Shaw, Adam Sales, and Neil Heffernan. 2022. Exploring Common Trends in Online Educational Experiments. In *Proceedings of the 15th International Conference on Educational Data Mining*. 27.
[21] Anna Rafferty, Huiji Ying, Joseph Williams, et al. 2019. Statistical consequences of using multi-armed bandits to conduct adaptive educational experiments. *Journal of Educational Data Mining* 11, 1 (2019), 47–79.
[22] Carlos Riquelme, George Tucker, and Jasper Snoek. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127* (2018).
[23] Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58, 5 (1952), 527–535.
[24] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning* 11, 1 (2018), 1–96.
[25] Alexander O Savi, Joseph Jay Williams, Gunter KJ Maris, and Han van der Maas. 2017. The role of A/B tests in the study of large-scale online learning. (2017).
[26] Adish Singla, Anna N Rafferty, Goran Radanovic, and Neil T Heffernan. 2021. Reinforcement learning for education: Opportunities and challenges. *arXiv preprint arXiv:2107.08828* (2021).
[27] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
[28] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3-4 (1933), 285–294.
[29] Wessel N van Wieringen. 2015. Lecture notes on ridge regression. *arXiv preprint arXiv:1509.09169* (2015).
[30] Jiayu Yao, Emma Brunskill, Weiwei Pan, Susan Murphy, and Finale Doshi-Velez. 2021. Power Constrained Bandits. In *Machine Learning for Healthcare Conference*. PMLR, 209–259.
[31] Yang Zhi-Han, Shiyue Zhang, and Anna Rafferty. 2022. Adversarial bandits for drawing generalizable conclusions in non-adversarial experiments: an empirical study. In *Proceedings of the 15th International Conference on Educational Data Mining*. 353.