

Time Slice Imputation for Personalized Goal-based Recommendation in Higher Education

Weijie Jiang

jiangwj.14@sem.tsinghua.edu.cn

Tsinghua University & University of California, Berkeley

Zachary A. Pardos

zp@berkeley.edu

University of California, Berkeley

ABSTRACT

Learners are often faced with the following scenario: given a goal for the future, and what they have learned in the past, what should they do now to best achieve their goal? We build on work utilizing deep learning to make inferences about how past actions correspond to future outcomes and enhance this work with a novel application of backpropagation to learn per-user optimized next actions. We apply this technique to two datasets, one from a university setting in which courses can be recommended towards preparation for a target course, and one from a massive open online course (MOOC) in which course pages can be recommended towards quiz preparation. In both cases, our algorithm is applied to recommend actions the learner can take to maximize a desired future achievement objective, given their past actions and performance.

CCS CONCEPTS

• **Applied computing** → **Education**; • **Information systems** → **Recommender systems**.

KEYWORDS

goal; university; enrollment; online courses; deep learning

ACM Reference Format:

Weijie Jiang and Zachary A. Pardos. 2019. Time Slice Imputation for Personalized Goal-based Recommendation in Higher Education. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19)*, September 16–20, 2019, Copenhagen, Denmark. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3298689.3347030>

1 INTRODUCTION

The common form of personalized adaptation in education technologies, such as Intelligent Tutoring Systems, has been to give a variable amount of practice problems to a student based on the system’s estimate of their cognitive mastery [4]. When a learning system contains non problem-based learning materials, such as videos, book pages, and discussion posts (e.g., in online courses), or where complex prerequisite structures are expected (e.g., among college courses), a broader view of adaptivity is desirable whereby course content or courses themselves can be recommended to the learner. Collaborative approaches can give normative suggestions

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6243-6/19/09...\$15.00

<https://doi.org/10.1145/3298689.3347030>

[2], but not all learners with the same data sequences want to achieve the same goal, and often their goals change. Therefore, education is a domain in which personalized recommendation, based on an individual’s goal, is important.

We build on previous research utilizing recurrent neural networks (RNN) to infer prerequisite relationships in middle and high school math topics [6] and on prior work adapting RNNs to the task of college course preparation recommendation [3]. In this paper, we (a) improve the tractability of generating a recommendation from the architecture introduced in [3] and (b) apply this improvement to the task of course preparation recommendation, as it was applied to in that prior work, and extend its application to MOOC quiz preparation, where it has not yet been applied (Fig. 1). Our improvement in tractability is achieved by treating the space of recommendations (e.g., courses or course pages) as weights in a neural network that can be learned by SGD on a per-learner basis to optimize for a future goal (e.g., grade on a course or quiz). Prior work [3, 6] approached this optimization by enumerating every possible recommendation and evaluating it as a potential input to the model.

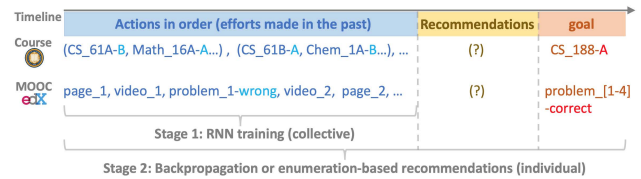


Figure 1: Example sequences used for time slice imputation

2 THE GOAL-BASED MODEL FRAMEWORK

We illustrate the generalized goal-based recommendation framework applied to higher education contexts in Figure 2. The first step conducts the model training from collective sequences (a_1, a_2, \dots, a_t) of all learners to learn an embedding of courses concatenated with grade, in the college context, or course pages and past quiz performance, in the MOOC context, with the objective function of predicting their future grade on any arbitrary next course or MOOC quiz a_t . All the parameters represented by the black arrows in Figure 2 are learned first (collective). Then, these parameters are fixed, and an individual learner’s historical actions are presented to the model to get their hidden state h_{t-1} . Finally, we employ backpropagation (or enumeration) to surface recommendations a_t at time slice t for the learner’s individual goal a'_{t+1} . Domain-based filters are applied to candidate recommendations in order to mitigate the chance of spurious inferences making their way to the learner.

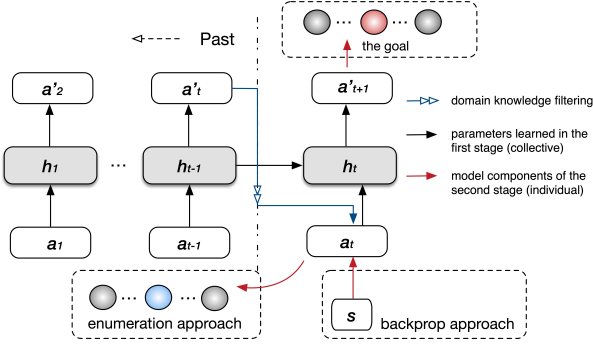


Figure 2: Goal-based recommendation architecture

3 APPLICATION TO COURSE PREPARATION

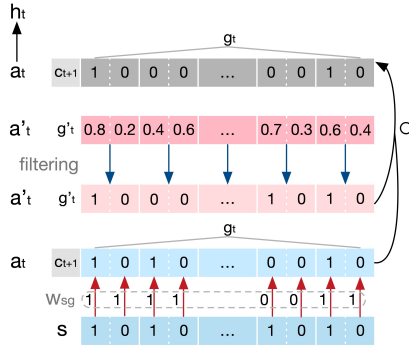


Figure 3: Input topology for individual backprop-based college course preparation

In the college or university setting, we consider the goal of achieving an above threshold grade on an individually specified target course. The task of the recommender is to suggest a course that is expected to increase the student's preparation for the target course, given their course (and grade) history. This suggested course must also be one that the model predicts the student can pass. We inherit the course grade prediction model from [3] (*Model 2*), which was the best performing of the evaluated architectures¹. In this context, $a_t = (c_{t+1}, g_t)$, where g_t is set to be a multi-hot encoding to represent the combination of courses enrolled in and the grades received for those courses out of the total n courses in semester t , i.e., $g_t = (g_t^1, g_t^2, \dots, g_t^n)$, and g_t^i is set to be a one-hot encoding to represent possible grade categories for course i in semester t , $g_t^i = (s_i^{\geq B}, s_i^{< B})$. Here, we define two categories using a grade achievement threshold of B , which divides students into performance groups we will refer to as *high performance* and *lower performance*. The combination of courses students take can affect their outcomes in those courses [1, 7]. To capture this composition effect, we specify an additional input of c_t , set to be a multi-hot encoding of multiple courses co-enrolled in semester t , $c_t = (c_t^1, c_t^2, \dots, c_t^n)$. Therefore, the input is a concatenation of c_{t+1}

¹The LSTM variant of an RNN was used in all model architectures.

with g_t , aiming at predicting grades for semester $t + 1$, i.e., g_{t+1} . After training the model, any out-of-sample student's hidden state (i.e., abstract "knowledge" state) can be inferred by presenting their enrollment and grade histories to the model. This hidden state is then used to infer which next course will produce a hidden state in which the student is predicted to perform above threshold on their target course. Candidate next course are selected by exhaustive enumeration, as in [3], and by backpropagation, as introduced here.

Algorithm 1 Backprop-based course recommendation

```

1: procedure FIRST STAGE (INFERENCE)
2:    $h_{t-1}, g'_t = \text{GradePredictionModel}(g_1 \dots g_{t-1}, c_2 \dots c_t, h_0)$ 
3: procedure SECOND STAGE (INDIVIDUAL FITTING)
4:    $s \leftarrow (1, 0)_n$ 
5:    $w_{sg} \leftarrow 1$ 
6:    $g'_t \leftarrow \text{filters}(g'_t)$ 
7:    $c_{t+1} \leftarrow \text{target course (one-hot)}$ 
8:    $g_{t+1} \leftarrow \text{target course grade (one-hot for grade } \geq B)$ 
9:   loop:
10:    if  $\text{DecreasedProportion}(\text{loss}) < \sigma$  then return  $g_t$ 
11:    else
12:       $g_t \leftarrow w_{sg} \odot s \odot g'_t$ 
13:       $\neg, g'_{t+1} \leftarrow \text{GradePredictionModel}(c_{t+1}, g_t, h_{t-1})$ 
14:       $\text{loss} = \text{MaskedCrossEntropy}(g'_{t+1}, g_{t+1})$ 
15:       $\text{Backpropagation}(\text{loss})$ 
16:    $\text{recommended\_courses} = \text{Top10}(g_t)$ 

```

The backpropagation-based recommendation stage (Algorithm 1) parameterizes potential recommendations as weights to be learned by beginning with the first level of the input at time slice t set as a constant starting vector $s = (1, 0)_n$ where n is the number of unique courses and the position in s corresponding to the $\geq B$ grade category is set to 1. Then s is element-wise multiplied by the weight coefficients w_{sg} , initially set to an all **1s** vector. Additional constraints are added in the form of filters based on domain knowledge from [3]. For example, a recommended course cannot be one the student has previously passed and also cannot be of a higher course designation level than the target course (e.g., an upper division course, intended for 3rd and 4th year students, cannot serve as preparation for a lower division course, intended for 1st and 2nd year students). Therefore, g'_t is updated based on all filters by setting all the B positions to 1 with regard to the remaining candidate courses and the other positions to 0. Next, we update g_t by element-wise multiplying with g'_t to ensure the courses that are filtered out will not have ones in their positions in g_t , which is the input to the hidden states h_t . Then, the updated g_t is concatenated with c_{t+1} as the final input. Figure 3 demonstrates the whole process discussed above, namely, the calculation of the updated g_t and the final a_t . The objective of this recommendation phase of the model is to maximize the probability of attaining an above threshold grade (i.e., B) on the target course. Finally, we take the top-10 courses by coefficient value in g_t which adhere to the domain filters and grade performance constraint.

3.1 Experiment

3.1.1 Dataset. We used the same dataset as in [3], which contained 164,196 anonymized undergraduate students with 4.8 million class enrollments in 10,430 courses from Fall 2008 through Spring 2017 at UC Berkeley. Our sampling of courses to serve as target courses in our offline experiments remain the same as in prior work, where the criterion was to choose historically difficult courses. These courses were set as the target in the Fall 2016 semester, with Spring 2016 as the recommendation semester, and the previous semesters as the students' course histories. Only students with at least two semesters prior to Spring 2016 were included. The model was considered to have made a successful set of recommendations if at least one of the 10 recommended courses for 2016 Spring matched a student's actual enrollment in that semester who achieved above threshold on the target course. The overall accuracy was the number of students we have correct predictions for divided by the total number of students.

Table 1: Course Recommendation Evaluation

recommendation model		accuracy		
model	Top-N	high perf. students	gap	overall
RNN+Enumeration	top-10	57.65	21.21	49.60
RNN+Backprop	top-10	77.17	6.51	75.78

3.1.2 Experiment Results. We report the recommendation accuracy (Table 1) on students achieving above threshold performance on the target course ("high performance students") and the accuracy gap between high performance students and lower performance students with respect to the target course. We compare the backpropagation-based recommendation inference technique ("RNN+Backprop"), introduced in this paper, to the exhaustive enumeration method, introduced in [3]. Higher recommendation accuracy on high performance students than on lower performance students is support, but not proof, of the reasonableness of the preparation course recommendations being made. Therefore, the ideal results should be high recommendation accuracy on high performance students and a large accuracy gap between high performance students and lower performance students. We find that the backpropagation-based recommendation method achieves around 26% overall recommendation accuracy improvement but also has a far reduced ability to differentiate between course histories producing an above or below threshold grade on our selection of target courses.

3.1.3 Exemplars. (1) A student majoring in Mathematics², who achieved an *A* in *Discrete Mathematics* and a *B* in *Linear Algebra*, is recommended the top course of *Data Structures* as preparation for the target course of *Foundations of Data Science*. A student majoring in Sociology, with neither math nor computer science class work, is recommended the course *The Beauty and Joy of Computing*, an introduction to programming, as the top preparation course for the same target course. (2) The same student majoring in Mathematics

²Student major is presented to the reader for context; however, it did not improve grade prediction and was therefore not included in the model.

is recommended the course *Introduction to Economics* in preparation for *Economic Analysis-Macro*, while the same student majoring in Sociology, who has achieved a *B* in *Introduction to Economics*, is recommended the course *Linear Algebra*.

4 APPLICATION TO MOOC QUIZ PREPARATION

The goal-based recommendation model for learners in MOOCs is designed to suggest resources within the online course (e.g., book pages and videos) that are anticipated to help them prepare for an upcoming assessment or assessment they are struggling with, such as a quiz (i.e., collection of problems). This approach is not applicable exclusively to massive open online courses; however, MOOCs possess the important attributes of logging all course material views and individual problem answers and attracting large scale attendance in the tens of thousands of learners. In the MOOCs context, a time slice corresponds to a moment a student conducts an action in the online course. This could be playing a particular video, opening a text book page, or answering a problem. These actions appear in the MOOC logs as *go_to_page*, *go_back_to_page*, and *go_to_the_next_page* for page actions; *playing_video* for video actions; and *problem_check* for problem answer actions. All the past actions of a learner may contribute to whether the learner will answer a problem correctly in the future. Similar to the course grade prediction model, an RNN is used to infer a hidden state based on an embedding of objects, in this case course pages for a single course, in order to maximize predicted correctness on future attempted problems. A learner's action at time slice t is denoted by $a_t = (v_t, p_t)$; $v_t = (v_t^1, v_t^2, \dots, v_t^n)$ represents the total n pages with book text or videos in that course which are candidates for recommendation, where only the page a learner viewed at time slice t is set to 1, and others to 0; $p_t = (p_t^{1c}, p_t^{1i}, \dots, p_t^{mc}, p_t^{mi})$ represents whether the learner answered a problem correctly at time slice t , where m is the number of problems in that course, $(p_t^{kc}, p_t^{ki}) = (1, 0)$ denotes the k th problem is answered correctly, and $(p_t^{kc}, p_t^{ki}) = (0, 1)$ denotes the k th problem is answered incorrectly. A learner can only conduct one action in a time slice, so a_t is a one-hot encoding, where only one of v_t and p_t can be a one-hot encoding at a time slice, and the other is a zero vector. The problem correctness prediction model can be expressed as $h_{t-1}, p'_t = RNN(v_{t-1}, p_{t-1}, h_{t-2})$. Note that we do not aim to predict the next page a learner is going to [5], so v'_t is not included in the output of the model. The backprop-based approach to recommendation is shown in Algorithm 2, with the input topology for the recommendation time slice in Figure 4.

4.1 Masked Loss

Similar to the course grade prediction task, where only grade labels for courses a student received a grade for can be used in calculating the loss, in the context of MOOCs, only the problems the learner answered can be used. Unlike the college or university setting, time slices in MOOCs may or may not contain a grade label. Learners may be viewing pages or videos in a time slice instead of answering questions; therefore, the predictions at time slices with no problem answering actions are masked, and the predictions of problems not answered at a time slice with a problem answer are also masked in the otherwise vanilla cross-entropy loss function. The masked loss

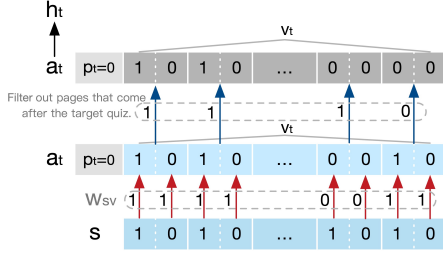


Figure 4: Input topology for individual backprop-based MOOC quiz preparation

function is expressed as:

$$\text{loss} = - \sum_t \sum_{k, (p_{t+1}^{kc}, p_{t+1}^{ki}) \neq 0} (p_{t+1}^{kc}, p_{t+1}^{ki})^T \log \text{softmax}(p_{t+1}^{kc}, p_{t+1}^{ki})$$

Algorithm 2 Backprop-based MOOC page recommendation

```

1: procedure FIRST STAGE (INFERENCE)
2:    $h_{t-1}, p'_t \leftarrow \text{ProbCorrPredictionModel}(p_1 \dots p_{t-1}, v_1 \dots v_{t-1}, h_0)$ 
3: procedure SECOND STAGE (INDIVIDUAL FITTING)
4:    $s \leftarrow (1, 0)_n$ 
5:    $w_{sv} \leftarrow 1$ 
6:    $p_{t+1} \leftarrow$  all correct for the target quiz (multi-hot for
    $(p_{t+1}^{kc}, p_{t+1}^{ki}) = (1, 0)$  where problem  $k$  is in the target quiz)
7: loop:
8:   if  $\text{DecreasedProportion}(\text{loss}) < \sigma$  then return  $v_t$ 
9:   else
10:     $v_t \leftarrow w_{sv} g \circ s$ 
11:     $v_t \leftarrow \text{FilterOutPagesComeAfterTargetQuiz}(v_t)$ 
12:     $p'_{t+1} \leftarrow \text{ProbCorrPredictionModel}(v_t, p_t, h_{t-1})$ 
13:     $\text{loss} = \text{MaskedCrossEntropy}(p'_{t+1}, p_{t+1})$ 
14:     $\text{Backpropagation}(\text{loss})$ 
15:  $\text{recommended\_pages} = \text{Top10}(v_t)$ 

```

4.2 Experiment

We evaluated the MOOC page recommendation in two stages; one in which the model's ability to predict future performance is evaluated, and a second in which we evaluate the would-be recommendations of the model in hindsight. In course recommendation, the predictive accuracy was reported in prior work. As MOOCs were not a context used in prior work, we present this predictive evaluation using a 10-fold cross validation to train and test the accuracy of problem correctness prediction. For evaluation of the recommendations from this model, we set the goal to be maximizing performance on an upcoming quiz. A quiz is a collection of problems that appear on the same page of a MOOC. A single problem could potentially serve as the goal; however, we believe that the course pages and videos may not be so granular as to pertain to only a single problem. Additionally, maximizing the average performance on a collection of problems may produce more satisfying recommendations that focus on a general concept to be learned. We do not recommend quiz pages and filter out pages and videos appearing in chapters

that come after the target quiz to reasonably constrain candidate recommendations. In our evaluation, we divided the learner sequences into two groups: (a) learners whose action right before a quiz was among our top 10 recommendations and (b) learners whose action right before a quiz was not. Then, we calculated the average correctness of the quiz for each group, expecting a higher average correctness among the group whose action overlapped with our would-be recommendations.

4.2.1 Dataset. We used a dataset containing 1,869,406 learner actions from 4,763 learners with at least one problem answer in a Physics course on *edX*, a popular MOOC platform. The number of correct problem answers was 177,623 and the number of incorrect problem answers was 104,711. The remaining actions were non-problem answering actions involving the 868 book or video pages.

4.2.2 Experiment Results. The average problem correctness prediction accuracy of the model is 77.17%, compared to the proportion of correct answers, 62.91%. The average *AUC* is 0.8749, meaning the model has far-above-chance power to discriminate between the sequences that lead to correct answers and sequences that lead to incorrect answers. The results of the recommendation evaluation is shown in Table 2. We can see that learners whose actions corresponded with our would-be recommendations scored slightly higher than those whose actions did not, with the Enumeration model better distinguishing between the two than the Backprop model. We also see a relatively low overall accuracy in predicting actions before a quiz, with 11.04% from Enumeration and 12.17% from Backprop.

Table 2: MOOC Page Recommendation Evaluation

recommendation model		quiz percent correct		accuracy
model	Top-N	correct rec. grp.	gap	overall
RNN+Enumeration	top-10	72.12	8.10	11.04
RNN+Backprop	top-10	66.51	1.05	12.17

5 DISCUSSION AND LIMITATIONS

This recommendation approach has implications for other non-educational domains such as e-commerce, recommending reviews or product pages in service of a consumer's goal of choosing a product in a particular category. The approach has several limitations: (1) the backprop-based model infers independent probabilities for every potential recommendation instead of being constrained to a limited number of simultaneous inputs (2) this lack of constraint poses challenges for multi time-slice imputation (3) domain knowledge is required in order to constrain the inferences to help avoid illegitimate, spurious recommendations (4) the algorithm can give generic, but not personalized recommendations when no sequence data or other information has yet been collected from the learner.

ACKNOWLEDGMENTS

This work was partly supported by the National Natural Science Foundation of China (71772101/71490724) and the United States National Science Foundation (1547055/1446641).

REFERENCES

- [1] Michael Geoffrey Brown, R. Matthew DeMonbrun, and Stephanie D. Teasley. 2018. Conceptualizing Co-enrollment: Accounting for Student Experiences Across the Curriculum. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge (LAK '18)*. ACM.
- [2] Asmaa Elbadrawy and George Karypis. 2016. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 183–190.
- [3] Weijie Jiang, Zachary A Pardos, and Qiang Wei. 2019. Goal-based Course Recommendation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. ACM, 36–45.
- [4] Kenneth R Koedinger, John R Anderson, William H Hadley, and Mary A Mark. 1997. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED)* 8 (1997), 30–43.
- [5] Zachary A Pardos, Steven Tang, Daniel Davis, and Christopher Vu Le. 2017. Enabling real-time adaptivity in MOOCs with a personalized next-step recommendation framework. In *Proceedings of the Fourth (2017) ACM Conference on Learning@Scale*. ACM, 23–32.
- [6] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.
- [7] Zhiyun Ren, Xia Ning, Andrew Lan, and Huzefa Rangwala. 2019. Grade Prediction Based on Cumulative Knowledge and Co-taken Courses. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*. EDM, 158–167.