Course Recommendation of MOOC with Big Data Support: A Contextual Online Learning Approach

Yifan Hou, Pan Zhou*, Jie Xu†, *Member, IEEE*, Dapeng Oliver Wu ‡, *Fellow, IEEE*
*School of Electronic Information and Communications, Huazhong University of Science and Technology
†Department of Electrical and Computer Engineering, University of Miami, USA
‡Department of Electrical and Computer Engineering, University of Florida, USA
Email: yifan.hou.z@gmail.com, panzhou@hust.edu.cn

***Abstract*—With the advent of the big data era of MOOC, enrolled students and offered courses become numerous and diverse, resulting in a large amount of data and complex curriculum relationships. Thus how to recommend appropriate course to improve students' learning outcomes has become a daunting task. The state-of-the-art works ignore some significant features in course recommendation of MOOC: heterogeneity of large-scale user groups, sequence problem in courses and foreseeable quantitative explosion of courses and users. This paper proposes a systematic methodology for recommending personalized courses with considering the sequence of learning curriculum. The system works by recommending the course with the highest reward to a user. New feedback of the user is then recorded and will be used to improving the performance of recommendation for future students. The core component is a novel online learning algorithm based on hierarchical bandits with known smoothness. We analyze the performance of our proposed online learning algorithm in terms of regret, and prove the asymptotic optimality of the proposed algorithm. Experimental results are provided to verify our theory.**

***Index Terms*—MOOC, course recommendation, big data, online learning, hierarchical bandits**

I. Introduction

Due to its novelty and openness, MOOCs have taken higher education by storm in recent years [1], [2], with over millions of enrolled students and over thousands of courses. However, a recent study shows that the completion rate of MOOCs is below 10% [3]. One of the main reasons is mismatched background knowledge and skills of students [4]. Thus how to recommend proper-difficulty courses to users to improve the learning effectiveness in MOOCs becomes a key problem.

To provide high-accuracy recommendation, three essential points should be considered: heterogeneity of numerous students, relationships between courses and predictable big data-oriented model. With MOOC becoming increasingly popular, the compound of students is fairly complicated. Users are from all over the world, having quite diverse interests or education background. Thus the context information of users is non-negligible, without which the accuracy of recommendation would have a significant deviation. As for courses, there are many sequential restricts between them. For example, before we start "Stochastic Process", the course "Probability Theory" should be finished first. Ignoring the sequentiality during recommending process would give student courses unmatched with their background knowledge, which might cause the high dropout rates. MOOC would have been developing for years due to the advanced concept and convenient learning method, which means the number of students and courses are and will grow further. Thus the system is required to handle the large-scale dataset to deal with the development trend.

In this paper, we propose a contextual recommender system to deal with the challenges mentioned above. We consider a measurable context space with Lipschitz condition, where space is divided into many subspaces to represent different types of students. The course space obeys a weak Lipschitz condition and is organized in a Monte-Carlo Tree [5] structure. For different types of users, the course trees are different. Each tree node represents a set of courses. Starting with the root node containing the entire course set, course recommendation becomes more refined and accurate by growing child tree nodes containing subsets of the courses. By defining and working with the dissimilarity function of courses, the proposed system is agnostic to the specific number of courses and hence, it can handle a large number of course databases. To capture the prerequisite dependencies among courses, course clusters are defined. We prove that the proposed algorithm achieves a sublinear learning regret compared to the optimal recommendation policy. Therefore, it is asymptotically optimal. Experimental results are provided to verify our theoretical conclusions. Moreover, our system could also perform well in other popular applications with large dataset and complex data structure.

II. Related Work

There are lots of existing works on course recommendation of MOOC. Collaborative filtering approach is very welcomed. There are three branches of filtering-based approaches: collaborative filtering [6], [7], content-based filtering [8] and hybrid approach [9]. Collaborative filtering approach gathers the learning records of users and classifies them into groups based on the records, then the recommended items are selected according to related records of that group. Content-based filtering recommends items to the users which are relevant to their previous learning records. Hybrid approach is the combination of two methods above. Due to the principles of filtering methods, it is obvious that if the number of learning records or courses becomes large, the deviation of recommendation would be unacceptable. Works with collaborative filtering

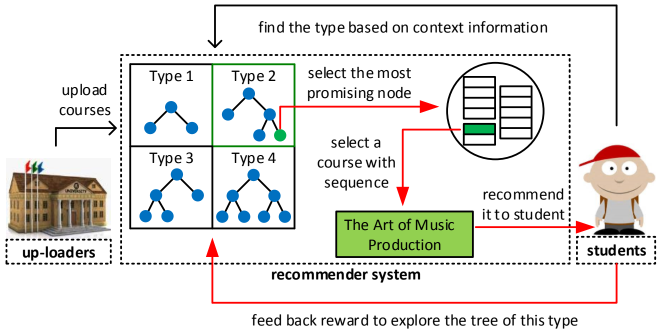

Fig. 1: Course Recommendation System of MOOC

performed well when MOOC was initially proposed.However, with the rapid development, the completion rate becomes fairly low with filtering models.

Online learning methods could handle relatively large datasets and consider context information in the meanwhile. A growing number of related works have been applied into course recommendation [10], [11], which also consider the sequence problem during recommending process. Some similar ideas based on contextual bandits [12], [13] are applied to other recommender systems, e.g. article recommendation [14] and social network [15]. However, the performance of these works is still not well with very large dataset. Taking this into consideration, some works supporting the really large-scale dataset have been proposed, where they could handle infinite data theoretically [16], [17] based on Monte-Carlo Tree structure [5]. But these works do not consider the context information, which plays an important role in course recommendation of MOOC. Thus in this paper we consider a contextual online learning algorithm based on Monte-Carlo Tree Search (MCTS) [5] that could deal with big data. Besides, the sequentiality between courses has also been considered, and hence our system is specifically design for course recommendation.

III. Problem Formulation

A. System Model

Figure 1 illustrates the considered MOOC course recommendation system. The student set is denoted by $\mathcal{S}$, which may expand over time. Students arrive at the MOOC platform in sequence $t = 1, 2, ..., T$. In practice, the sequence t can be the sequence of students' finished courses or others that can represent the sequential concept. At each time t, a student $s \in \mathcal{S}$ arrives and the system recommends a course to this student depending on his/her personal information (e.g. age, cultural background, nationality, educational level etc.) and learning record (e.g. what courses have been taken). We use $x(t)$ to denote the personal/context information of this student s at time t. Note that a student may come back with multiple times to study different courses.

The set of courses is denoted by $\mathcal{C}$. Courses are further organized in course clusters (which will be defined shortly), the set of which is denoted by $\mathcal{G}$. There are up-loaders in the system, which provide the learning resources previously. For student s who arrives at time t, the system first determines the course tree of this student type, which is a personalized course data structure built using previous similar students studied courses and their feedback, based on the student s's context information $x(t)$. Using the personalized course tree, the system searches for the course cluster $g \in \mathcal{G}$ with the highest reward estimate for student s and then recommends a course c of g depending on the students learning record. Once the student finishes the course, he/she sends his/her degree of satisfaction (or reward), denoted by $r_{x,c}(t)$, as feedback to the recommendation system, which uses this information to update the estimated rewards of course c and cluster g, thereby improving course recommendation for future students.

B. Context Model for Individualization

Student context information space $\mathcal{X}$ is a multi-dimensional space with d_X dimensions. Take educational level dimension for instance, 0 represents elementary, 1 represents expert and any value between 0 and 1 represents a level between elementary and expert. For analytical simplicity and without loss of generality, we normalize the value space on every content dimension to $[0, 1]$. The entire context space thus can be represented by $\mathcal{X} = [0, 1]^{d_X}$, which is a unit hypercube geometrically. Context space is further partitioned into smaller subspaces as follows. On each context dimension, the value space is uniformly divided into n_T parts. As a result, the entire context space is divided into $(n_T)^{d_X}$ parts, where each part is a d_X-dimensional subspace with dimensions $\frac{1}{n_T} \times \frac{1}{n_T} \times ... \frac{1}{n_T}$. Let $\mathcal{P}_T$ denote the set of subspaces. Therefore, students are divided into $(n_T)^{d_X}$ types depending on their context information. The system utilizes the similarity among students of the same type for course recommendation. We call n_T the *slicing number*, which needs to be carefully designed depending on the value of T. If n_T is too small (too few types), then student context information is not efficiently utilized to enable personalized course recommendation. If n_T is too large (too many types), then there are not sufficiently many students of the same type to enable effective learning.

C. Cluster-Course Model for Recommendation

Each course is described by its feature vector c which includes information regarding the course topic, in which language the course is taught, the targeted professional level and the institution that offers this course etc. Course space is represented by $\mathcal{C}$ with d_C dimensions. We use a dissimilarity function $D_C^x(c, c')$ to model the difference between any two courses $c, c' \in \mathcal{C}$, which is specific to the student context $x \in \mathcal{X}$ and formally defined as follows.

Definition 1. ***(Dissimilarity)*** *The dissimilarity* $D_C^x(c, c')$ *between any two courses* $c, c' \in \mathcal{C}$ *for a context* $x \in \mathcal{X}$ *is a non-negative mapping* $D_C^x : C \times C \to R_+$ *with* $D_C^x(c, c') = 0$. *The dissimilarity* $D_C^P(c, c')$ *between any two courses* $c, c' \in \mathcal{C}$ *for a student type* $P \in \mathcal{P}_T$ *is* $D_C^P(c, c') = \sup_{x \in P} D_C^x(c, c')$.

Note that course dissimilarity is student context-specific. In this paper, we consider that the course dissimilarity functions are given, which may be constructed using education domain

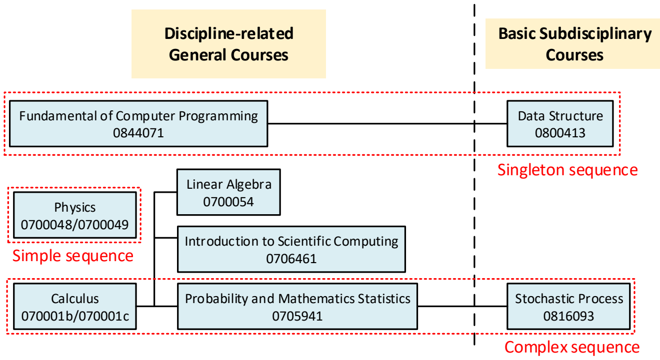

Fig. 2: Cluster-Course Model: Three Basic Types of Clusters

knowledge. Typically, courses that are more relevant have smaller dissimilarity between them.

An important distinct feature of course recommendation compared to other recommender systems is that courses have prerequisite dependencies. Ignoring this dependency results in low recommendation efficacy. To capture this feature, we introduce the concept of course cluster, which is a sequence of courses connected by prerequisite requirements. Figure 2 illustrates some example courses and their prerequisites dependencies in the undergraduate curriculum of Huazhong University of Science and Technology. Depending on the specific dependency structure, we define three types of clusters: *A Simple cluster* is composed of courses that do not have prerequisite dependencies with courses outside of this cluster; *A Singleton Cluster* is composed of one course, which does not have any prerequisite dependencies with other courses; *A Complex Cluster* is composed of courses that may have prerequisite dependencies with courses outside of this cluster.

For each course cluster $g \in \mathcal{G}$, we denote L_g^s as the sequence number of the course in the cluster that student s is supposed to take. We collect the completion states L_g^s for all course clusters $g \in \mathcal{G}$ for student s in the notation $\mathcal{L}_s = \{L_g^s\}_{g \in \mathcal{G}}$.

As aforementioned, the system utilizes Monte-Carlo Tree [5] structure to maintain the course clusters and make personalized course recommendations. For every type $P \in \mathcal{P}_T$ students, the system keeps a different binary course tree, with nodes on the tree representing subsets of course clusters. Let $\mathcal{N}_{h,i}^P, \forall i \in \{1, 2, ...2^h\}$ denote a tree node of depth h for type $P \in \mathcal{P}_T$ students. Clearly, there are maximally 2^h such nodes. With a little abuse of notation, we also let $\mathcal{N}_{h,i}^P \in \mathcal{G}$ to denote the subset of course clusters associated with this tree node. Since this is a binary tree, it has the following properties

$$\mathcal{N}_{0,1}^P = \mathcal{G}, \tag{1}$$

$$\mathcal{N}_{h,i}^P \cap \mathcal{N}_{h,j}^P = \emptyset \ \ (i \neq j), \tag{2}$$

$$\mathcal{N}_{h,i}^P = \mathcal{N}_{h+1,2i-1}^P \cup \mathcal{N}_{h+1,2i}^P. \tag{3}$$

Property (1) indicates that the root node contains all course clusters and hence all courses. Property (2) states that course clusters associated with two different nodes of the same depth are mutually exclusive. Property (3) describes the relationship between a node and its two child nodes. The binary tree structure also implies $\mathcal{G} = \cup_{i=1}^{2^h} \mathcal{N}_{h,i}^P$, namely the entire set of course clusters can be covered by all tree nodes of the same depth. Next, we define the diameter of a tree node, which describes the maximum dissimilarity between any two courses on that node.

Definition 2. ***(Diameter)*** *The diameter of node $\mathcal{N}_{h,i}^P$ is defined as $diam(\mathcal{N}_{h,i}^P) = \sup_{c,c' \in \mathcal{N}_{h,i}^P} D_C^P(c, c')$.*

Note that the *diameter* of a node is contingent upon the adopted dissimilarity function. As will be clear later when we describe our algorithm, the course tree grows as the algorithm learns the rewards of the courses by exploring different courses. Intuitively, the diameter of a node should decrease exponentially with the increase of the depth on average under a "proper" tree growing process (i.e. a process that tries to divide a node into two child nodes as evenly as possible). Considering the big data-oriented situation, we make this intuitional formal in Assumption 1.

Assumption 1. ***(Partition uniformity)*** *There exist constants k_1, m and θ such that for any student type $P \in \mathcal{P}_T$, the tree growing process satisfies $\frac{k_1}{\theta}(m)^h \leq diam(\mathcal{N}_{h,i}^P) \leq k_1(m)^h$.*

We call θ the partition uniformity parameter with respect to k_1 and m. This assumption restricts the possible deviation of courses on a node $\mathcal{N}_{h,i}^P$. We note that the constants depend on the adopted dissimilarity function and we need them only for the purpose of theoretical analysis. They are not needed for running our algorithm.

D. The Regret of Learning

We call this feedback information mentioned above the recommendation *reward*, denoted by $r_{x,c}(t) \in [0, 1]$, depending on the students context x and the recommended course c. The reward is modeled as an i.i.d. random variable drawn from some unknown distribution with a *priori* unknown expected reward $u_{x,c} = \mathbb{E}[r_{x,c}(t)]$. The i.i.d assumption models the fact that the major influencers of the reward are user context and the specific course, which are typically stationary. A simple trick that uses two i.i.d. processes to bound a non-i.i.d. process [18] can be adopted to capture other nonstationary influencers such as user mood and societal trends.

The optimal course for student context x is therefore $c^{x*} = arc\max_{c \in \mathcal{C}} u_{x,c}$. The optimal course for students of type P is defined as $c^{P*} = arc\max_{c \in \mathcal{C}, x \in P} u_{x,c}$. Since c^{P*} is relevant to student type rather than specific context, we define $u_{c^{P*}}$ as the expected reward of overall optimal course and its student type. We now make a widely-adopted assumption on the rewards.

Assumption 2. ***(Lipschitz condition)***

(a) (Lipschitz condition for context) There exists $L_X > 0$ and α such that for two random contexts $x, x' \in \mathcal{X}$, we have $|u_{x,c} - u_{x',c}| \leq L_X ||x - x'||^\alpha$ in terms of any course $c \in \mathcal{C}$.

(b) (Weak Lipschitz condition for course) For two random courses $c, c' \in \mathcal{C}$ with a given context vector $x \in \mathcal{X}$, we have $u_{x,c} - u_{x,c'} \leq \max\{u_{x,c^{x*}} - u_{x,c}, D_C^x(c, c')\}$.

We note that for our algorithm to run, only the parameter α needs to be known but not the parameter L_X [19]. Notice that a more standard Lipschitz assumption is $|u_{x,c} - u_{x,c'}| \leq D_C^x(c,c')$, which is a stronger assumption than the above *weak Lipschitz condition*. Under the above assumption,

$$u_{x,c} - u_{x,c'} \leq \max\{u_{c^{P*}} - u_{x,c}, D_C^P(c,c')\}. \tag{4}$$

In this paper, we use learning regret to measure the recommendation reward loss due to learning under uncertainty, which is defined as follows.

$$\mathbb{R}(T) = \sum_{t=1}^{T} u_{x(t),c^{x(t)*}} - \left[\sum_{t=1}^{T} r_{x,c}(t)\right], \tag{5}$$

where $u_{x(t),c^{x(t)*}}$ is the expected reward by recommending the optimal course to student of context $x(t)$ and $r_{x,c}(t)$ is realized reward derived by our algorithm. The learning regret reflects the convergence rate to the optimal recommendation policy. If the regret is sublinear in the number of arrived students T, namely $\mathbb{R}(T) = O(T^\gamma)$ with $0 < \gamma < 1$, then the algorithm converges to the optimal policy since $\lim_{T\to\infty} \mathbb{R}(T)/T \to 0$.

IV. Contextual Hierarchical Tree

In this section we propose our online learning algorithm CHT strategy to learn the optimal course recommendation policies using big data of MOOCs, and prove that our algorithm achieves a sublinear learning regret.

We introduce some additional useful notations first. We define $C(\mathcal{N}_{h,i}^P)$ as the set of all nodes on the subtree whose root node is $\mathcal{N}_{h,i}^P$, namely $\mathcal{N}_{h,i}^P$ and all its descendants. Recursively, $\mathcal{N}_{h,i}^P$ can be represented by

$$C(\mathcal{N}_{h,i}^P) = \mathcal{N}_{h,i}^P \cup C(\mathcal{N}_{h+1,2i-1}^P) \cup C(\mathcal{N}_{h+1,2i}^P).$$

We use $\mathcal{N}(t)$ to denote the tree node (i.e. the set of course clusters) selected by our algorithm at time t. Based on the tree node selection history, the number of times that courses in $\mathcal{N}_{h,i}^P$ have been selected for student type P up to time t, denoted by $T_{h,i}^P(t)$, can be calculated as

$$T_{h,i}^P(t) = \sum\nolimits_{t=1}^{T} \mathbb{I}\{\mathcal{N}(t) \in C(\mathcal{N}_{h,i}^P)\}.$$

We use Γ^P to denote the set of tree nodes that have been selected for student type P. Moreover, the path of tree node selection is denoted by Ω^P, namely the specific tree nodes that are selected as the tree depth increases. Two important concepts are introduced next.

Reward Upper Confidence Bound (B value). The reward upper confidence bound for node $\mathcal{N}_{h,i}^P$ is

$$B_{h,i}^P(t) = \hat{\mu}_{h,i}^P(t) + D_d + D_n + D_c, \tag{6}$$

where $\hat{\mu}_{h,i}^P(t)$ is the sample-mean reward of recommending courses in node $\mathcal{N}_{h,i}^P$ to students of type P up to time t, i.e.

$$\hat{\mu}_{h,i}^P(t) = \frac{T_{h,i}^P(t-1)\hat{\mu}_{h,i}^P(t-1) + [T_{h,i}^P(t) - T_{h,i}^P(t-1)]r_{x,c}(t)}{T_{h,i}^P(t)}, \tag{7}$$

and D_d, D_n, D_c are positive real numbers representing the reward deviations caused by different factors during the recommending process. Specifically, $D_d = \sqrt{k_2 \ln T / T_{h,i}^P(t)}$ captures the confidence bound due to the exploration and exploitation tradeoff. According to Assumption 1 and Assumption 3, $D_n = k_1(m)^h$ captures the dissimilarity between different course clusters and the dissimilarity between courses

Require: Related parameters, Time T and Clusters $\mathcal{G}$.
Subroutines: Selection − Expansion and Backpropagation
Initialize: For any $s \in \mathcal{S}$, let $L_i^s = 0$ with any $g \in \mathcal{G}$; $\Gamma^P = \{\mathcal{N}_{0,1}^P\}$, $E_{1,i}^P(0) = E_{max}$ $(i = 1,2)$ for any $P \in \mathcal{P}_T$.

```
1:  for t = 1, 2, ...T do
2:      Find context subspace P due to x(t) from student s
3:      Initialize the current node N(t) ← N^P_{0,1}
4:      Build the path set of nodes Ω^P ← {N(t)}
5:      Call Selection − Expansion (Γ^P)
6:      Select a cluster g from the node N^P_{h,i} randomly
7:      if L^s_j > the number of courses in g then
8:          Select another cluster
9:      end if
10:     Refresh learning record L^s_j = L^s_j + 1
11:     Based on L^s_j, recommend the course c of g to student s
12:     Get the reward r_{x,c}(t) from student s
13:     for all nodes ∈ Ω^P do
14:         Refresh selected times T^P_{h,i}(t) = T^P_{h,i}(t − 1) + 1
15:         Refresh the average reward according to (7)
16:         Refresh the B value on the path according to (6)
17:     end for
18:     E^P_{h+1,2i−1} = E_max, E^P_{h+1,2i} = E_max
19:     Call Backpropagation (Ω^P)
20: end for
```

Fig. 3: Pseudocode for CHT algorithm

in the same cluster. Due to Assumption 2, $D_c = L_X(\frac{\sqrt{d_X}}{n_T})^\alpha$ is the reward deviation caused by the discrepancy in the student context space.

Reward Estimate (E value). The reward estimate for a tree node captures the reward upper confidence bound of itself and those of its child nodes,

$$E_{h,i}^P(t) = \min\Big\{B_{h,i}^P(t), max\{E_{h+1,2i-1}^P(t), E_{h+1,2i}^P(t)\}\Big\}. \tag{8}$$

Therefore, $E_{h,i}^P(t)$ imposes an optimistic and tighter upper bound on the reward of course clusters on node $\mathcal{N}_{h,i}^P$. Note that $E_{h,i}^P(t) \leq B_{h,i}^P(t)$. In particular, for the leaf cluster nodes, we have $E_{h,i}^P(t) = B_{h,i}^P(t)$.

The CHT algorithm works as follows. When a student $s \in \mathcal{S}$ arrives at the system at time t, the algorithm first determines the student type P using the students context information $x(t)$ (Line 2). Thus, the corresponding course tree can be found using P. After that, we initialize some parameters such as the search path $\Gamma^P = \{\mathcal{N}_{0,1}^P\}$ (Lines 3-4). Then the algorithm searches for the best course with four steps of Monte-Carlo Tree Search [5]. *Selection* and *Expansion* are completed by calling subroutine Selection − Expansion (Lines 5). Then a random course cluster is selected from node $\mathcal{N}_{h,i}^P$ (step *Simulation*) (Line 6). Based on the information of learning record L_j^s, a course c is recommended. If all courses in that cluster have been taken by the student, it selects a related cluster to recommend a course (Line 7-11). After the recommendation, the reward is fed back. Then the algorithm updates the B value and after that the two unexplored child nodes are added into the tree structure in terms of this student types P (Line 12-18). For the last step *Backpropagation*, the E value is updated by calling the Backpropagation subroutine of relevant nodes on the tree (Lines 19).

The CHT algorithm uses two subroutines, namely Selection − Expansion (Figure 4) and Backpropagation (Figure 5). The Selection − Expansion aims to find the node with

1: **for** all nodes $\mathcal{N} \in \Gamma^P$ **do**
2:    **if** $E^P_{h+1,2i-1} > E^P_{h+1,2i}$ **then**
3:       $Temp = 1$
4:    **else if** $E^P_{h+1,2i-1} < E^P_{h+1,2i}$ **then**
5:       $Temp = 0$
6:    **else**
7:       $Temp \sim Bernoulli(0.5)$
8:    **end if**
9:    $\mathcal{N} \leftarrow \mathcal{N}^P_{h+1,2i-Temp}$
10:    Select the better node of child node into the path set $\Omega^P \leftarrow \Omega^P \cup \{\mathcal{N}\}$
11: **end for**
12: Select leaf node $\mathcal{N}$ and add it into the explored set $\Gamma^P \leftarrow \Gamma^P \cup \{\mathcal{N}\}$

Fig. 4: Pseudocode for Selection – Expansion

1: **for** $\Omega^P \neq \{\mathcal{N}^P_{0,1}\}$ **do**
2:    $\mathcal{N}(t) \leftarrow$ one leaf of Ω^P
3:    Refresh the values of E value according to (8)
4:    Delete the $\mathcal{N}(t)$ from Ω^P
5: **end for**

Fig. 5: Pseudocode for Backpropagation

highest reward estimate. Based on (8) we could find that node layer by layer and record the path in the same time (Line 1-11). Besides, the newly explored node is selected and added into set Γ^P (Line 12). The Backpropagation subroutine updates the E value with bottom-up sequence based on the path set by (8). It starts with leaf node, and finishes when root node is left in path set only. Note that the updated nodes are deleted from the set Ω^P during the process.

Theorem 1 discusses the learning regret with upper bound.

Theorem 1. *Considering the situation when parameters in CHT satisfy the assumptions for some dissimilarity, we could get the regret as*

$$\mathbb{E}[R(T)] \leq 16L_X(\sqrt{d_X})^\alpha (C_n \ln T)^{(1-\gamma)} T^\gamma + \frac{25}{[L_X(\sqrt{d_X})^\alpha]^{d_C+2}} (C_n \ln T)^{-\gamma} T^\gamma,$$

where C_n is a constant and $\gamma = \frac{d_X+\alpha(d_C+2)}{d_X+\alpha(d_C+3)}$.

Proof. *The proof is available in supplementary [20].* □

From Theorem 1, it is obvious the order of regret bound is sublinear, that is

$$\mathbb{E}[R(T)] = \mathcal{O}[(\ln T)^{1-\gamma} T^\gamma]. \quad (9)$$

Equation (9) indicates that our algorithm CHT has the sublinear regret $\lim_{T\to\infty} \mathbb{R}[T]/T = 0$, which assures that the optimal course could be found. Note that parameter α indicates the order of randomness with respect to context. The condition $\alpha = 0$ means the distribution is completely disordered, the regret bound is linear with $\mathcal{O}[(\ln T)^{1-\gamma} T^\gamma] = \mathcal{O}(T)$, which means the algorithm may not able to work. When $\alpha \to \infty$, we could get the best result with optimal distribution of context with $\mathcal{O}[(\ln T)^{1-\gamma} T^\gamma] = \mathcal{O}\left[T^{\frac{d_C+1}{d_C+2}} (\ln T)^{\frac{1}{d_C+2}}\right]$.

V. Numerical Results

We consider three benchmark algorithms that also utilize tree structures: ACR [13], HOO [16] and HCT [17]. Table I lists these algorithms in terms of seven comparison items, where "×" means nonsupport of this item. Adaptive Clustering Recommendation algorithm (ACR) [13] considers the contextual bandit problem. It utilizes tree exploration to reduce the set of elective items, where optimal item could finally be found in this way. The tree of items (arms) is built before the recommending process, which means this model is not appropriate for large-scale dataset or unknown-quantity data. Hierarchical Optimistic Optimization tree algorithm (HOO) [16] and High-confidence tree (HCT) [17] consider the bandit problem with infinite arms, however both of them could only work for one type of user since context is not considered. HCT introduces the threshold of nodes into the model and improves the curve dithering of regret.

A. Dataset

The dataset is composed of three parts: the data of courses, the context information of students and the feedback reward records. We get course data in the biggest MOOC platform in China, "iCourse"[1], which contains nearly all the Chinese online courses. We focus on courses offered by 125 universities in China. There are more than 3500 courses that make it difficult for students to choose. We use 2467 courses as the recommendation database. The course features include course's type, discipline, school that offered the course, instructor's type, duration, certificate requirements, compactness (school schedule), degree of heat (number of selected times). We use $d_C = 8$ course features in our experimentation. Since "iCourse" is the synthesis of MOOC platforms, the cluster information is incomplete. Thus in the experiments we treat each course as a singleton cluster.

We collect context information of 4939 anonymized students in Huazhong University of Science and Technology and Central China Normal University who have learned online before, with around 20 thousand learning records. To ensure the diversity of samples, the students are selected from different schools/departments and grades.The context features contain college entrance examination score, province of origin, current GPA, department, major, grade, school, nationality and other basic information including gender and age. We have $d_X = 10$ context features. The scores of courses and the degree of satisfaction have been collected as the reward, where we use the average value of the two factors to denote the learning effectiveness of students.

Note that the information has some defect, where some students or courses may not offer all the information above. We treat the missing value as NULL, which means any other values are the same with it. For other misty information e.g. curriculum evaluation, it is difficult to quantify or count. Thus we do not consider it.

B. Regret Comparison

Figure 6 illustrates the performance loss of algorithms in terms of cumulative regret and average regret. The overall trend is shown in cumulative regret diagram, we can see that

[1] http://www.icourses.cn

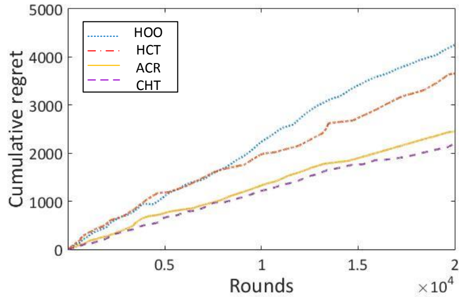

Fig. 6: Cumulative regret comparison

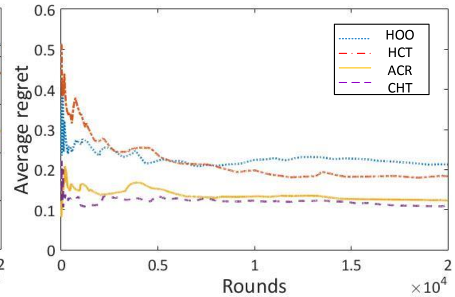

Fig. 7: Average regret comparison

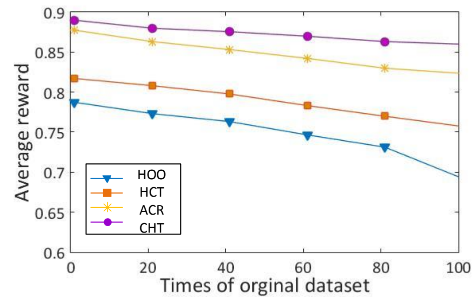

Fig. 8: Average regret v.s. scale of dataset

TABLE I: Theoretical Comparison

Algorithm	Big data	Context	Data Sequence	Time Complexity	Space Complexity	Regret
HOO [16]	✓	×	×	$\mathcal{O}(T \ln T)$	$\mathcal{O}(T)$	$\mathcal{O}\left(T^{\frac{d+1}{d+2}} (\ln T)^{\frac{1}{d+2}}\right)$
HCT [17]	✓	×	×	$\mathcal{O}(T \ln T)$	$\mathcal{O}\left(T^{\frac{d}{d+2}} (\ln T)^{\frac{2}{d+2}}\right)$	$\mathcal{O}\left(T^{\frac{d+1}{d+2}} (\ln T)^{\frac{1}{d+2}}\right)$
ACR [13]	×	✓	×	$\mathcal{O}\left(T^2 + K_E T\right)$	$\mathcal{O}\left(\sum_{l=0}^{E} K_l + T\right)$	$\mathcal{O}\left(T^{\frac{d_I+d_C+1}{d_I+d_C+2}} \ln T\right)$
RHT	✓	✓	✓	$\mathcal{O}(T \ln T)$	$\mathcal{O}(T)$	$\mathcal{O}\left(T^{\frac{d_X+\alpha(d_C+2)}{d_X+\alpha(d_C+3)}} (\ln T)^{\frac{\alpha}{d_X+\alpha(d_C+3)}}\right)$

considering context information could improve the recommended accuracy considerably.The advantage is even more obvious for the average regret in Figure 7. We evaluate the algorithm with different scales of datasets by increasing the number of courses and duplicating originally existing ones. We take the training rounds as 20 thousand, and run the algorithms with different scales of datasets. Figure 8 shows the results of average reward.Since more factors e.g. context and sequence are considered, our model can handle the growth of data better. The parameters should be chosen based on the practical situation. And the principle of parameter selection is to make more data available as much as possible while ensuring that the model does not overflow.

VI. Conclusion

This paper proposed an online learning algorithm for the course recommendation in MOOCs with big data support. We consider the dissimilarity among courses rather than specific quantity to handle tremendously large dataset. Providing personalized course recommendation improves the recommendation accuracy remarkably. The sequentiality of courses is also considered, with using cluster to recommend proper courses to users. Our experiment results on a real-world dataset showed that our algorithm significantly outperforms existing algorithms. Future work includes developing algorithms that can deal with dynamically increasing course datasets.

References

[1] L. Pappano, "The Year of the MOOC," *The New York Times*, vol. 2, no. 12, 2012.

[2] T. A. Baran, R. G. Baraniuk, A. V. Oppenheim, P. Prandoni, M. Vetterli, "MOOC adventures in signal processing: Bringing DSP to the era of massive open online courses," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 62-83, 2016.

[3] K. Jordan, "MOOC completion rates: the data," *Reference Source*, 2013.

[4] Y. Belanger, J. Thornton, "Bioelectricity: A quantitative approach Duke Universitys first MOOC," *Duke Center for Instructional Technology*, 2013.

[5] G. M. J. B. Chaslot, M. H. M. Winands, H. J. V. D. Herik, J. W. H. M. Uiterwijk, B. Bouzy, "Progressive Strategies for Monte-Carlo Tree Search," *New Mathematics and Natural Computation*, vol. 4, no. 3, pp. 343-357, 2008.

[6] C. G. Brinton, M. Chiang, "MOOC performance prediction via clickstream data and social learning networks," *IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2299-2307.

[7] M. Zhou, A. Cliff, A. Huang, et al, "M-CAFE: Managing MOOC student feedback with collaborative filtering," *Proceedings of the Second (2015) ACM Conference on Learning@ Scale. ACM,* 2015, pp. 309-312.

[8] M. J. Pazzani, D. Billsus, "Content-based recommendation systems," *The adaptive web* Springer Berlin Heidelberg, 2007, pp. 325-341.

[9] M. K. Khribi, M. Jemni, O. Nasraoui, "Recommendation systems for personalized technology-enhanced learning," *Ubiquitous learning environments and technologies*, 2015, pp. 159-180.

[10] Y. Meier, J. Xu, O. Atan, and M. van der Schaar "Predicting grades," *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 959-972, 2016.

[11] J. Xu, T. Xing, and M. van der Schaar, "Personalized course sequence recommendations," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5340-5352, 2016.

[12] T. Lu, D. Pl, M. Pl, "Contextual Multi-Armed Bandits," *AISTATS*, 2010, pp. 485-492.

[13] L. Song, C. Tekin, M. van der Schaar, "Online learning in large-scale contextual recommender systems," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 433-445, 2014.

[14] L. Li, W. Chu, J. Langford, R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," *Proceedings of the 19th international conference on World wide web. ACM*, 2010, pp. 661-670.

[15] D. Bouneffouf, A. Bouzeghoub, A. L. Ganarski, "A contextual-bandit algorithm for mobile context-aware recommender system," *International Conference on Neural Information Processing*, Springer Berlin Heidelberg, 2012, pp. 324-331.

[16] S. Bubeck, R. Munos, G. Stoltz, C. Szepesvari, "X-armed bandits," *Journal of Machine Learning Research*, pp. 1655-1695, 2011.

[17] M. G. Azar, A. Lazaric, and E. Brunskill, "Online Stochastic Optimization under Correlated Bandit Feedback," *ICML*, 2014, pp. 1557-1565.

[18] C. Tekin, M. van der Schaar, "Distributed online big data classification using context information," *IEEE Annual Allerton Conference: Communication, Control, and Computing*, 2013, pp. 1435-1442.

[19] A. Slivkins, "Contextual bandits with similarity information," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2533-2568, 2014.

[20] Y. Hou, Y. Feng, P. Zhou, J. Xu, "Supplementary for 'Course Recommendation of MOOC with Big Data Support: A Contextual Online Learning Approach'," https://www.dropbox.com/s/gom0ged3s5r382d/yifan_gc_sup.pdf?dl=0