Guess You Like: Course Recommendation in MOOCs

Xia Jing
Department of Computer Science and Technology,
Tsinghua University
jingx14@mails.tsinghua.edu.cn

Jie Tang
Department of Computer Science and Technology,
Tsinghua University
jietang@tsinghua.edu.cn

ABSTRACT

Recommending courses to online students is a fundamental and also challenging issue in MOOCs. Not exactly like recommendation in traditional online systems, students who enrolled the same course may have very different purposes and with very different backgrounds. For example, one may want to study “data mining” after studying the course of "big data analytics” because the former is a prerequisite course of the latter, while some other may choose “data mining” simply because of curiosity.

Employing the complete data from XuetangX[1], one of the largest MOOCs in China, we conduct a systematic investigation on the problem of student behavior modeling for course recommendation. We design a content-aware algorithm framework using content based users’ access behaviors to extract user-specific latent information to represent students’ interest profile. We also leverage the demographics and course prerequisite relation to better reveal users’ potential choice. Finally, we develop a course recommendation algorithm based on the user interest, demographic profiles and course prerequisite relation using collaborative filtering strategy. Experiment results demonstrate that the proposed algorithm performs much better than several baselines (over 2X by MRR). We have deployed the recommendation algorithm onto the platform XuetangX as a new feature, which significantly helps improve the course recommendation performance (+24.6% by click rate) comparing with the recommendation strategy previously used in the system.

KEYWORDS

MOOCs, Personalization, Course recommendation

ACM Reference format:
Xia Jing and Jie Tang. 2017. Guess You Like: Course Recommendation in MOOCs. In *Proceedings of WI ’17, Leipzig, Germany, August 23-26, 2017,* 7 pages.
DOI: 10.1145/3106426.3106478

1 INTRODUCTION

The rapid development of Internet results in the big problem of information overload. Recommendation technology has been widely used in Internet applications to help people find their favorite content from the huge overloaded information. For example, the famous e-commerce platforms such as amazon.com and taobao.com recommend commodities to users based on their browsing and purchase history [14]; Netflix.com uses recommendation to help people find their interested films and videos using collaborative filtering [5]; Youtube.com builds personalized homepage to show recommended videos for every user according to the playing history [8]. Recommendation system already becomes one of the most important and effective tools to help reduce the cost of information navigation.

Massive open online courses (MOOCs) become popular in recent years. Lots of MOOCs platforms have been built around the world. For example, Coursera, edX and Udacity are offering students an unprecedented opportunity to access superior courses from prestigious universities. In China, XuetangX is one of the largest MOOCs platforms, offering more than 1,000 courses and having attracted over 6,000,000 users. MOOCs not only transform traditional courses into online form, but also provide a chance to study users’ learning behaviors using big data.

In this paper, we aim to understand the purpose of online learning behaviors and study how to improve personalized course recommendation in MOOCs. Solving this problem is non-tricky and has the following challenges:

- **Sparsity**: Students of MOOCs usually do not choose many courses because learning a course is a time-consuming task. In XuetangX, for instance, statistics show that each course usually lasts for several weeks and a student enrolls only 1.3 courses on average. Traditional recommendation algorithms such as collaborative filtering who treat student-course as user-item just like in e-commerce situation may get very coarse results. Moreover, two students who enroll a same course may be interested in different parts of the course. It is unreasonable to treat them as same.
- **Anti-interest**: Enrollment (course choosing) behaviors are not just influenced by students’ interest. For example, a student would not enroll in a course which contains very similar content with another course he/she has studied before though this course matches his/her interest very much according to some metrics. Another example, if two courses have obvious prerequisite relationship such as “C++ Basic Programming" and “C++ Advanced Programming", then students may be more possible to follow the prerequisite order. It inspires us to combine the specific MOOCs situation to our recommendation system and try to find more useful features except interest.
- **Cold start**: Cold start is a classic problem in recommendation system. One common practice is using popular courses

[1] https://xuetangx.com

WI ’17, Leipzig, Germany

DOI: 10.1145/3106426.3106478

regardless of students' interest when we are short of students' behaviors. However, if we have some other information besides user activities, we may get better results.

To address these challenges, we design an algorithm framework to extract user-specific latent information from their historical access behaviors to represent their interest profile. This framework goes deep into the content which a student actually studied during his/her learning progress instead of treating a course as a indivisible item. Then we use a similarity based algorithm to utilize above interest profile. Besides, demographics and course prerequisite information are also used to better reveal users' potential choice and increase the diversity and abundance of recommendation results. The usage of demographics also helps us deal with the cold start problem.

Our framework has been deployed onto the XuetangX platform as a block named "Guess You Like". We perform both offline and online experiments on XuetangX. The offline results demonstrate the superior performance of our framework comparing with several baseline methods. Our framework also achieves a 24.5% improvement in online environment comparing with the strategy previously used in XuetangX.

The rest part of this paper will be organized as follows. Section 2 lists previous studies closely related to our work and briefly introduces our contribution. In section 3, we present the definition of our recommendation task and give our solution framework. Section 4 presents our experiment design and results with discussion. In the last section, we conclude with a summary of our work and an outlook to future work.

2 RELATED WORK

MOOCs are revolutionizing education in many respects[21, 23]. We review previous literatures about the following related topics.

2.1 Behavior Modeling

A number of studies focus on behavior modeling of students in MOOCs platform. Breslow et al.[7] study the diversity and influence of user characteristic in the first online course in edX. Guo and Reinecke[11] examine the demographic difference of navigation behavior and give their suggestions on course design. DeBoer et al.[9] study how to predict the demographic information of students using their learning behavior. Qiu et al.[16] design a model which incorporates demographic information, forum activity and learning behavior to predict the learning effectiveness. Wilkowski et al.[24] study the relation between prior skill and completion rate of students in Google MOOCs platform.

2.2 Student Engagement

Another line of research examines the engagement of students on MOOCs. Anderson et al.[2] propose a taxonomy for different engaging patterns in MOOCs. They also deploy a badge system to incent students and successfully increase their forum engagement. Dropout is a common phenomenon in MOOCs due to the unlimited web environment, which is main reflection of low engagement. Ramesh et al.[17] build a latent model for engagement using activities from forum and assignment and use the model to predict dropout. Rosé et al.[20] analysis the impact of three social factors on dropout behavior by employing a survival model. Similar method also can be found in [25]. Bayer et al.[4] also a classify model to predict dropout and school failure by exploiting more features from social information. There are also some studies try to increase user loyalty by giving suggestions on course design[12, 18].

2.3 Course Recommendation

Recommendation is a classical topic in data mining. Various algorithms including neighborhood based[10, 19, 22] such as user based collaborative filtering and item based collaborative filtering, latent interest model such as matrix factorization[13] and content aware model[3] have been used in web applications. In MOOCs, Piao and Breslin[15] propose a profile based strategy to do course recommendation in cold start situation. The proposed algorithm extracts user profile text from their LinkedIn pages and calculates the similarity with course profile text, then gives the recommendation results by the similarity. Apaza et al.[3] use LDA to train two different topic models on both college course syllabus and online course syllabus then a content based match algorithm is used to estimate the ratings from a user to all courses. Aher and Lobo[1] combine clustering and association rule mining algorithms to recommend courses using historical data.

Compared with previous studies, the main contribution of our work is that we exploit information from various sources including access history, course content, demographics and course relation. We design a uniform framework to combine them in order to make better use of all available information.

3 FRAMEWORK

In this section, we first give the definition of our recommendation problem then propose our hybrid framework with a subsection introducing our latent user interest model and a subsection introducing how we leverage background information to boost our model. At last, we introduce our online system on XuetangX.

3.1 Problem Definition

Like every classic recommendation task, there are two basic elements user and item in our course recommendation task, where a user represents a registered student and an item represents an open course in XuetangX. We use U to denote the set of users and C to denote the set of courses. Each $u \in U$ has enrolled some courses denoted by $C_u \subseteq C$ and each $c \in C$ has its enrollment set denoted by $U_c \subseteq U$. Let $E = \{(u,c) | u \in U, c \in C_u\}$ be the set of all enrollment relations. Given (U, C, E), our goal is recommending courses to a specific user u which are not in C_u. More precisely, we devote ourselves to give an ordered courses list R_u where courses are ordered by the probability user u will enroll in the future.

In our task, a course can be regarded as a set of web pages. For each user $u \in U$, we also have his/her historical web accessing logs which can be represented by a sequence of web pages with length n_u, i.e., $L_u = (d_1, d_2, ..., d_{n_u})$ which means user u accessed $d_1, d_2, ..., d_{n_u}$ successively in the past. The text content of each web page is available. Moreover, the basic demographic features such as gender, age and education level denoted by F_u are also available. Finally, our course recommendation task can be represented by:

$$f : (U, C, E, u, L_u, F_u) \rightarrow R_u$$

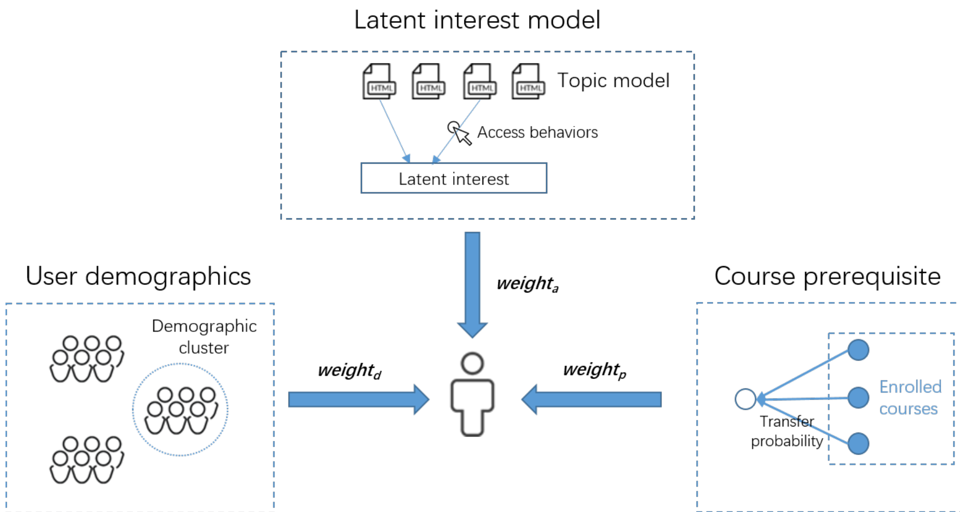

Figure 1: Overview of our method: a hybrid recommendation framework integrated leverages user access behaviors, user demographics and course prerequisite.

3.2 Interest Model

User interest modeling is the most important target of our framework. We want to build a simple but universal representation of user interest which will be useful not only in course recommendation but also in other personalization tasks. In our experiment environment XuetangX, a naive solution is directly combining user enrollment and the tags of enrolled courses. While this solution suffers from the small size of tag set and the sparsity of enrollment described in Introduction section. We emphasize that a course should not be the basic element and enrollment behavior should not be the basic relation in user interest modeling in order to avoid coarse results. In our framework, on the one hand, we utilize text content of courses instead of tags, on the other hand, we extract user interest from historical access behaviors which contain much more abundant information than enrollment behaviors. By this way, user interest will be modeled by what he/she actually accessed instead of what courses he/she enrolled. This subsection can be visualized as the top dotted box in figure 1.

In order to understand the content of the web pages in historical access logs which is closely related to the interest of students, we extract the text content and employ topic model to give a latent representation for each accessible page of XuetangX. Topic model is a series of algorithms used to discover the "topics" from a set of documents. It provides a way to define a topic as the distribution over words in a fixed vocabulary and represent a document as the distribution over topics. Topic model has been widely used in various kinds of text mining tasks.

Let D be the document set containing all pages in accessing logs. Our framework trains a topic model over D using latent Dirichlet allocation (LDA)[6] algorithm after filtering irrelevant content. So that we get a k-dimensional vector representation $\mathbf{t_d}$ for each document $d \in D$ where k is the topic number. Table 1 shows part of our topic results. Topics in table 1 have their own domains but all relate to computer science. We employ these examples to explain how

Table 1: Topic examples: Each row in table lists top 10 related words of a topic in our LDA results. Words have been translated from Chinese.

Top 10 words
iteration, list, enumeration, card, judge, successively, find, number, index, update
right child, splay, zag, parent, left child, zig, AVL, unbalance, ancestor, locality
function, pointer, array, define, data, call, point to, member, int, initialization
search, search engine, web page, query, relate, recommendation, researcher, term, keyword, link
......

topic model helps us to subdivide a coarse tag into multiple topics which provides us a way to represent user interest more detailed. By regarding the access behaviors L_u as the positive feedback from u, we employ a vector summation of $\mathbf{t_d}$ for each $d \in L_u$ to represent the latent interest of u which can be written as

$$\mathbf{p_u} = \frac{1}{n_u} \sum_{d \in L_u} \mathbf{t_d} \quad (1)$$

where p_u is the latent interest representation of u. It is easy to see that the dimension of $\mathbf{p_u}$ is also k.

The latent representation of user interest provides us a convenient way to calculate the similarity between users which is an important measurement in recommendation system. Traditional collaborative filtering (CF) algorithm also calculates the similarity while its similarity is only based on user-item matrix. Matrix factorization (MF) is another available algorithm for recommendation.

MF also proposes a latent factor space to represent both user and item but it is hard to interpret. There is a disadvantage in both CF and MF that they only use the relation information between users and items. In our framework, we combine above LDA based latent representation with user-based CF to make a better use of the content of items. The specific method is using cosine metric between users' latent interest vector to represent the similarity between them instead of dealing with it from the user-item matrix. The cosine similarity between two users u_i, u_j can be written as:

$$sim(u_i, u_j) = \frac{\mathbf{p}_{u_i}^T \cdot \mathbf{p}_{u_j}}{\|\mathbf{p}_{u_i}\| \times \|\mathbf{p}_{u_j}\|} \tag{2}$$

For the convenience of computation, we use a matrix form representation $P = \{\mathbf{p'_{u_1}}; \mathbf{p'_{u_2}}; ...; \mathbf{p'_{u_{|U|}}}\}$ to denote the interest of all users where $\mathbf{p'_u} = \mathbf{p_u}/\|\mathbf{p_u}\|$ means the normalization of $\mathbf{p_u}$. Then the similarity matrix S can be simply written as:

$$S = P^T \times P \tag{3}$$

where $S_{i,j}$ is the result of $sim(u_i, u_j)$.

Having similarity between users, we can get a weight on each $(user, course)$ pair by collaborative filtering strategy, that is:

$$weight_a(u, c) = \frac{1}{K} \sum_{u' \in U_{u,K}} sim(u, u') \times I_{C_{u'}}(c) \tag{4}$$

where $U_{u,K}$ denotes the top-K similar user set of u and $I_{C_{u'}}(c)$ is an indicator function whose value is 1 when $c \in C_{u'}$.

3.3 Background Information

To further boost performance, we try to use users' background information besides access behaviors. In this section, we will introduce how we combine the demographic information and the course prerequisite relation to our above model. User profile, such as gender, age, job, social network etc, has been used in some recommendation models especially in cold start situation. In our task, course prerequisite relation is also an important information which affects users' enrollment behaviors due to the particularity of MOOCs.

3.3.1 *Demographics.*

Demographics is an important category of information which has been widely used in user behavior modeling[9, 16] and recommendation[15]. While in MOOCs, just a few users fill in the user profile with their demographics. In our dataset, only 32.3% of users fill at least one item of demographics and 6.5% of users fill completed demographics. In order to utilize demographic information, we try to find common preference pattern in a user group with relatively small size. As shown in the left dotted box of figure 1, we employ a K-Modes algorithm to partition users into different classes based on the features extracted from users' demographic information including gender, age and education level. Each cluster has its owner preference on courses enrolling according to the enrollments of internal users. Then every user can take the average rating (value 1 for enrolling, value 0 for no enrolling) of users who belong to the same cluster as a demographics related weight on every course. We can write this weight down as:

$$weight_d(u, c) = \frac{1}{|U_{l_u}|} \sum_{u' \in U_{l_u}} I_{C_{u'}}(c) \tag{5}$$

where l_u denotes the cluster label of user u, U_{l_u} denotes the user set with label l_u and I is an indicator function.

3.3.2 *Course Prerequisite.*

Prerequisite of courses exists in both real and online education. For example, there are two courses "Data Structure I" and "Data Structure II" which have obvious prerequisite relation in XuetangX (the former is more fundamental). To measure the relation between enrollments of different courses, we define a transfer probability $tp(c_i, c_j)$ on each ordered courses pair (c_i, c_j) by counting how much percentage of users enroll course c_j after they enroll course c_i, that is:

$$tp(c_i, c_j) = \frac{|\{u | u \in U_{c_i} \bigcap U_{c_j}, t_{u,i} < t_{u,j}\}|}{|U_{c_i}|} \tag{6}$$

where $t_{u,i}$, $t_{u,j}$ respectively denote the time when u enrolls courses c_i and c_j. For example, $tp("DataStructureII", "DataStructureI")$ is 2x higher than $tp("DataStructureI", "DataStructureII")$ which indicates higher transfer tendency from "Data Structure I" to "Data Structure II".

We leverage above transfer probability by simply utilizing it as another kind of weight over $(user, course)$ pair. Each user will get a rating on a new course weighted by the summation of the transfer probability from his/her enrolled courses to the specified new course, that is:

$$weight_p(u, c) = \sum_{c' \in C_u} tp(c', c) \tag{7}$$

Looking from another point of view, proposed transfer probability can be treated as a special similarity between courses and equation 7 can be treat as a process of item based collaborative filtering.

Finally, the total weight of pair (u, c) can be written as:

$$\begin{aligned} weight(u, c) &= \alpha \times weight_a(u, c) \\ &+ \beta \times weight_d(u, c) + \gamma \times weight_p(u, c) \end{aligned} \tag{8}$$

where α, β, γ are parameters to control the proportion of weights from different sources.

The whole framework can be written as Algorithm 1.

3.4 Online System

We discuss our algorithm framework with the technical group of XuetangX after fully verification in offline dataset and design an online system for real application scenario. Unlike in offline dataset, online system should consider more about time cost and scalability.

The are two different types of tasks in our framework. One of them does not need to run frequently because the input of these tasks is relatively constant. Tasks belonged to this type include LDA training and users clustering among which LDA training is a time-consuming task. They are run in a low frequency.

Another type of task consists of procedures including generating latent user interest, calculating users' similarity and final recommending based on collaborating filtering. We run these tasks every day to catch the dynamic interest of users.

Moreover, some necessary rules are needed in online environment for better user experience. For example, courses in XuetangX may have a date bound of enrolling, so our system should filter overdue courses in the recommendation results. Another example is that a course may have different versions, so our system should always give the latest version. All of these rules can be summarized

Algorithm 1 Hybrid weight course recommendation

Require:
User set U, course set C, enrollment set E, document set D, historical access logs of each user L_u, demographics of each user F_u.
Ensure:
Recommendation results for each user R_u.

```
Train document topics t_d for each document d ∈ D by employing LDA on D;
Calculate latent user interest p_u for each u ∈ U by equation 1;
Calculate user similarity matrix by equation 3;
Train demographics based cluster for each user u ∈ U by employing K-Modes;
for each u ∈ U do
    Calculate user interest related weight, i.e. weight_a(u, c), by equation 4;
    Calculate user demographics related weight, i.e. weight_d(u, c), by equation 5;
    Calculate course prerequisite related weight, i.e. weight_p(u, c), by equation 7;
    Calculate final weight(u, c) by equation 8;
    Let R_u be the sorted list of C ordered by its weight(u, c) in descending order.
end for
```

as some rule based adjustment after running algorithm. To address this issue, we design a submodule to handle the output of the algorithm. This submodule allows us to remove or add some courses according to some predetermined rules. In addition, we also design fallback strategy for those newbie users who neither enroll any course nor have any useful profile. In this situation, we will give course recommendation by random picking up some courses from a popular courses list.

We deploy above system in January 2017 onto the homepage of XuetangX as a block named "Guess You Like". This block provides a top-5 courses recommendation according to the results of our algorithm framework. The architecture of the system is shown in Figure 2. Tasks in "low frequency" part (left top box in the figure) are run once a week to catch the variation of course content and user profile. Other tasks are run once a day to model the dynamic user interest.

4 EXPERIMENT

4.1 Dataset

The dataset used in our study is formed by offline part and online part. The offline part consists access logs and enrollment logs collected during September 2016 and October 2016. We take the data in September 2016 as the training set of our framework and the enrollment data in the first week of October 2016 as the test set to validate the quality of course recommendation. Table 2 lists the basic information of offline dataset.

We have deployed our framework onto XuetangX so that we have opportunity to observe the affect in practical environment. The online part dataset is collected from logs in the first month after system online (2017.01.10 2017.02.10).

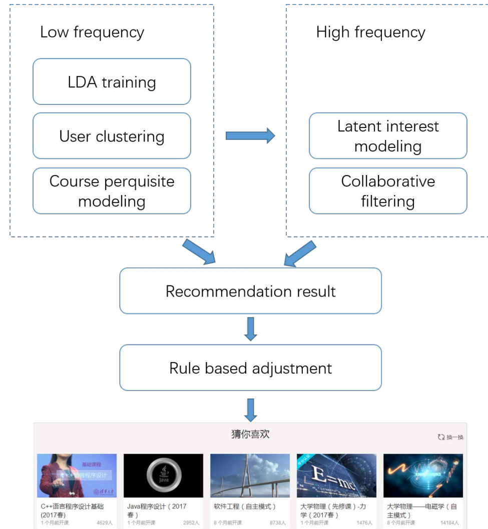

Figure 2: Architecture of online course recommendation system

Table 2: Description of offline dataset

Type	Number
# user	114303
# course [1]	1242
# access log	10096014
# document	15528

[1] This number may be different from the courses number on XuetangX official site because there may be multiple versions of courses in our dataset while XuetangX official site only shows the latest version.

4.2 Comparison

We name our methods as Hybrid Content-Aware Course Recommendation (HCACR) and Content-Aware Course Recommendation (CACR) where the latter one does not use background information which is used to the check the affect of background information. We compare our methods with traditional Item-Based Collaborative Filtering (IBCF), User-Based Collaborative Filtering (UBCF) and Random recommendation (Random).

In the offline experiment, we use Mean Reciprocal Rank (MRR) as the metric of performance. In our task, MRR can be written as follows by regarding each enrollment in test set as a query:

$$MRR = \sum_{u \in U} \sum_{c \in C'_u} \frac{1}{|C'_u|} \frac{1}{rank_u(c)} \tag{9}$$

where C'_u denotes the enrollment set of user u in test set which did not take into account the enrollments before October 2016 and $rank_u(c)$ denotes the rank of course c in the recommendation result for user u, namely R_u. If c does not exist in R_u, $rank_u(c)$ will be ∞. Larger MRR means better performance. In HCACR, each parameter is set from 0 to 1 with a step size 0.05 to find the optimal combination and finally α, β, γ are set to 0.4, 0.05, 0.55 after well tuning. Baseline

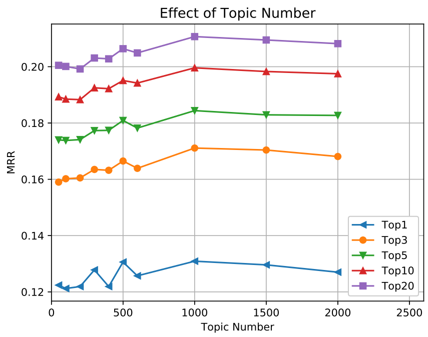

Figure 3: Performance of different topic numbers

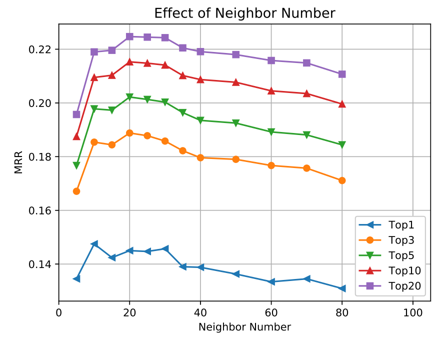

Figure 4: Performance of different neighbor numbers

methods are also tuned to their best performance to ensure fair comparison. In baseline UBCF, similarity between users u and v is measured by:

$$sim(u,v) = \frac{\sum_{c \in C_u \bigcap C_v} \frac{1}{log(1+|U_c|)}}{\sqrt{|C_u| \times |C_v|}} \tag{10}$$

which is a cosine like similarity with punishment of hot user who enrolled to many courses. The similarity between courses in IBCF is in the same form.

While in the online experiment, we use Click Through Rate (CTR) and Click Value Rate (CVR) from click to enrollment as the metrics of performance. Our framework will be compared with another recommendation strategy previously used in XuetangX platform whose results come from an operation team, namely Manual Strategy. Both these recommendation blocks are shown on the homepage of XuetangX.

4.3 Result

4.3.1 Topic Number. The number of topic is an important parameter which is hard to be determined automatically. Firstly, we try to find the best practical topic number in our topic model. In this part of experiment, topic number is varied from 50 to 2000. Figure 3 shows the performance of CACR model using different topic numbers. It shows that a topic model with about 1000 topics is best for us to model course content and user interest according to the results. So we set topic number to 1000 in our follow-up experiments. It means that both document and user interest will be represented by a vector with 1000 dimensions.

4.3.2 Neighbor Number. In collaborative filtering strategy, taking how many neighbors into account is an important problem which is sensitive to the quality the result. We investigate the performance of CACR model with different neighbor number. Figure 4 presents the comparison result. The result can be concluded as: the performance increases with the increase of neighbor number at first then decreases. According to the observation above, we pick a practical value 20 as the value of neighbor number parameter in our follow-up experiments.

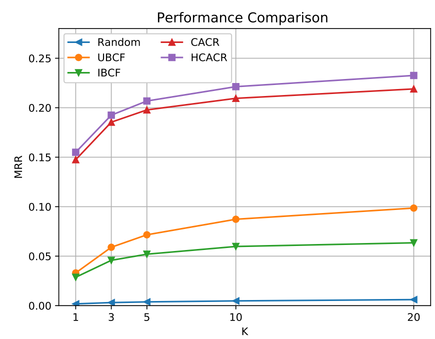

Figure 5: Performance of different algorithms in top-K courses recommendation

4.3.3 Offline Performance. Moreover, we analyze the performance of different algorithms in offline dataset. The results in Figure 5 show that our framework performs over 3x better in top-1 recommendation and over 2x better in top-10 recommendation when compared with baseline methods. It demonstrates that our algorithm works well in representing user interest and measuring similarity between users. Furthermore, the gap between CACF and HCACR shows that though there is little performance difference in top-1 recommendation, HCACR works better when results size becomes larger (+6.2% in top-20 recommendation). That is, background information will help us to mine more long-tail user interest in real situation.

Inspired by the gap between top-5 performance and top-10/top-20 performance, we add a "switch" button on our "Guess You Like" block to fit the long-tail requirement over the limited recommendation positions. It allows users to see more than 5 recommended courses by clicking this button.

4.3.4 Online Performance. We further evaluate the performance of online experiments in this section. Figure 6 shows the comparison of average CTR on each recommendation position. Our framework gets a 24.6% performance improvement versus the manual strategy on average during the first online month. Then, we analyze how much percentage of users enroll the course they chose from recommendation result, that is CVR. Statistical result shows that 19.3%

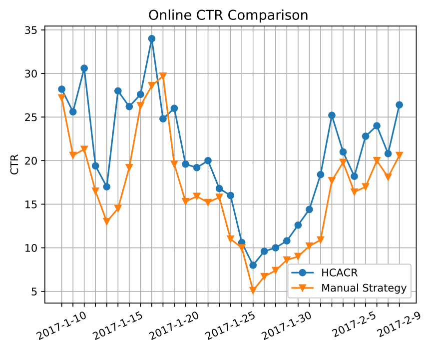

Figure 6: Performance of online system in click through rate

of users product enrollment behavior after clicking the course displayed in recommendation positions. This number is 17.2% under manual strategy. It means that our method does better in navigating users to their interested courses. Observations in this section demonstrate the validity of our method in real situation.

5 CONCLUSION

In this paper, we study the course recommendation problem in MOOCs platform. We propose a hybrid algorithm framework which combines collaborative filtering and topic model to extract user-specific latent interest from historical access behaviors and recommend based on latent interest. Background information including demographics and course prerequisite are also used to boost our method. We not only compare our framework with some traditional methods in offline dataset, but also build an online system to run our model on XuetangX platform. Result of experiments demonstrates the superior performance of our framework. In addition, we are working with XuetangX on several interesting functions, e.g. automatic QA, to increase user engagement. Our latent interest representation would be helpful in other personalization situations.

As future work, it would be interesting to boost recommendation performance by mining information from other user behaviors e.g. video behaviors. It also would be interesting to study how to recommend different kinds of content such as video and knowledge point to satisfy diverse personalized requirement in MOOCs.

REFERENCES

[1] Sunita B Aher and LMRJ Lobo. 2013. Combination of machine learning algorithms for recommendation of courses in E-Learning system based on historical data. *Knowledge-Based Systems* 51 (2013), 1–14.
[2] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2014. Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web*. ACM, 687–698.
[3] Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, and José Ochoa Luna. 2014. Online Courses Recommendation based on LDA.. In *SIMBig*. 42–48.
[4] Jaroslav Bayer, Hana Bydzovská, Jan Géryk, Tomás Obsivac, and Lubomir Popelinsky. 2012. Predicting Drop-Out from Social Behaviour of Students. *International Educational Data Mining Society* (2012).
[5] James Bennett, Stan Lanning, and others. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. New York, NY, USA, 35.
[6] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
[7] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. 2013. Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment* 8 (2013).
[8] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and others. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 293–296.
[9] Jennifer DeBoer, Glenda S Stump, Daniel Seaton, Andrew Ho, David E Pritchard, and Lori Breslow. 2013. Bringing student backgrounds online: MOOC user demographics, site usage, and online learning. In *Educational Data Mining 2013*.
[10] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.
[11] Philip J Guo and Katharina Reinecke. 2014. Demographic differences in how students navigate through MOOCs. In *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 21–30.
[12] Sarah Kellogg. 2013. Online learning: How to make a MOOC. *Nature* 499, 7458 (2013), 369–371.
[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[14] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.
[15] Guangyuan Piao and John G Breslin. 2016. Analyzing MOOC Entries of Professionals on LinkedIn for User Modeling and Personalized MOOC Recommendations. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 291–292.
[16] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. 2016. Modeling and predicting learning behavior in MOOCs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 93–102.
[17] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé III, and Lise Getoor. 2014. Learning latent engagement patterns of students in online courses. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 1272–1278.
[18] Justin Reich. 2015. Rebooting MOOC research. *Science* 347, 6217 (2015), 34–35.
[19] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 175–186.
[20] Carolyn Penstein Rosé, Ryan Carlson, Diyi Yang, Miaomiao Wen, Lauren Resnick, Pam Goldman, and Jennifer Sherer. 2014. Social factors that contribute to attrition in MOOCs. In *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 197–198.
[21] Cathy Sandeen. 2013. Integrating MOOCs into traditional higher education: The emerging "MOOC 3.0" era. *Change: The magazine of higher learning* 45, 6 (2013), 34–39.
[22] Upendra Shardanand and Pattie Maes. 1995. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 210–217.
[23] M Mitchell Waldrop. 2013. Campus 2.0. *Nature* 495, 7440 (2013), 160.
[24] Julia Wilkowski, Amit Deutsch, and Daniel M Russell. 2014. Student skill and goal achievement in the mapping with google MOOC. In *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 3–10.
[25] Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rosé. 2013. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, Vol. 11. 14.