

# Question Recommendation with Constraints for Massive Open Online Courses

Diyi Yang  
Language Technologies  
Institute  
Carnegie Mellon University  
Pittsburgh, PA, 15213  
diyiy@cs.cmu.edu

David Adamson  
Language Technologies  
Institute  
Carnegie Mellon University  
Pittsburgh, PA, 15213  
dadamson@cs.cmu.edu

Carolyn Penstein Rosé  
Language Technologies  
Institute  
Carnegie Mellon University  
Pittsburgh, PA, 15213  
cprose@cs.cmu.edu

## ABSTRACT

Massive Open Online Courses (MOOCs) have experienced a recent boom in interest. Problems students struggle with in the discussion forums, such as difficulty in finding interesting discussion opportunities or attracting helpers to address posted problems, provide new opportunities for recommender systems. In contrast to traditional product recommendation, question recommendation in discussion forums should simultaneously consider constraints on both students and questions. These considerations include (1) *Load Balancing* - students should not be over-burdened with too many requests; and (2) *Expertise Matching* - students should not be requested to address problems they are not capable of addressing. In this work, we formulate a novel constrained question recommendation problem to address the above considerations. We design a context-aware matrix factorization model to predict students' preferences over questions, then build a max cost flow model to manage the constraints. Experimental results conducted on three MOOC datasets demonstrate that our method significantly outperforms baseline methods in optimizing overall forum welfare, and in predicting which specific questions students might be interested in.

## Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval—*Information Filtering*

## Keywords

Constrained Question Recommendation; Massive Open Online Courses; Load Balance; Expertise Matching

## 1. INTRODUCTION

Massive Open Online Courses (MOOCs) have drawn a great deal of attention recently. MOOC platforms, such as Coursera<sup>1</sup> and EdX<sup>2</sup>, have demonstrated extremely high

<sup>1</sup><https://www.coursera.org/>

<sup>2</sup><https://www.edx.org/>

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 the author(s) 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](mailto:permissions@acm.org).  
*RecSys '14*, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.  
Copyright 2014 ACM 978-1-4503-2668-1/14/10 ...\$15.00.  
<http://dx.doi.org/10.1145/2645710.2645748>.

student registration rates, with equally dramatic rates of attrition. MOOC discussion forums, where course instructors and students ask questions, discuss ideas, provide help to or even socialize with other students, face a similar problem, in that the number of questions asked in a forum increases rapidly, while around half of the posted questions are never resolved. Discussion that is productive for learning must be accessible and engaging to the students involved [4]. With thousands of questions for thousands of students to sift through, the challenge is in directing students to accessible, interesting topics, while simultaneously possibly helping unanswered questions get answered. Recommender systems, which have proven to be powerful in addressing similar problems [10], can be utilized to suggest to each student a short list of questions that they might be interested in. Students will be able to find questions that match their interests, and more otherwise unanswered questions will be addressed by students capable of answering them.

Traditional product recommendation [12] focuses on the extent to which a product will be desirable to users. Additionally, there have been some explorations of question recommendation in general discussion forums [22]. However, in contrast to existing work, question recommendation for MOOC discussion forums must address reciprocal constraints representing the needs of both the students who post questions and the needs of the students who might be able to offer some help or support in response.

In particular, the related constraints include (1) *Load Balancing*, respecting that students have limited time for participating in the discussion; (2) *Expertise Matching*, students have differing levels of expertise, and some questions will therefore be too difficult, whereas some are so simple that they do not make use of the students' particular skills and abilities. These facets of question recommendation have not been fully addressed in existing work. Thus we propose this *Constrained Question Recommendation* (CQR) task to take into account load balancing and expertise matching constraints. The focus of our work is on the solution formalism, which can be tailored to alternative operationalizations of skills, abilities, and difficulty levels. To the best of our knowledge, this is the first work describing a question recommendation problem with constraints. Our main contributions in this work are as follows:

- We propose a question recommendation task in the MOOC discussion forums, and formulate a constrained question recommendation problem, which considers constraints on both students and questions.

- We design a context-aware matrix factorization combined with rich contextual information to predict students’ interests over questions in the MOOC discussion forum, then build a max cost flow model to fit the constraints and optimize the overall welfare of the forum community.
- Experiments conducted on three different MOOC datasets demonstrate the effectiveness of our model over several baselines.

This paper is organized as follows. In Section 2, we present related work. Section 3 focuses on the problem’s motivation and definitions. Section 4.1 gives the context-aware relevance prediction model. Section 4.2 discusses constraint satisfaction, where we adopt a max cost flow model to satisfy constraints. Experimental setup and discussion of results are presented in Section 5. We conclude our work and point out the implications in Section 6.

## 2. RELATED WORK

The field of recommendation can be categorized into two kinds of work. The first is recommendation of products [10], a one-way recommendation where the focus is how desirable an item will be to users. This includes recommending books [12], music [23], Tweets [5], and movies. For instance, Koren et al. [9] proposes a combination of latent modeling and similarity based matching between users and movies to suggest movies that fit users’ needs. The second is social recommendation [21, 11] which requires the match should be attractive in both directions, such as friend recommendation in social media like Facebook and online dating services [3]. For example, Pizzato et al. [15] propose a reciprocal recommender system for online dating where profile messages are utilized to learn users’ interests and consider also preferences of the recommendation recipients.

Traditional question recommender systems [16], such as those that recommend questions to potential answerers in Yahoo! Answers, are mostly focused on approximating the affinity between users and questions [22]. Hu et al. [6] introduces a user modeling method that estimates the interests and professional areas of each user in order to generate a suitable user set to answer a given question. Similarly, Kabutoya et al. [8] uses the question and answer histories of each user, combining collaborative filtering schemas and content based filtering schemas to recommend questions. Most existing question recommenders can be categorized into the product recommendation field because they do not address the issue of reciprocity, or the potential for constraints associated with users and questions. Especially for question recommendation in the context of MOOC forums, recommendations should be based not only on what would be attractive to the recipient [19], but also on how their expertise and question-answering capacity can be balanced against the load of available questions [20].

In this work, we seek to develop a recommender algorithm to suggest questions to users, respecting both their level of expertise and capacity for taking on work (e.g., load balancing). There are many effective recommendation techniques, including content based filtering [13] and collaborative filtering, etc. Collaborative filtering can be categorized as model based [10] or neighborhood based methods [9]. Matrix factorization models [10], are one of the most successful model-based approaches to predict

the affinity between users and items. Rendle et al. [17] proposes a context-aware factorization machine. Such matrix factorization models enable us to integrate rich contextual information, such as implicit feedback [9, 23] and user attribute information [1, 5] for predicting matches. However, these approaches do not address our concerns with respect to load balancing.

## 3. OVERVIEW

In this section, we present our motivation for conducting the constrained question recommendation task and define related terms that will be used in later sections.

### 3.1 Motivation

Engagement in course-related discussions can have a positive influence on learning [4]. In the MOOC context, where the interaction or guidance of instructors is limited and dropout is very high, it is critical to increase the participation and engagement of students in the course [7]. Guiding students to topics that suit their interests among a large set of questions is challenging. This problem is even more pronounced for students who post questions seeking help as well as the would-be helpers who are unable find the questions they’re most able to respond to [18]. Suggesting appropriate questions to students is not simply determined by matching their interests, but is also subject to constraints on the students’ time and the questions’ difficulty.

Traditional question recommendation approaches mostly focus on figuring out the best matches of topic-interest between student question pairs, ignoring practical constraints like load balancing and expertise matching. In fact, ignoring constraints is likely to cause a long tail problem [14] - for example, a large number of students may be directed to just a few popular questions, and too many questions may be recommended to a small set of high-expertise students. However, there are limits on how many questions any one student can answer, and such imbalanced recommendations may leave other students idle. Further, popular questions might not need such a large number of students to offer help. These imbalances will not only lead to a waste of valuable human resources, but also leave many un-popular questions deserted, and their askers at a loss for help.

In this work, we focus on how to recommend a list of questions to students on the given set of constraints on both students and questions (here expressed as load balancing and expertise matching). Our constrained question recommendation framework guides appropriate students to questions in a discussion forum such that (1) students will benefit from their interaction with questions that are interesting to them, (2) questions are discussed by students with sufficient expertise, and (3) students are not over-burdened with too many requests. By taking such constraints into account, we produce a set of recommendations that is mutually beneficial to both the community of question-askers and potential question-answerers. To the best of our knowledge, this is the first work to address question recommendation with constraints on both students and questions.

### 3.2 Definitions

Existing recommender systems usually produce “relevance” scores to rank the pool of available questions for a given user, and offer the highest-ranked ones as recommendations. Here, we make a similar choice (although

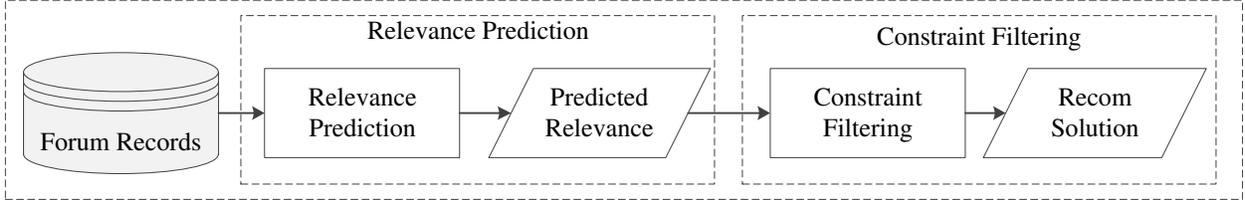


Figure 1: The Framework of Constrained Question Recommendation

subject to constraints) and denote the “relevance” score of question  $q$  to student  $u$  as  $r_{u,q} \in [0, 1]$ . For a specific student  $u$ , the larger  $r_{u,q}$ , the more relevant the question is. In what follows, we introduce our Constrained Question Recommendation (CQR) problem. To model load balancing and expertise matching, we first define several values associated with students and questions, including student capacity, student capability, and question hardness.

**DEFINITION 1.** *Capacity* of student  $u$  is a load balance range  $[L_u, R_u]$ , i.e. the number of questions that would be comfortable for a student  $u$  should be no less than  $L_u$  and no more than  $R_u$ .

**DEFINITION 2.** *Capability*  $B_u$  is the capability/expertise of student  $u$  to solve problems.

**DEFINITION 3.** *Hardness*  $H_q$  stands for how difficult the question  $q$  is, or how much expertise it requires.

The *Load Balancing* constraint requires that the number of questions recommended to students be in the range of the students’ Capacity, and that the number of students we guide to a given question should not be too few or too many; while *Expertise Matching* means that among students we guide to a specific question, there should be at least one student whose Capability is equal to or larger than the Hardness of that question. Those constraints enable us to give the formal definition of CQR. CQR is a general question recommendation framework that recommends a list of questions to students, subject to load balancing and expertise matching. It is worth mentioning that existing question recommender systems are a special case of CQR that ignores all constraints.

## 4. FRAMEWORK

Figure 1 presents the overview of our constrained question recommendation framework. To address this constrained question recommendation problem, in the first step, we design a context-aware matrix factorization model to predict students’ preferences over questions (relevance prediction); in what follows, we build a max cost flow network using the predicted relevance scores to meet the constraints (constraint filtering) and show how to solve this model to provide the final recommendation list.

### 4.1 Context-Aware Relevance Prediction

In this section, we briefly present our context-aware relevance prediction model, which can be fit into a class of popular matrix factorization models [17]. The predicted relevance scores are the basis of the subsequent constraint filtering phase. The relevance matrix between students

and questions is denoted as  $R$  with entry  $r_{u,q}$  representing the relevance between student  $u$  and question  $q$ . The relevance score  $r_{u,q}$  is 1 if and only if student  $u$  has joined the discussion of question  $q$ . 0 entries are obtained by negative sampling as described below where we specify our experimental settings. This context-aware relevance prediction model mainly exploits three aspects of contextual information as follows, i.e. *Student Features*, *Question Features*, and finally *Implicit Feedback*.

**Student Features** include statistical features related to individual students, such as their posting and replying activities in the discussion forum. Here, **Question Count**, denoted as  $\phi_u$ , represents how many unique questions the student has participated in prior to the current week and could indicate that student’s typical participation level, which could then be used as a long term activity level expectation. **Previous Count**  $\theta_u$ , is calculated to indicate how many posts the student has made in the last week, and can be regarded as a short term or local activity level. **Cohort**  $\gamma$ , representing the week in which this student registered for the course, could be treated as a proxy for the level of motivation (i.e. early course registration is typically associated with above average commitment to the course) [25]. This can be formally described as:

$$\hat{r}_{u,q} = bias + (P_u + \phi_u \Phi + \theta_u \Theta + \Gamma_\gamma)^T Q_q$$

Here,  $\hat{r}_{u,q}$  is the predicted score of  $u$  on  $q$ ; *bias* represents various biases related to student and question, and  $P_u$  and  $Q_q$  are latent vectors associated with  $u$  and  $q$ .  $\Phi$ ,  $\Theta$  are each one-dimensional feature vectors and  $\Gamma_{(\cdot)}$  is a feature vector with dimension of different cohort numbers.

**Question Features** are those statistical features related to questions [24]. **Reply Number**  $\delta_q$  is utilized to evaluate how many replies and comments a question  $q$  has garnered until now. Similarly, **Question Length**  $l_q$  is designed to estimate how much effort and expertise is required to answer the current question, which is computed as the total number of words appearing in that question, including all its replies and comments. We formulate this as:

$$\hat{r}_{u,q} = bias + P_u^T (Q_q + \delta_q \Delta + l_q L)$$

where,  $\Delta$  and  $L$  are two one-dimensional features associated with Reply Number and Question Length respectively.

**Implicit Feedback** [10] represents whether students express their implicit preferences over questions through their participation in other discussions(or not). Denote  $U(q)$  as the set of students participating in question  $q$  and  $\varphi_{(v)}$  as the predicted preference  $v$  of  $u$ , then we have:

$$\hat{r}_{u,q} = bias + P_u^T (Q_q + \frac{1}{\sqrt{|U(q)|}} \sum_{v \in U(q)} \varphi_v)$$

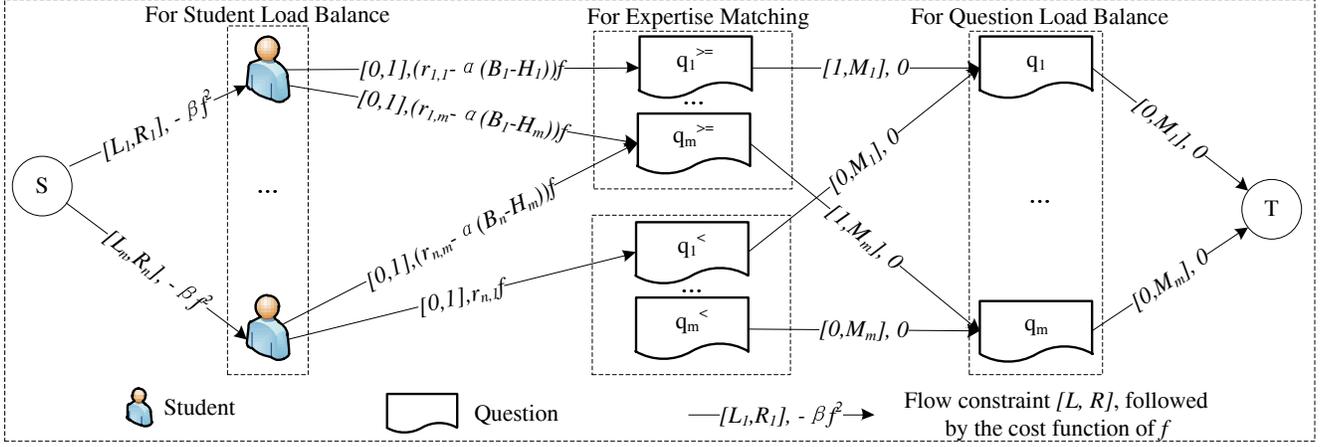


Figure 2: The Max Cost Flow Network Model

Taking those three feature sets into account, we formulate our context-aware relevance prediction model as below:

$$\hat{r}_{u,q} = bias + (P_u + \phi_u \Phi + \theta_u \Theta + \Gamma_\gamma)^T \left( Q_q + \delta_q \Delta + l_q L + \frac{1}{\sqrt{|U(q)|}} \sum_{v \in U(q)} \varphi_v \right) \quad (1)$$

The relevance score should be in the range  $[0, 1]$  and the observed matrix is binary, thus it is common to use a sigmoid function to ensure the final prediction in the range and logistic loss as the loss function:

$$f(\hat{r}_{u,q}) = \frac{e^{\hat{r}_{u,q}}}{e^{\hat{r}_{u,q}} + 1} \quad (2)$$

$$l(\hat{r}_{u,q}, r_{u,q}) = r_{u,q} \ln(1 + e^{-\hat{r}_{u,q}}) + (1 - r_{u,q}) \ln(1 + e^{\hat{r}_{u,q}}) \quad (3)$$

To decrease the risk of overfitting, the  $l_2$ -norm is chosen as the regularization term and  $\lambda$  is the learning rate. The parameters of this model can be learned by the feature based matrix factorization framework [17], which is a class of state-of-the-art methods that can utilize contextual information and implicit feedback [9].

$$L = \sum_{u,q} l(\hat{r}_{u,q}, r_{u,q}) + \frac{1}{2} \lambda (\|P\|^2 + \|Q\|^2 + bias^2) \quad (4)$$

## 4.2 Max Cost Flow Constraint Filtering

In this section, we firstly formalize the CQR task as an integer linear programming problem, which is equivalent to a max cost flow model and then we describe how we solve this max cost flow model.

### 4.2.1 Integer Linear Programming Form of CQR

Before giving a detailed description of the max cost flow model, we start with an introduction to our notation. Here, we denote  $f_{u,q} \in \{0, 1\}$  as an indicator of whether the question  $q$  is recommended to the student  $u$ . Recall the definitions in Section 3.2,  $r_{u,q} \in [0, 1]$  is the relevance score between the question  $q$  and the student  $u$ . We formulate the constraints by using  $f_{u,q}$  in Section 3.2, and we obtain the rephrased representation as below:

- **Load Balancing:** For each student  $u$ ,  $L_u \leq \sum_q f_{u,q} \leq R_u$  must be satisfied - we can only recommend a proper amount of questions to a student; Additionally, the load  $\sum_q f_{u,q}$  of student  $u$  should be minimized. Each question  $q$  should satisfy  $0 \leq \sum_u f_{u,q} \leq M_q$ . Here,  $M_q$  means that each question has a limit on the number of students who can be invited to participate. If there is no constraint,  $M_q$  can be  $+\infty$ .
- **Expertise Matching:** For each question  $q$ , the constraint  $\exists u, B_u \cdot f_{u,q} \geq H_q$  must be satisfied - at least one student  $u$  that we guide to question  $q$  should have an expertise no less than that which  $q$  requires<sup>3</sup>. Besides, term  $\sum_u \sum_q \mathbb{I}(B_u \cdot f_{u,q} > H_q) (B_u - H_q)$  should be minimized, which enables us to avoid wasting human resources by assigning overly-qualified students to answer questions. The intuition is that we should guide students to proper questions, and avoid assigning tasks that are not challenging.

Taking those constraints into consideration, we then can give a mathematical formulation of our CQR problem. Here,  $U$  and  $Q$  are student and question sets respectively.  $\mathbb{I}(\cdot)$  is an indicator function.  $r_{u,q}$  is generated by context-aware matrix factorization model.  $\alpha, \beta$  are parameters used to balance the importance of different components.

$$\begin{aligned} & \max \sum_{u,q} f_{u,q} \cdot r_{u,q} - \beta \cdot \sum_u \left( \sum_q f_{u,q} \right)^2 \\ & \quad - \alpha \cdot \sum_q \left( \sum_u \mathbb{I}(B_u \cdot f_{u,q} > H_q) (B_u - H_q) \right) \\ & \text{subject to } \forall u \in U, q \in Q, f_{u,q} \in \{0, 1\} \\ & \quad L_u \leq \sum_q f_{u,q} \leq R_u, 0 \leq \sum_u f_{u,q} \leq M_q \\ & \quad \forall q \in Q, \exists v \in U, B_v \cdot f_{v,q} \geq H_q \end{aligned} \quad (5)$$

<sup>3</sup>Here, we want to help each question get response/feedback and do not allow some questions to be left with no reply.

## 4.2.2 Max Cost Flow Model and Algorithm

To maximize the objective function in Equation 5, we construct a concave cost network illustrated in the graphical representation in Figure 2. Conducting max cost flow on this network gives an optimal solution for the integer linear programming problem in Equation 5.

**THEOREM 1.** *The cost network constructed in Figure 2 is equivalent to the problem defined in Equation 5.*

**PROOF.** First, the maximum concave cost flow (MCCF) problem can be formulated as the following optimization problem:

$$\begin{aligned} & \max \sum_{(a \rightarrow b) \in E(G)} Cost_{a \rightarrow b}(Flow_{a \rightarrow b}) \\ & \text{subject to} \\ & \forall a (\neq S, T) \in V(G), \sum_b Flow_{a \rightarrow b} = \sum_b Flow_{b \rightarrow a} \\ & L_{a \rightarrow b} \leq Flow_{a \rightarrow b} \leq R_{a \rightarrow b} \end{aligned} \quad (6)$$

Here, the  $Flow_{a \rightarrow b}$  is the flow from node  $a$  to  $b$ ,  $Cost_{a \rightarrow b}$  is the concave cost function on the edge. In our network, there are only (negative) square functions, linear functions, and constant functions. Then, if we set  $f_{u,q}$  by the flows in the network as below:

$$f_{u,q} = \begin{cases} flow_{u \rightarrow q} \geq, & \text{if } B_u \geq H_q \\ flow_{u \rightarrow q} <, & \text{if } B_u < H_q \end{cases}$$

Then, the objective in MCCF is equal to Equation 5. After that, we consider the remaining constraints one by one.

- The **Load Balance** of students is bounded by the capacity constraints  $[L_u, R_u]$  between the source node  $S$  and the student nodes  $u$  with penalty cost  $-\beta \cdot f^2$ ; the question constraints are limited by the flow constraints  $M_q$  between question nodes  $q$  and the sink node  $T$ .
- The lower bound of the flow constraints between nodes  $q^{\geq}$  and question nodes  $q$  guarantees **Expertise Matching** constraint, i.e. there is at least one high expertise student guided to question  $q$ .

In summary, all constraints in CQR problem are satisfied in the network flow model and these constraints are the only limits in the network. Thus the MCCF problem defined here is equivalent to the CQR problem. Then we can formally build the network as shown in Figure 2, solve the MCCF problem by computing the maximum cost flow of network [2], and get the final recommendation plan by examining flows between  $u$  and the corresponding  $q^{\geq}$  or  $q^<$ .  $\square$

## 5. EXPERIMENT

In this section, we present a set of experiments, and a discussion of their results. We start with an introduction to the three MOOC datasets, followed by experiments in our framework and a comparison of their performance against several baselines. We also explore some practical tradeoffs by investigating how the number of relevant edges used in the optimization of the objective function affects the overall recommendation performance.

Dataset	Student	Question	Candidate Pairs
Accountable	1,148	511	582,945
Fantasy	770	603	460,448
Python	3,633	3,079	11,172,604

**Table 1: Statistics on Datasets**

## 5.1 Dataset

Our experiments are conducted on the discussion forums from three different types of courses from the Coursera MOOC platform <sup>4</sup>, (1) ‘Accountable Talk: Conversation that works’, hereafter shortened to ‘Accountable’, is a professional development course for teachers; (2) ‘Fantasy and Science Fiction: the human mind, our modern world’, shortened to ‘Fantasy’, is a course about literature writing and discussion; (3) ‘Learn to Program: The Fundamentals’, shortened to ‘Python’, teaches Python programming skills. Summary statistics are presented in Table 1. For our purposes, a question is defined as a thread starter post asking a question or requesting help. In our datasets, most posts (around 82%) are about proposing and resolving problems - since the Coursera discussion forums mainly focus on Lectures, Exercises, Assignments and Exams. Thus, these datasets are quite appropriate for evaluating our CQR problem.

## 5.2 Relevance Prediction Discussion

In this part, we firstly explain our experimental settings and evaluation metrics. Then we describe and discuss the comparison performance of our model with baselines.

### 5.2.1 Settings and Metrics

To evaluate the performance of relevance prediction, we randomly divide the observed records into train and test sets, with 70% and 30% of records respectively. The ground truth only contains student participation history, namely whether the students participated in a question discussion ( $r_{u,q} = 1$ ), denoted as positive samples. To simulate a realistic scenario, we randomly generate negative examples ( $r_{u,q} = 0$  entries) for each positive instance by selecting threads the user did not participate in. Specifically, for each  $r_{u,q} = 1$ , we sample 5 different questions  $p$  such that  $r_{u,p}$  is not observed, and set  $r_{u,p} = 0$ . In the context of question recommendation, the relevance score is binary and we use Mean Average Precision (MAP) [5] (with 95% confidence interval) as the evaluation metric. MAP@K focuses on the first top  $K$  questions in the recommended list and we use MAP@1, MAP@3, and MAP@5 for the comparison.

### 5.2.2 Recommendation Performance

We proposed two baselines to compare the performance with the context-aware matrix factorization **CAMF**, in order to validate that we can provide accurate relevance score for constraint filtering modeling. Baseline **Popularity** is a simple method in which only the popularity of questions is considered. Baseline **BasicMF** is the plain matrix factorization model without incorporation of contextual features. The comparison results are summarized in Figure 3. From this figure, we can observe that, 1) BasicMF works

<sup>4</sup>Permission to conduct research on Coursera datasets was provided by Coursera.

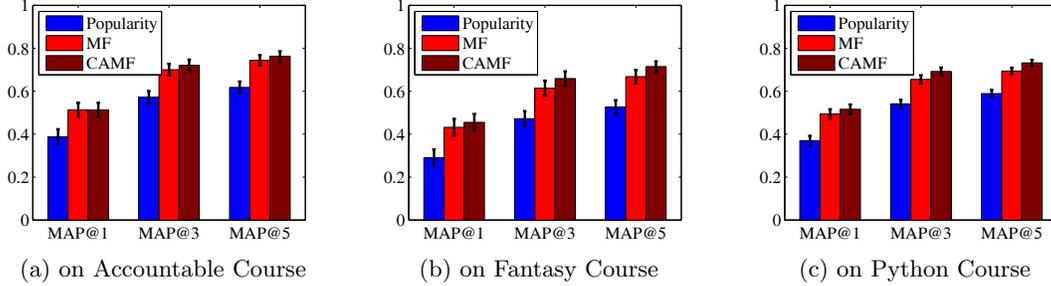


Figure 3: MAP Performance Comparison on Three MOOC Forums

better than Popularity, because it identifies the latent match between student preferences and question properties; 2) CAMF archives the best performance among three models due to consideration of MOOC specific contextual features; 3) The performance of MAP@1 achieves 0.513 on the Accountable Course, 0.455 on the Fantasy Course, and 0.516 on the Python Course, which guarantees quite accurate input for constraint filtering.

### 5.3 Constraint Filtering Discussion

We begin the introduction to this part with metrics and baselines of constraint filtering. In what follows, we present the performances of our CQR framework applied to predicted scores from the earlier relevance prediction part; Further exploration on how filtering relevant edges affects the overall community benefit is also discussed.

#### 5.3.1 Metrics and Baselines

There is no widely used standard metric to evaluate our CQR problem. For possible comparison, we design three evaluation metrics. Two types of coverage: Student Coverage (SC) and Question Coverage (QC) i.e. how many questions/students are recommended to a student/question on average, respectively. These are used to reflect the load balance statistics of students and questions.

$$QC = \frac{1}{|Q|} \sum_u \sum_q f_{u,q}, SC = \frac{1}{|U|} \sum_q \sum_u f_{u,q}$$

Overall Community Benefit (OB) stands for the objective function value defined in Equation 5, representing the sum of relevance scores we achieve excluding the loss caused by violation of recommendation constraints. Higher OB means better performance and is our most important evaluation metric. Nevertheless, it is important to consider the three metrics together in order to evaluate model performance, i.e. the community spends SC effort and generates QC activity on thread in order to achieve OB welfare. The student question pairs we use here are all unobserved pairs with relevance scores predicted by CAMF in the first phase. To compare the effectiveness of our CQR framework (with max concave cost flow method denoted as **MCCF**), we propose three baselines as below:

- **Student-Question**<sup>5</sup>: First, select top  $R_u$  relevant questions for each student  $u$ . Then among those selected student-question pairs, choose the top  $M_q$  students for each question.

<sup>5</sup>For the first two baselines, we ignore the Expertise Matching constraint and the lower bound of Load Balance

Method	SC	QC	OB
Question-Student	2.986	1.329	1342
Student-Question	4.810	2.141	1715
Greedy	9.977	4.441	2534
MCCF	8.697	3.871	<b>2897</b>

Table 2: Community Benefit on Accountable Course

Method	SC	TC	OB
Question-Student	2.419	1.683	1120
Student-Question	3.668	2.873	1376
Greedy	7.745	6.065	2324
MCCF	6.715	5.258	<b>2652</b>

Table 3: Community Benefit on Fantasy Course

- **Question-Student**: Similar to Student-Question, instead we first select the top  $M_q$  relevant students for each question. Then among those selected pairs, choose the top  $R_u$  questions for each student  $u$ .
- **Greedy**: Based on the decreasing order of relevance score, first traverse this order and add one high expertise student to each question; in a second pass, guarantee the lower bound of each student in terms of question assignment; in the third pass, for each student consider the tentatively assigned pairs ordered by decreasing relevance, and retain the pairs that meet the constraints.

#### 5.3.2 Constraint Approximation

The approach we propose in this paper does not make specific assumptions about the constraints on specific students' capabilities or capacity, or specific question difficulty. Rather, we present a parameterized approach that can be used with any desired method for estimating these specific constraints. The metadata about student and question constraints are not explicitly expressed in the forum data we have access to. For the experiments presented, we approximated the implicit capability and capacity of students and the hardness of questions based on available forum data, just as an example of one way these might be formulated. In particular, the hardness  $H_q$  of a question  $q$  is defined here as the length of that question in terms of number of words. The underlying assumption is that longer questions are more complex, and thus require higher expertise in order to offer effective help. However, we acknowledge the limitations of such an operationalization and do not make any formal claims about its adequacy. The

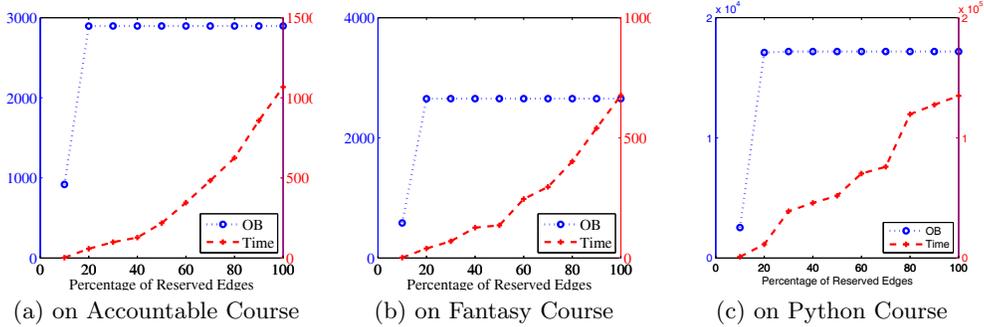


Figure 4: Overall Benefit and Running Time Changes over Edge Ratio

Method	SC	QC	OB
Question-Student	1.657	1.405	5017
Student-Question	2.201	1.865	6089
Greedy	6.950	5.890	13712
MCCF	6.134	5.189	<b>17185</b>

Table 4: Community Benefit on Python Course

upper bound  $R_u$  of capacity of  $u$  is defined as how many questions this student has participated so far in the forum; we set the lower bound  $L_u$  as 1, since every student needs at least one question to be actively involved. To capture the capability  $B_u$  of student  $u$ , we formulate it as  $B_u = \max\{H_q | q \in T(u)\}$ , i.e. the expertise  $B_u$  of student  $u$  is the maximum hardness of all questions he/she participated in at earlier time points. Here,  $T(u)$  is the question set that student  $u$  has participated in. Various approximation strategies can be adopted in our framework to function within realistic scenarios. We leave this operationalization and validation work to subsequent research.

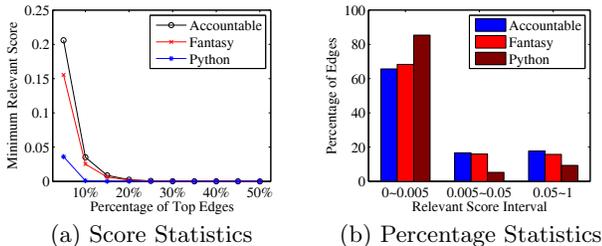


Figure 5: Why top  $\sim 20\%$  edges are enough for OB

### 5.3.3 Overall Community Benefit Comparison

In this section, we compare our max cost flow model with three baselines. Results on three datasets are listed in the Table 2, Table 4, Table 3. Empirically, we set  $\alpha$  as 0.005 and  $\beta$  as 0.001. From the three tables, we can observe that the Student-Question and Question-Student baselines have consistently lower SC, QC and OB compared to the two others. This is because in the first step in both baselines, traditional top-k selection is applied such that many students are directed to a small set of popular questions and a lot of popular questions attract almost all attention of students. Therefore, the coverage is low, which leads to the low overall benefit. That is why we consider constraints here: traditional top-k question recommendation

results in considerable waste of resources in its assignment. MCCF achieves an OB value of 2652 on the Fantasy course, 327 higher than the OB of Greedy; the SC and QC of MCCF are lower than Greedy, which indicates that our MCCF achieves a significantly higher overall benefit to the community with less work on average per participant, and less extraneous activity on threads. This conclusion is consistent across three datasets.

### 5.3.4 Edge Ratio Exploration

As shown above, the max cost flow model achieves better performance in optimizing OB under the constraints of students and questions. To explore how many edges we should reserve in the constraint filtering phase, we vary the percentage of top relevant edges in the network per each student and measure the OB obtained. That is, we use the top  $x\%$  most relevant questions for each student to run CQR model. The machine we used is Intel(R) Core i7-4700MQ with CPU @2.40GHz. Each subfigure in Figure 4 illustrates two curves; the upper curve represents how OB changes as the edge ratio varies. The lower curve shows how the running time of the corresponding model changes as the edge ratio varies. Based on this we conclude that the top 20% of relevant edges are sufficient to gain a relatively satisfying OB. For example in Figure 4(a), the OB is 2897 if we use the top 100% edges; but if we only keep the top 20% edges, we achieve an OB of 2894. The two OB metrics has no big difference but the later model only needs 57 seconds, which is only 5% of original running time. The reason why OB is quite low when we use top 10% edges is because there is no feasible solution to the CQR problem under this number of edges.

We also conduct statistics to analyze why around 20% edges are enough to get a satisfying OB. Figure 5(a) presents minimum relevance scores of different top  $x$ , and Figure 5(b) gives the statistics of the percentage of edges falling in different relevance score bins. The minimum relevance scores after top 20% edges are almost zero, and relevance scores in the range  $[0, 0.005]$  accounts for around 80% of all records. Edges with lower relevance scores contribute less to OB. Based on Figure 4, we conclude that using around top 20% edges for our model can achieve high OB and need fair time cost. Thus, this top edge ratio and the minimum relevance score exploration can help speed up our model.

## 6. CONCLUSION

In this paper, we have formulated the constrained question recommendation (CQR) problem that simultaneously

considers potential load balancing and expertise matching constraints. Our framework employs a novel combination of (1) a context-aware matrix factorization model to predict students' preferences over questions; (2) and a max cost flow model to optimize community benefit under multiple constraints. Experimental results show that our method is significantly better than proposed baselines. It is worthwhile to mention that we could adopt any state-of-the-art strategies to predict the relevance scores and conduct constraint approximation using them. However, since metadata about constraints including student capability, student capacity and question difficulty are not explicitly expressed, how to accurately approximate such constraints remains a challenge for future work. In the future, we plan to conduct online deployment studies in MOOC platforms by using our constrained question recommendation framework to test its effect in practice. In addition, we will also consider how to generalize this recommendation with constraints problem to other contexts and address scalability issues with the constraint satisfaction.

## ACKNOWLEDGEMENT

This research was supported by NSF Grant IIS-1320064. The authors want to thank Tianqi Chen, Weinan Zhang, Jingbo Shang, Miaomiao Wen for their helpful discussions and the anonymous reviewers for their valuable comments.

## 7. REFERENCES

- [1] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 19–28, New York, NY, USA, 2009. ACM.
- [2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice hall, 1993.
- [3] M. S. Amin, B. Yan, S. Sriram, A. Bhasin, and C. Posse. Social referral: Leveraging network connections to deliver recommendations. In *Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12*, pages 273–276, New York, NY, USA, 2012. ACM.
- [4] R. M. Carini, G. D. Kuh, and S. P. Klein. Student engagement and student learning: Testing the linkages\*. *Research in Higher Education*, 47(1):1–32, 2006.
- [5] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 661–670, New York, NY, USA, 2012. ACM.
- [6] D. Hu, S. GU, S. Wang, L. Wenyin, and E. Chen. Question recommendation for user-interactive question answering systems. In *Proceedings of the 2Nd International Conference on Ubiquitous Information Management and Communication*, ICUIMC '08, pages 39–44, New York, NY, USA, 2008. ACM.
- [7] R. Junco. The relationship between frequency of facebook use, participation in facebook activities, and student engagement. *ComputersEducation*, 58(1):162–171, 2012.
- [8] Y. Kabutoya, T. Iwata, H. Shiohara, and K. Fujimura. Effective question recommendation based on multiple features for question answering communities. In *ICWSM*, 2010.
- [9] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.
- [10] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.
- [11] J. Kunegis, G. Gröner, and T. Gottron. Online dating recommender systems: The split-complex number approach. In *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*, pages 37–44. ACM, 2012.
- [12] D.-R. Liu and Y.-Y. Shih. Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences. *Journal of Systems and Software*, 77(2):181–191, 2005.
- [13] P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer, 2011.
- [14] Y.-J. Park and A. Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18. ACM, 2008.
- [15] L. Pizzato, T. Rej, T. Chung, I. Koprinska, K. Yacef, and J. Kay. Reciprocal recommender system for online dating. In *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10*, pages 353–354, New York, NY, USA, 2010. ACM.
- [16] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th international conference on World wide web*, pages 1229–1230. ACM, 2009.
- [17] S. Rendle. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 995–1000. IEEE, 2010.
- [18] I. Roll, V. Aleven, B. M. McLaren, and K. R. Koedinger. Improving students help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, 21(2):267–280, 2011.
- [19] Y. Shi, D. Ye, A. Goder, and S. Narayanan. A large scale machine learning system for recommending heterogeneous content in social networks. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1337–1338. ACM, 2011.
- [20] W. Tang, J. Tang, T. Lei, C. Tan, B. Gao, and T. Li. On optimization of expertise matching with various constraints. *Neurocomputing*, 76(1):71–83, 2012.
- [21] L. Terveen and D. W. McDonald. Social matching: A framework and research agenda. *ACM Trans. Comput.-Hum. Interact.*, 12(3):401–434, Sept. 2005.
- [22] F. Xu, Z. Ji, and B. Wang. Dual role model for question recommendation in community question answering. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 771–780, New York, NY, USA, 2012. ACM.
- [23] D. Yang, T. Chen, W. Zhang, Q. Lu, and Y. Yu. Local implicit feedback mining for music recommendation. In *Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12*, pages 91–98, New York, NY, USA, 2012. ACM.
- [24] D. Yang, M. Piergallini, I. Howley, and C. Rose. Forum thread recommendation for massive open online courses. In *Proceedings of 7th International Conference on Educational Data Mining*, 2014.
- [25] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Workshop on Data Driven Education, Advances in Neural Information Processing Systems 2013*, 2013.