

# Constrained Question Recommendation in MOOCs via Submodularity

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

Jingbo Shang  
Computer Science  
Department  
Shanghai Jiaotong University  
Shanghai, China, 200240  
shangjingbo@gmail.com

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

## ABSTRACT

A recent area in which recommender systems have shown their value is in online discussion forums and question-answer sites. Earlier work in this space has focused on the problem of matching participants to opportunities but has not adequately addressed the problem that in these social contexts, multiple dimensions of constraints must be satisfied, including limitations on capacity and minimal requirements for expertise. In this work, we propose such a constrained question recommendation problem with load balance constraints in discussion forums and use flow based model to generate the optimal solution. In particular, to address the introduced computation complexity, we investigate the concept of submodularity of the objective function and propose a specific submodular method to give an approximated solution. We present experiments conducted on two Massive Open Online Course (MOOC) discussion forum datasets, and demonstrate the effectiveness and efficiency of our submodular method in solving constrained question recommendation tasks.

## Categories and Subject Descriptors

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

## Keywords

Constrained Question Recommendation; Max-cost Flow; Submodularity

## 1. INTRODUCTION

The past decade has witnessed an explosive growth of accessible information. Recommender systems provide one means for assisting users to find items and opportunities that are of interest, or to get other information needs met. In this paper, we focus specifically on a form of social recommendation where we direct participate in discussion forums to interesting discussions while increasing

the likelihood that posted information requests will garner the necessary attention from qualified question answerers. In such scenarios, it is necessary to consider constraints in a more comprehensive manner than in the simplest case of product recommendation, where it is only necessary to consider the desires of the consumer. In the case of social recommendation where people are being matched, the match must be desirable on both sides. In the case of information seeking, the additional constraints of asymmetric expertise and limitations on the number of requests a single participant can be expected to address must also be considered [10]. A similar issue comes up in the context of peer review. In this context, each reviewer may be assigned a personalized budget specifying the maximum number of papers to review. At the same time, regardless of popularity in the bidding process, each paper should not be reviewed by too many people [9].

Max cost flow based techniques have been applied in order to achieve optimal satisfaction in this load balanced constrained question recommendation problem [10]. However, they are too computationally expensive to generalize and scale to massive datasets. Therefore, in this paper, we propose a constrained question recommendation problem built on earlier formulation and introduce a submodularity method to reduce the computational complexity and speed up the optimization. The contributions of this paper are highlighted as below.

- We build on the max cost flow approach to globally optimizing a multi-dimensional set of constraints on question recommendation.
- We investigate the submodularity of this problem, prove its submodularity, and propose a submodular method to provide acceptable recommendations efficiently (with orders of magnitude improvement in efficiency).
- We present experiments conducted on two discussion forum datasets to demonstrate the effectiveness and efficiency of this submodular method.

This paper is organized as follows. Section 2 introduces related work. The constrained question recommendation problem is formulated in Section 3 while solving methods are presented in Section 4. Section 5 shows the experimental results, and conclusion is presented in Section 6.

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

*CIKM'14*, November 3–7, 2014, Shanghai, China.

Copyright 2014 ACM 978-1-4503-2598-1/14/11 ...\$15.00.

<http://dx.doi.org/10.1145/2661829.2662089>.

**Table 1: Notation**

Notation	Meaning
$U, I$	User/question set
$u, i$	User/question
$E$	Candidate user-question pairs
$r_{u,i} \in [0, 1]$	Given relevance score between $u$ and $i$

## 2. RELATED WORK

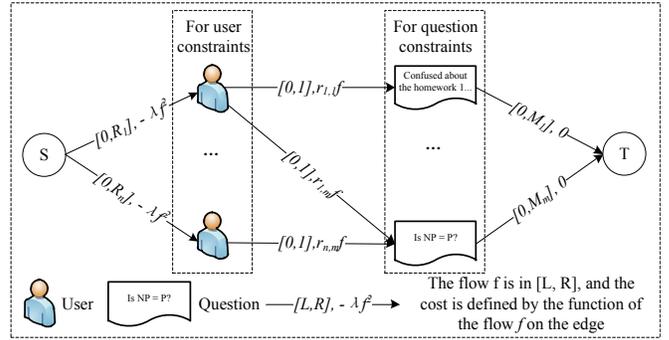
Traditional recommender systems mainly focus on approximating the extent to which a recommended item will be desirable to users. Collaborative filtering (CF) has been demonstrated to be an effective technique for accomplishing this goal. Neighborhood based models [8] and latent factor models [6] are two classical approaches of CF. For example, Koren et al. [5] designed a framework to combine latent modeling and neighborhood modeling between users and movies in order to provide better movie recommendation performance. When it comes to question recommendation, i.e. recommending questions to potential answerers, most work also focuses on predicting the affinity between users and question content. For instance, Kabutoya et al. [4] generates a suitable user set to answer a given question based on question and user histories using CF techniques.

However, such question recommender systems have not addressed the possible constraints associated with users and questions. Especially in the Massive Open Online Courses (MOOCs), recommending questions to students should be based on not only the affinity of students for the topic of the questions, but also under the constraints of student time budget and ability to answer the question. For instance, Yang et al. [10] propose a constrained question recommendation problem with load balance and expertise matching constraints and develop a max cost flow model to optimally generate recommendations. Similar recommendation tasks with constraints also include recommending reviewers for papers. Tang et al. [9] propose a minimum cost flow based model to meet kinds of constraints associated with reviewers and papers. However, the disadvantage of such flow models is the expensive computational complexity. Therefore, we propose to further develop this approach by investigating the submodularity of the objective function [7].

## 3. CONSTRAINED RECOMMENDATION FORMULATION

We begin our discussion by listing some basic notations in Table 1 that we will use later. The constrained question recommendation (CQR) task is to recommend a set of questions to users under the simultaneous constraints of users and questions. These balancing constraints can be phrased in this way: each question  $i$  is associated with a question capacity  $M_i$ , meaning we should not guide more than  $M_i$  users to a question since it might be easy and does not require too many people to provide help; for each user  $u$ , we associate a user capacity  $R_u$ , representing the idea that the number of questions that are comfortable for  $u$  to work on should not be larger than  $R_u$ . We formulate this constrained question recommendation problem as follows, and maximize this objective function, with the extent to minimize the load given to each user:

$$\begin{aligned} \text{OB} &= -\lambda \sum_{u \in U} (\sum_{i \in I} f_{u,i})^2 + \sum_{u,i \in E} f_{u,i} r_{u,i} \\ \text{subject to} \quad & \forall u \in U, \sum_{i \in I} f_{u,i} \leq R_u \\ & \forall i \in I, \sum_{u \in U} f_{u,i} \leq M_i \end{aligned} \quad (1)$$


**Figure 1: The Max-cost Flow Network Model**

Here,  $\lambda (\geq 0)$  is a non-negative coefficient for the load penalty and  $f_{u,i}$  is a binary indicator of whether question  $i$  will be recommended to user  $u$ . Instead of working on approximating the preference scores  $r_{u,i}$  of user  $u$  on item  $i$ , in this work, we assume  $r_{u,i}$  is *given*, i.e. generated by some state-of-the-art algorithms as in our earlier work, and focus on solving the constrained question recommendation problem efficiently.

## 4. OPTIMIZATION METHODS

In this section, we present three methods to maximize the objective function in Equation 1, which we experimentally compare in our evaluation. The first top-K technique will be the baseline for our experimental comparisons. It is a greedy method that efficiently provides possible solutions. The two experimental approaches are the max-cost flow model, which is theoretically optimal, but suffers from high computational complexity; and the submodularity approach, which can approximate the optimal solution well with less expensive computation compared to the max-cost flow model.

### 4.1 Traditional Top-K Recommendation

The traditional top-K method recommends questions to users by selecting the top-K ranked items for each user individually and ignores the capacity constraints. To fit our setting, we develop a post-processing model that prunes the extraneous users with low relevance scores. That is, we first run the traditional top-K item selection to recommend  $R_u$  items to each user. Then, for each item, if the number of recommended users exceeds its limit, we prune the users in the increasing order of their relevance scores (beginning with the least relevant).

### 4.2 Max-cost Flow Model

To maximize the objective function in Equation 1, we construct a concave cost network as summarized in Figure 1. Then we conduct a max-cost flow optimization over this network to get an optimal solution under the constraints. We prove that conducting max-cost flow on this network can give an optimal solution to the problem defined in Equation 1.

**THEOREM 1.** *The cost network constructed in Figure 1 is equivalent to the problem defined in Equation 1 and provides an optimal solution.*

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

---

**Algorithm 1** Submodular Method for CQR

---

```
1:  $A \leftarrow \emptyset, B \leftarrow E$ .
2: for user item pair  $\langle u, i \rangle \in E$  do
3:    $v_+ \leftarrow \max(OB_{ext}(A \cup \langle u, i \rangle) - OB_{ext}(A), 0)$ 
4:    $v_- \leftarrow \max(OB_{ext}(B \setminus \langle u, i \rangle) - OB_{ext}(B), 0)$ 
5:    $x \sim \text{Uniform}[0, 1]$ 
6:   if  $x \leq v_+ / (v_+ + v_-)$  then
7:      $A \leftarrow A \cup \langle u, i \rangle$ 
8:   else
9:      $B \leftarrow B \setminus \langle u, i \rangle$ 
10: return  $A (= B)$ 
```

---

problem:

$$\max \sum_{(a \rightarrow b) \in E(G)} Cost_{a \rightarrow b}(Flow_{a \rightarrow b})$$

subject to

$$\forall a (\neq S, T) \in V(G), \sum_b Flow_{a \rightarrow b} = \sum_b Flow_{b \rightarrow a}$$
$$0 \leq Flow_{a \rightarrow b} \leq R_{a \rightarrow b} \quad (2)$$

where,  $S$  is the source node and  $T$  is the sink;  $Flow_{a \rightarrow b}$  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. If we set  $f_{u,i}$  as the corresponding flow in the network  $flow_{u \rightarrow i}$ , the objective in MCCF is equal to Equation 1. After that, we consider the remaining constraints one by one.

- The user constraint is bounded by the capacity  $[0, R_u]$  from the source  $S$  to the user nodes  $u$ .
- The question constraint is limited by the flow range  $[0, M_i]$  from question nodes  $i$  to the sink  $T$ .

In summary, all constraints in CQR problem are satisfied in the max-cost flow model, and these constraints are the only limitations in the network. Thus, the MCCF problem defined here is equivalent to the CQR problem. Then we can formally build the network, solve the MCCF problem as shown in Figure 1, and get the final recommendation plan by examining flows between  $u$  and the corresponding  $i$ .  $\square$

Based on the Theorem 1, we can transform the integer programming problem in Equation 1 to a max-cost flow model with  $O(|U| + |I|)$  nodes and  $O(|E| + |U| + |I|)$  edges. Using the basic max-cost flow algorithm implemented using the Bellman-ford algorithm [1], we can compute the optimal solution in  $O((|U| + |I|)^2 |E|)$  (the scale of flows is in  $O(|U| + |I|)$  if we assume the  $R_u$  and  $M_i$  are both constants).

### 4.3 Submodular Method

Even though the constructed max cost flow model above can achieve an optimal solution, it suffers from the high computational complexity which prevents it from scaling to massive datasets. To overcome this issue, we introduce a submodular method to maximize Equation 1 here. First, we extend the objective function as described below,

$$OB_{ext} = \begin{cases} OB & \text{satisfied} \\ OB - C \sum_u \sum_i f_{u,i} & \text{violated} \end{cases} \quad (3)$$

where  $C$  is a sufficiently large constant that it helps the negative term dominate the value of  $OB_{ext}$  when it is

violated. It is clear that the optimal solution of  $OB_{ext}$  is equal to the  $OB$ 's optimal solution, because violating the constraints leads to an exponential extra penalty. In what follows, we show the submodularity exhibited by  $OB_{ext}$ .

**THEOREM 2.**  $OB_{ext}$  is a submodular function. The submodular property is defined as follows:  $\forall A \subseteq B, \langle u, i \rangle \notin B$ , s.t.  $OB_{ext}(A \cup \langle u, i \rangle) - OB_{ext}(A) \geq OB_{ext}(B \cup \langle u, i \rangle) - OB_{ext}(B)$ .

**PROOF.** Recall that  $f(a + \Delta) - f(a) \geq f(b + \Delta) - f(b)$  holds for any concave function  $f(x)$  when  $a \leq b$  and  $\Delta \geq 0$ . It is worth noting that concavity is a special case of submodularity. Adding a user-item pair  $\langle u, i \rangle$  is equal to setting  $f_{u,i}$  to 1 from 0. Consider the following three cases:

*Case 1,*  $A$  and  $B$  both violate the constraints. In this case, since  $-C \sum_u \sum_i f_{u,i}$  dominates  $OB_{ext}$  which is concave,  $OB_{ext}$  is submodular.

*Case 2,*  $A$  satisfies the constraints but  $B$  violates the constraints. Here are two subcases: (1)  $A \cup \langle u, i \rangle$  satisfies the constraints. In this subcase, the value of  $OB_{ext}(B \cup \langle u, i \rangle) - OB_{ext}(B)$  is sufficiently large so that the inequality holds. (2)  $A \cup \langle u, i \rangle$  violates the constraints. In this subcase, because the term  $\sum_u \sum_i f_{u,i}$  is the index of  $C$  and  $|A| \leq |B|$ , the inequality still holds. Thus  $OB_{ext}$  is submodular.

*Case 3,*  $A$  and  $B$  both satisfy the constraints. In this case, the relevant part ( $r_{u,i}$ ) of the  $OB_{ext}$  difference (either  $OB_{ext}(A \cup \langle u, i \rangle) - OB_{ext}(A)$  or  $OB_{ext}(B \cup \langle u, i \rangle) - OB_{ext}(B)$ ) is the same and the penalty part ( $-\lambda x^2$ ) is concave, thus  $OB_{ext}$  is submodular.  $\square$

Based on Theorem 2, the extended objective function  $OB_{ext}$  is a submodular function. Thus, we can utilize the RandomizedUSM [2] algorithm as shown in Algorithm 1, which guarantees a  $\frac{1}{2}$  approximation for maximizing nonnegative submodular functions. This algorithm has a time complexity  $O(|E|)$ .

## 5. EXPERIMENTS

In this section, we firstly present our datasets and experimental setup. Then we investigate whether the submodular method achieves the desired reduction in time complexity with minimal sacrifice of performance.

### 5.1 Dataset and Settings

Our experiments are conducted on two discussion forums of courses from Coursera<sup>1</sup>, (1) 'Accountable Talk: Conversation that works', shortened to 'A', is a teacher professional development course with 1,148 users, 511 questions, and 582,945 edges; (2) 'Fantasy and Science Fiction: the human mind, our modern world', shortened to 'F', is a course about literature appreciation, which has 770 users, 603 questions and 460,448 edges. In our datasets, most posts (around 82%) are about proposing and resolving problems.

Here, we assume that students have a certain time budget and cannot be directed to too many questions, thus  $R_u$  is defined as how many questions this student has participated so far in the forum; for questions, we empirically set  $M_i$  to 10. It is worth mentioning that one can simulate the use of different values of  $R_u$  and  $M_i$  based

<sup>1</sup>Permission to conduct research on Coursera datasets was provided by Coursera.

**Table 2: Objective Function Comparison**

Method	OB-A	OB-F
Top-K Method	1342.70	1120.43
Max-cost Flow	2898.17	2652.99
Submodular	2546.45	2238.58

**Table 3: Time Complexity Comparison**

Method	Theoretical	Time-A	Time-F
Top-K Method	$O( E \log E )$	0.14s	0.17s
Max-cost Flow	$O(( U  +  I )^2 E )$	341.42s	269.90s
Submodular	$O( E )$	0.045s	0.048s

on specific problem needs. To get the relevance score  $r_{u,i}$  of students’ affinity over questions, we adopted the feature-based matrix factorization technique [3], which is a state-of-the-art method. Similarly, one can adopt other strategies to generate the relevance scores. In this work, we assume the relevance scores are all given and focus merely on the fast optimization of the constrained recommendation problem. The optimization performance is evaluated by the objective value (OB) of Equation 1. For comparison, we use the Top-K Method and Max-cost Flow as baselines; the  $\lambda$  is set as 0.001. The machine we used is equipped with Intel Xeon(R) CPU E5-2680 @2.70GHz and 128GB memory.

## 5.2 Performance Comparison

We present the results of Top-K Method, Max-cost Flow and Submodular Method in Table 2. As we can see, Max-cost Flow performs the best on both datasets in terms of the objective function, because it gives the optimal solution; the runner-up is the Submodular Method, and there are no substantial difference between the OB values of the two models. The Submodular Method achieves 87.9% of the optimal solution on the Accountable Course and 84.4% of the optimal solution on the Fantasy Course. Top-K Method gives the lowest OB, because it does not take into account the possible constraints in its first step and many pairs are pruned to satisfy the load balance constraints in the second step. Furthermore, the Submodular method is even more efficient than the baseline Top-K approach.

Taking both the time complexity comparison presented in Table 3 and the objective function optimization comparison into account, we can have a better understanding of the advantage of the Submodular Method. For Table 3,  $O(|E|)$  is the number of candidate user-question pairs and also the number of edges in the built max-cost flow network.  $O(|U| + |I|)$  is the node number. Since  $R_u$  and  $M_i$  can be treated as constants, the flow is in  $O(|U| + |I|)$ . Therefore, the theoretical time complexity of Max-cost Flow is  $O((|U| + |I|)^2|E|)$ , about  $O((|U| + |I|)^4)$ . However, the theoretical complexity of Submodular Method is only  $O(|E|)$ , about  $O((|U| + |I|)^2)$ . The running times on the two datasets also validate the theoretical results. Based on the running time on the Accountable Course, Submodular Method costs 0.01% of the running time of the Max-cost Flow method; this is consistent on the Fantasy Course. Taking into account the objective function optimization and time complexity comparison results, we can conclude that, the Submodular Method can give an 80% approximation of the optimal solution with very large reduction of the time complexity.

## 6. CONCLUSION

In this work, we formulate a constrained question recommendation problem with load balancing constraints in online discussion forums. To address the high computational complexity introduced by the optimal flow model, we investigate the submodularity of the objective function and propose a submodular method to speed up the optimization, with little sacrifice in quality. In future work, we propose to investigate submodularity further and derive a theoretical approximation bound.

## ACKNOWLEDGEMENT

This research was funded in part by NSF grants IIS-1320064 and OMA-0836012.

## 7. REFERENCES

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- [2] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 649–658. IEEE, 2012.
- [3] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. SVDFeature: A toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research*, 13:3619–3622, 2012.
- [4] Y. Kabutoya, T. Iwata, H. Shiohara, and K. Fujimura. Effective question recommendation based on multiple features for question answering communities. In *ICWSM*, 2010.
- [5] 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, 2008.
- [6] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.
- [7] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.
- [8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW ’01, pages 285–295, 2001.
- [9] W. Tang, J. Tang, and C. Tan. Expertise matching via constraint-based optimization. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 34–41. IEEE, 2010.
- [10] D. Yang, D. Adamson, and C. Rose. Question recommendation with constraints for massive open online courses. In *Proceedings of the Eighth ACM Conference on Recommender Systems*, RecSys ’14, 2014.