Social Learning Networks: Efficiency Optimization for MOOC Forums

Christopher G. Brinton, Swapna Buccapatnam, Felix Ming Fai Wong, Mung Chiang, and H. Vincent Poor
Department of Electrical Engineering, Princeton University
{cbrinton, swapnab, mwthree, chiangm, poor}@princeton.edu

***Abstract*—A Social Learning Network (SLN) emerges when users exchange information on educational topics with structured interactions. The recent proliferation of massively scaled online (human) learning, such as Massive Open Online Courses (MOOCs), has presented a plethora of research challenges surrounding SLN. In this paper, we ask: How *efficient* are these networks? We propose a framework in which SLN efficiency is determined by comparing user benefit in the observed network to a benchmark of maximum utility achievable through optimization. Our framework defines the optimal SLN through utility maximization subject to a set of constraints that can be inferred from the network. Through evaluation on four MOOC discussion forum datasets and optimizing over millions of variables, we find that SLN efficiency can be rather low (from 68% to 82% depending on the specific parameters and dataset), which indicates that much can be gained through optimization. We find that the gains in global utility (*i.e.,* average across users) can be obtained without making the distribution of local utilities (*i.e.,* utility of individual users) less fair. We also discuss ways of realizing the optimal network in practice, through curated news feeds in online SLN.**

I. Introduction

Social Learning Network (SLN) encapsulates a range of scenarios in which a number of people learn from one another through structured interaction. The proliferation of online communication has given rise to a number of SLN applications, ranging from Question and Answer (Q&A) sites (*e.g.,* Quora), to enterprise social networks (*e.g.,* Jive), to platforms for online education. They have created learning networks among askers/answerers, employees, and students, respectively [1].

Within the realm of online (human) learning, one of the most profound applications of SLN today is the Massive Open Online Course (MOOC). MOOCs, offered by platforms such as Coursera, edX, and Udacity, have scaled distance education to previously unimaginable sizes, reaching hundreds of thousands of students within single sessions of a course [2]. But they also suffer from low completion rates, often attributed to a variety of factors, such as low teacher-to-student ratios and a lack of face-to-face interaction [3].

In an effort to alleviate some of these problems, MOOC platforms provide discussion forums within each course. These forums serve as the primary means for interaction between students (through user-generated posts/comments), providing an avenue for question asking and answering similar to the structure of Q&A sites [1]. While MOOC forums can be monitored by instructors and teaching staff, the large volume of posts made by students (*e.g.,* >25K for one of the datasets in Table I) makes it infeasible for the staff to reach out and answer every question. As a result, the efficacy of these forums hinges on the notion that when a student posts a question on a topic, one (or more) of her peers will respond with an answer sufficient in quality, *i.e.,* that strong social ties will form between those seeking information on certain topics and those who are experts on these topics [4].

In this paper, we are motivated by the following three questions related to the SLN of MOOC discussion forums:

- *How efficient is the observed information exchange between users?*
- *What does the ideal SLN look like?*
- *How does the structure of the ideal SLN differ from that of the observed SLN?*

SLN optimization. To study these questions, we propose a novel framework for modeling SLN efficiency (Sec. II), which compares the benefit obtained by users in the observed network to that which can theoretically be obtained through optimization. Each student is modeled individually as possessing certain levels of seeking (*i.e.,* question asking) and disseminating (*i.e.,* question answering) tendencies on a set of latent educational topics for the course. Additionally, our framework models the social structure of the SLN, which captures the level of connectivity between each pair of users. There is some existing work on studying the content of MOOC forums (*e.g.,* [5]) and some studying the graph structure (*e.g.,* [6]); our work considers a unified view of both components.

Taken together, these components give us a natural way of defining user benefit: Determine (i) the match between a user's seeking tendency and the disseminating tendencies of her neighbors (*i.e.,* her benefit from learning), and (ii) the match between the user's disseminating tendencies and the seeking tendencies of her neighbors (*i.e.,* her benefit from teaching [7]). In an ideal setting, both of these would be as large as possible. Therefore, our optimization searches for the SLN that is most compatible with the individual tendencies of the users, trading off the global utility (*i.e.,* average benefit) and local utility (*i.e.,* individual benefits). It also accounts for the fact that the amount a student will participate in the forums is constrained by her own resource limitations. Different from the optimization of users to questions proposed in [8], our method accounts for the difference between seeking and disseminating tendencies of users over a multidimensional topic space.

Performance evaluation. After presenting our framework, we decide on several specific algorithms for inference and

978-1-4673-9953-1/16/$31.00 ©2016 IEEE

optimization (Sec. III). In particular, since our optimization involves several million variables corresponding to the weights in a directed user-to-user graph, it poses unique computational challenges. With these algorithms in place, we perform an efficiency evaluation on four real world MOOC datasets (Sec. IV); in comparing the observed and optimal SLN, we make three key observations:

- The observed efficiencies are rather low, ranging from 68% to 82% of the optimal depending on the specific parameters and dataset.
- The optimal SLN has a much more homogeneous structure, with both outgoing degree and edge weight distributions becoming more uniform.
- The optimal SLN does not penalize the fairness of local utilities, and in fact increases utility for individual users in the majority of cases.

More generally, there are three steps involved in improving the efficiency of an SLN: (i) defining the ideal SLN through an optimization, (ii) solving for the optimal SLN, and (iii) implementing the optimized network in practice. In presenting and evaluating our efficiency framework, our focus in this work is the first two steps, which allows us to quantify the gains that will be obtained if our modeling assumptions hold in reality. The third step involves designing a system to enforce the optimized interaction structure in the corresponding SLN, for which the observed efficiencies may be higher or lower for a number of reasons. In Sec. IV-E, we briefly discuss possibilities for this last step (*e.g.,* a curated news feed for recommended interactions), and outline the key challenges.

II. SLN Efficiency

To evaluate the efficiency of an SLN, we pose the following question: *How much are users benefiting from the observed network structure relative to how much they could benefit from an optimized structure?* In this section, we will present our efficiency framework, consisting of our graph model (Sec. II-A), utility model (Sec. II-B), and optimization (Sec. II-C).

A. Graph Modeling of SLN (W, S, and D)

We first define and model the fundamental components of an SLN:

Users. At its core, an SLN is a network of users (*i.e.,* learners) sharing information on different topics. Let $u \in \mathcal{U}$ denote user u in the set of users $\mathcal{U} = \{1, 2, ...\}$ that comprise the SLN.

Network. In studying efficiency, we are interested in the interaction structure between users. We define $W = [w_{u,v}]$, for $u, v \in \mathcal{U}$ ($w_{u,u} = 0$) to be the weighted adjacency matrix of the user-to-user network, where $w_{u,v}$ represents the spread of information from u to v. More concretely, we consider $0 \leq w_{u,v} \leq 1$ to be the probability that u will respond to v when v makes a post, with $w_{u,v} \neq w_{v,u}$ in general.

Topics. Discussions in an SLN center around a series of (possibly latent) topics. Let $k \in \mathcal{K} = \{1, 2, ...\}$ denote topic k in the set $\mathcal{K}$ of discussion topics for the SLN.

Seeking and disseminating. With respect to each topic, a user will have some tendency towards disseminating information (*i.e.,* providing answers or facts about the material) or seeking information (*i.e.,* asking questions about the material). In order to capture this behavior, we define $s_{u,k} \geq 0$ to be user u's seeking tendency on topic k, and $d_{u,k} \geq 0$ as her disseminating tendency on k, with $S = [s_{u,k}]$ and $D = [d_{u,k}]$. In an ideal scenario, users seeking information would be receiving responses from those disseminating on the same subject matter, *i.e.,* the corresponding $w_{u,v}$ would be large.

There are many possibilities on how to infer W, S, and D from SLN data. In Sec. III, we describe the methods that we employ for doing so in this work.

B. Utility Modeling of SLN (B and E)

We identify two types of user benefit from an SLN:

(1) Learning benefit. Intuitively, user u will gain from having higher connections to those who tend to disseminate information on topics that u asks questions on. We quantify this as $s_{u,k} \cdot f(\sum_v w_{v,u} d_{v,k})$, where $w_{v,u} d_{v,k}$ captures the expected amount of response provided from v to u on topic k, and f is a concave function to capturing diminishing return associated with receiving more response. This entire term is weighted by $s_{u,k}$, which weighs each topic differently depending on how much information u is seeking on the topic in the first place.

(2) Teaching benefit. In peer-to-peer learning, users also draw benefit from acting as teachers to others, *i.e.,* from learning by teaching [1], [7]. For user u, this can be quantified as $d_{u,k} \cdot f(\sum_v w_{u,v} s_{v,k})$, where $w_{u,v} s_{v,k}$ captures the amount by which u will provide information to user v that is sought by v about topic k, and f captures the diminishing return aspect of learning from teaching. This entire term is weighted by $d_{u,k}$, which is a measure of the amount of information u provides about the topic.

Global and local utility. Let $B = [b_{u,k}]$ be the matrix of user-topic benefits, where $b_{u,k} \geq 0$ is the utility obtained by user u with respect to topic k. These benefits are modeled as:

$$\begin{aligned} b_{u,k} = s_{u,k} \log(1 + \sum_v w_{v,u} d_{v,k}) \\ + \alpha_u \cdot d_{u,k} \log(1 + \sum_v w_{u,v} s_{v,k}). \end{aligned} \tag{1}$$

Here, α_u quantifies the benefit of teaching relative to learning for user u, and we choose $f(x) = \log(1 + x)$ because it is a standard concave utility function. We will discuss the approach we take for setting α_u in Sec. IV.

(i) Local utility: The local utility l_u of an SLN to a specific user u is defined as the total benefit obtained by u across all topics k. From (1), this is obtained as $l_u = \sum_k b_{u,k}$.

(ii) Global utility: The global utility g is defined as the average local utility across users. From (1), $g = \sum_u \sum_k b_{u,k} / |\mathcal{U}|$.

Deficit: Let $E = [\epsilon_{u,k}]$ be the matrix of user-topic deficits, where

$$\epsilon_{u,k} = \max(0, s_{u,k} - \sum_v w_{v,u} d_{v,k}) \tag{2}$$

is the deficit between u's seeking tendency for k and the incoming disseminating tendency from her neighbors. $\epsilon_{u,k} = 0$ implies that u's seeking tendency on k is satisfied, *i.e.,* she is

receiving enough information from her neighbors. Notice that for S and D constant, higher $\epsilon_{u,k}$ implies lower $\sum_v w_{u,v} d_{v,k}$, which in turn implies a decrease in $b_{u,k}$ in (1). Hence, a higher deficit $\epsilon_{u,k}$ across k implies a lower local utility l_u for user u.

C. Optimizing SLN

We seek the combination of weights W in the SLN that will maximize the global utility g of the SLN while minimizing the deficits E in information provided to specific u, k pairs. Formally, our optimization problem is given as:

$$\text{maximize} \quad F_{\alpha_u,\alpha_\epsilon} = \frac{1}{|\mathcal{U}|} \sum_u \sum_k (b_{u,k} - \alpha_\epsilon \cdot \epsilon_{u,k}) \tag{3a}$$

$$\text{subject to} \quad \sum_v w_{v,u} d_{v,k} \geq s_{u,k} - \epsilon_{u,k}, \ \ \forall u,k \tag{3b}$$

$$\sum_v w_{u,v} \leq \bar{w}_u, \ \ \forall u \tag{3c}$$

$$0 \leq w_{u,v} \leq 1, \ w_{u,u} = 0, \ \ \forall u,v \tag{3d}$$

$$\epsilon_{u,k} \geq 0, \ \ \forall u,k \tag{3e}$$

$$\text{variables} \quad W, E \tag{3f}$$

α_ϵ is the deficit penalty, which captures the tradeoff in importance between the two (possibly conflicting) objectives. In particular, we want to avoid solutions that would route the vast majority of information spread to those users u with highest seeking tendencies (which could occur since $b_{u,k}$ is proportional to $s_{u,k}$). Since higher deficit $\epsilon_{u,k}$ across topics k causes lower local utility l_u, α_ϵ is also trading off the maximization of g and the minimization of the impact on l_u.

In (3), the objective (3a) is a concave function, because the $b_{u,k}$ are concave while the $\epsilon_{u,k}$ are linear. There are two linear constraints (besides bounds):

1) Seeking tendency satisfaction (3b): $\sum_v w_{v,u} d_{v,k}$ measures the amount of information transferred to user u on topic k. Ideally, this should meet u's seeking tendency $s_{u,k}$. If it does not, then $\epsilon_{u,k}$ captures the deficit, and the objective is penalized according to the weight of α_ϵ.

2) Load balancing (3c): This constraint captures the fact that each user has a finite capacity on the amount of participation she can provide, which depends on a number of external factors, *e.g.,* time commitments and willingness to use the forums in the first place. In order to bound overload, we define $\bar{w}_u$ to be the maximum amount of interaction that u can provide, and restrict $\sum_v w_{u,v}$ (*i.e.,* the total outgoing response probability) to not exceed this. We will discuss how to infer $\bar{w}_u$ from the observed network in Sec. III.

Efficiency metrics. Let $F_{\alpha_u,\alpha_\epsilon}(W,E)$ be the value of (3a) with respect to the variables W and E for fixed S, D, and parameters α_u, α_ϵ. The efficiency of a given SLN with respect to the objective function F is quantified as

$$\eta^F_{\alpha_u,\alpha_\epsilon} = F_{\alpha_u,\alpha_\epsilon}(\hat{W},\hat{E}) / F_{\alpha_u,\alpha_\epsilon}(W^*,E^*). \tag{4}$$

Here, $\hat{W}$ and $\hat{E}$ are the observed network and deficit terms, S and D are the observed seeking and disseminating tendencies, and W^* and E^* are the optimized versions of W and E obtained from solving (3). In other words, η^F is the fraction of the (penalized) global utility achievable in the optimal network that is already obtained by the observed network.

We are also interested in a measure of efficiency without the penalty term. To this end, let $g_{\alpha_u,\alpha_\epsilon}(\hat{W},\hat{E})$ be the observed global utility. The efficiency with respect to g is quantified as

$$\eta^g_{\alpha_u,\alpha_\epsilon} = g_{\alpha_u,\alpha_\epsilon}(\hat{W},\hat{E}) / g_{\alpha_u,\alpha_\epsilon}(W^*,E^*), \tag{5}$$

where $g(W^*,E^*)$ is the global utility of the optimal network.

III. Inference and Optimization Algorithms

To compute the efficiency measures (4),(5), we need to determine the observed social network ($\hat{W}$) and deficits ($\hat{E}$), and we need to solve the optimization (3) to obtain W^* and E^*. For $\hat{E}$ and solving (3), we must also infer the seeking (S) and disseminating (D) tendencies of the SLN, and we also need the user capacities ($\bar{w}_u$) for (3). In this section, we will describe how we infer these quantities and solve (3).

Forum structure. Each MOOC has a single forum, which consists of a series of threads. Each thread is comprised of one or more posts, with each post written by a single user. A post, in turn, can have one or more comments attached to it. For our purposes, we do not distinguish between posts and comments, and will refer to them both as posts.[1]

In what follows, let $r \in \mathcal{R}$ denote thread r in the set of threads $\mathcal{R} = \{1,2,...\}$ for a course, ordered chronologically by creation time. Let $p_r \in \mathcal{P}_r$ denote post p in the set $\mathcal{P} = \{1,2,...\}$ for r, also indexed chronologically.[2] Each p has an associated user $\mu(p)$, creation time $t(p)$, and text $x(p)$ written by $\mu(p)$. Here, $x = (x_1, x_2, ...)$ is the sequence of words and punctuation marks written by the user, where $x_i \in \mathcal{X}$ is the index into the dictionary $\mathcal{X}$; $\mathcal{X}$ is the set of all words and marks that appear across all posts in the course forum.

A. Computing the Observed Social Network ($\hat{W}$ and $\bar{w}_u$)

The first component to infer is the observed user-to-user network $\hat{W} = [\hat{w}_{u,v}]$. With $\mathcal{P}_{r,u} \subseteq \mathcal{P}_r$ as the subset of posts in r made by user u, there are a number of possibilities for doing so. For one, we could use the co-participation count between u and v across threads $\mathcal{R}$ as a measure of $\hat{w}_{u,v}$, *e.g.,* through the one-mode projection $\sum_r \min(|\mathcal{P}_{r,u}|, |\mathcal{P}_{r,v}|)$ [6]. But applying the user-thread bipartite graph directly leads to a symmetric $\hat{W}$. While this may be a valid assumption for friendship networks [3], it is not realistic to assume that interaction in an SLN is symmetric, since u answering v does not imply v will answer to u with the same probability.

We infer the $\hat{w}_{u,v}$ instead through the following approach: *If v makes a post in r, what is the probability that u will respond to this post?* In doing so, we use the following heuristic to infer which posts are meant as responses to others: if a unique post $p' \in \mathcal{P}_{r,u}$ is made by u after the post $p \in \mathcal{P}_{r,v}$ (*i.e.,* $t(p') > t(p)$), then p' is counted as a response to p. Formally,

[1] If comments were always written in response to posts, then the relationship between them could be useful for inferring Q&A tendencies in Sec. III-B2. But MOOC users do not abide to this structure consistently [3].

[2] We will drop subscripts like r when the context makes it clear.

let $n_{u,v}$ be the number of times that u posts after v, with $n_{u,u} = 0$.[3] With $N_v = \sum_r |\mathcal{P}_{r,v}|$ as the number of times v posted in the course, $\hat{w}_{u,v} = n_{u,v}/N_v$ is the fraction of times u responded to v. Since the N_v are diverse among forum users, giving each u varying opportunities to respond to v in the first place, we can apply a shrinkage estimator [10] to smoothen the $\hat{w}_{u,v}$ towards u's overall response rate $\sum_j n_{u,j}/\sum_j N_j$:

$$\hat{w}_{u,v}(\sigma) = \frac{n_{u,v} + \sigma(\sum_j n_{u,j}/\sum_j N_j)N_{max}}{N_v + \sigma N_{max}}, \qquad (6)$$

where σ is the smoothing parameter and $N_{max} = \max_i N_i$.

(6) with $\sigma = 0$ is our definition of the observed SLN. Operationally, as σ is increased, a user will spread his overall response rate more uniformly among the other users in the SLN. We will consider the effect of smoothing in Sec. IV.

Interaction capacity $\bar{w}_u$: Recall that $\bar{w}_u$ from (3c) is the outgoing capacity of user u. With $\hat{w}_{u,v}$ as the observed response probability from (6), we know at least that $\sum_v \hat{w}_{u,v}$ is an attainable level of participation from u. Hence, we take a conservative approach and set $\bar{w}_u = \sum_v \hat{w}_{u,v}$.

B. Inferring Seeking and Disseminating Tendencies (S and D)

The next components to infer are the seeking $S = [s_{u,k}]$ and disseminating $D = [d_{u,k}]$ tendencies.[4] We estimate $s_{u,k}$ and $d_{u,k}$ in three steps: (1) extracting the forum topics from the text, (2) inferring whether each post is a question or an answer, and (3) computing $s_{u,k}$ and $d_{u,k}$ from these quantities.

1) Topic extraction: We employ Latent Dirichlet Allocation (LDA), a popular generative model for topic extraction from a collection of documents [12]. LDA has been applied to discussion forums in several studies, *e.g.,* in [4], [13].

Consider a collection of documents $\mathcal{N}$, where each $n \in \mathcal{N}$ is written as a series of indices $d_n = (d_{n,1}, d_{n,2}, ...)$ to its constituent words, $d_{n,j}$ being an index into the dictionary $\mathcal{X}'$ (we will discuss the choice of n and $\mathcal{X}'$ further below). Under LDA [12], each document n is modeled as a random mixture over a set of topics $\mathcal{K}$, and each $k \in \mathcal{K}$ is in turn characterized as a distribution over $\mathcal{X}'$. The document-topic distributions $\theta = [\theta_{n,k}] \in [0,1]^{|\mathcal{N}|\times|\mathcal{K}|}$ are such that $\theta_{n,k}$ gives the proportion of n made up of k, and the topic-word distributions $\phi = [\phi_{k,x}] \in [0,1]^{|\mathcal{K}|\times|\mathcal{X}'|}$ are such that $\phi_{k,x}$ gives the fraction of k made up of word x. Under the generative process for LDA, each word position j in document n is assigned a single topic $c_{n,j}$, where $c_{n,j} \in \mathcal{K}$ is chosen from a multinomial distribution over $\theta_n = \{\theta_{n,1}, ..., \theta_{n,|\mathcal{K}|}\}$. With $k = c_{n,j}$, the specific word $x_{n,j} \in \mathcal{X}'$ for each position is then chosen from a multinomial distribution over $\phi_k = \{\phi_{k,1}, ..., \phi_{k,|\mathcal{X}'|}\}$.[5]

In developing LDA for our application, we must choose at which granularity of content to define a document, and which words $\mathcal{X}' \subset \mathcal{X}$ to be considered within each document. We use each post p as a separate document (similar to in [13]) because there could be multiple topic proportions within a thread (*i.e.,* the discussions may evolve over time). From the set of words and punctuation marks $\mathcal{X}$, we obtain $\mathcal{X}' \subset \mathcal{X}$ by: (i) removing all URLs, (ii) removing all punctuations, (iii) removing all stopwords from an aggressive 635 stopword list,[6] (iv) stemming all words left in $\mathcal{X}$, and finally (v) removing all words of length 1. We will see in Sec. IV-B1 that these methods and choices result in sets of topics that are qualitatively representative of key course discussions.

2) Question/answer tendency: With the post-topic distributions θ, the next step towards inferring $s_{u,k}$ and $d_{u,k}$ is to determine if each post p is a question or an answer. We define $Q(p)$ as an indicator of whether the text $x(p)$ is a question ($Q(p) = 1$) or not ($Q(p) = 0$). We will describe our specific method for determining $Q(p)$ below; to reduce noise associated with each $Q(p)$ irrespective of the method, we will consider the averaged question tendency $q_{u,r,k}$ of user u in thread r for topic k. This is defined as the weighted-average $Q(p)$ for u, with respect to the post-topic proportions $\theta_{p,k}$:

$$q_{u,r,k} = \frac{\sum_{p\in\mathcal{P}_{r,u}} \theta_{p,k} \cdot Q(p)}{\sum_{p\in\mathcal{P}_{r,u}} \theta_{p,k}}. \qquad (7)$$

There are two types of algorithms for question detection: rule-based methods, *e.g.,* whether a question mark is present [4], and learning-based methods, *e.g.,* a classifier analyzing sequences of parts of speech [14]. In our work, we apply a combination of rule-based methods, because some of them have high quality already; for example, in [14], question-mark detection had an F1-score (F1) of roughly 85%.[7] Formally, let $?_p$ denote the event "question mark $\in \mathbf{x}(p)$", let $\texttt{5W1H}_p$ denote "who, what, where, when, why, or how $\in x(p)$", and let $\texttt{UG}_p$ denote "please, thanks, help, confuse, grateful, or appreciate $\in x(p)$".[8] $Q(p)$ is determined as: $Q(p) = ?_p \cup \texttt{5W1H}_p \cup \texttt{UG}_p$ if $p = 1$; $Q(p) = ?_p \cap (\texttt{5W1H}_p \cup \texttt{UG}_p)$ if $p \neq 1$. We conditioned $Q(p)$ this way because a high proportion of the first posts in threads ($p = 1$) will be questions, with users creating threads for this purpose in the first place; for all other posts ($p \neq 1$), we add additional protection against false positives.

To test $Q(p)$, we obtained human-generated labels on roughly 750 of the posts taken across our datasets in Sec. IV (for procedural details, see our technical report [9]). Our $Q(p)$ obtained an accuracy of 0.86 and an F1 of 0.65 when compared to the labels. This accuracy is quite high, but the F1 is lower than those cited in other work [14], which emphasizes the importance of averaging in (7) to reduce noise in $Q(p)$.

3) $s_{u,k}$ and $d_{u,k}$ estimation: From (7), we estimate the disseminating and seeking tendency of user u on topic k as

$$d_{u,k} = \sum_r (1 - q_{u,r,k}) \cdot \log(1 + \sum_{p\in\mathcal{P}_{r,u}} \theta_{p,k} \cdot |x'_p|), \qquad (8)$$

$$s_{u,k} = \sum_r q_{u,r,k} \cdot \log(1 + \sum_{p\in\mathcal{P}_{r,u}} \theta_{p,k} \cdot |x'_p|), \qquad (9)$$

[3]The algorithm to compute $n_{u,v}$ is given in our online technical report [9].

[4]These could be inferred independent of k, similar to in [8], but this is undesirable because the topics discussed in MOOC are diverse [3], [11].

[5]The multinomials here are single trials.

[6]http://www.webconfs.com/stop-words.php

[7]The (balanced) F1-score of a classifier is the harmonic mean of the precision and recall, which is a standard way of evaluating a classifier [10].

[8]5W1H are standard question words. We observed that urgency/gratitude (UG) words tend to appear frequently in question posts too.

where $q_{u,r,k}$ is from (7) and x'_p, $\theta_{p,k}$ are the sequence of words and post-topic distributions from Sec. III-B1. The inclusion of text length here captures the fact that longer posts tend to contain more information. In the case of $d_{u,k}$, intuitively, more information in text containing topic k should increase u's disseminating tendency on k. In the case of $s_{u,k}$, it implies that the user is willing to spend more time on k. We employ $\log$ again to capture diminishing returns with higher post size.

Out of the quantities needed in (3),(4),(5), we now have methods to infer S, D, $\hat{W}$, and $\bar{w}_u$. Only $\hat{E}$ remains, which can now be obtained from (2) using $\hat{W}$, S, and D.

C. Solving for the Optimal Network (W^* and E^*)

The final component to specify is the algorithm to solve the optimization (3). This is a convex optimization problem, because the objective (3a) is concave and the constraints (3b)-(3e) are affine. This means that it is solvable, in theory, using off-the-shelf tools based on standard algorithms such as interior point methods. However, the number of variables in our problem is $|\mathcal{U}|\times\ (|\mathcal{U}|-\ 1+|\mathcal{K}|)$; with just 1K users (which is on the order of the smallest dataset in Table I), there are already over 1M variables, which makes these standard methods computationally intractable [15]. As a result, we instead derive a projected gradient descent method, for which three steps are repeated in sequence: Gradient, Projection, and Objective. The pseudocode is given in Algorithm 1, and the individual steps are as follows:

1) Gradient step: Here, the gradient of (3a) must be computed with respect to each variable. For variables $w_{u,v}$ and $\epsilon_{u,k}$, it is easy to show that

$$\frac{\partial F}{\partial w_{u,v}} = \frac{1}{|\mathcal{U}|}\sum_k \left(\frac{d_{u,k}s_{v,k}}{1+\sum_i w_{i,v}d_{i,k}} + \frac{\alpha_u d_{u,k}s_{v,k}}{1+\sum_j w_{u,j}s_{j,k}} \right) \tag{10}$$

$$\frac{\partial F}{\partial \epsilon_{u,k}} = -\frac{\alpha_\epsilon}{|\mathcal{U}|}. \tag{11}$$

In Algorithm 1, the procedure moves in the direction of the gradient ∇F in each iteration, by the step size γ.

2) Projection step: The solution from the gradient update is then projected onto the feasible region of (3). Since the constraints are affine, this problem can be cast as a linearly-constrained quadratic program. Formally, with W' and E' as the variables before projection, we solve:

$$\begin{array}{ll} \text{minimize} & ||W-W'||_2^2 + ||E-E'||_2^2 \\ \text{subject to} & \text{Constraints (3b)-(3e)} \\ \text{variables} & W, E \end{array} \tag{12}$$

In Algorithm 1, the function P refers to solving (12).

3) Objective step: Finally, the objective F is re-computed for the updated W, E. The algorithm terminates once the percent change in F between two successive iterations is below a small threshold T.

IV. Datasets and Results

In this section, we evaluate the efficiency of four MOOCs, and compare the properties of the observed and optimal SLN.

Algorithm 1 Projected gradient descent for solving (3).

Input: S, D, $\bar{w}_u\ \forall u$, $\alpha_u\ \forall u$, α_ϵ, $N=|\mathcal{U}|$, γ, T
Initialize: $F[-1] \leftarrow -\infty$, $W[0] \overset{Uni}{\sim} [0,1]^{N\times N}$, $n \leftarrow 0$
$E[0] \leftarrow \max(0, S - W^T[0] \times D)$ {max is element-wise}
$F[0] \leftarrow F(W[0], E[0])$
while $(F[n]-F[n-1])\ /\ |F[n-1]| \geq T$ **do**
  $W'[n+1] \leftarrow W[n] + \gamma \cdot \nabla F(W[n])$ {$\nabla F(W[n])$ from (10)}
  $E'[n+1] \leftarrow E[n] + \gamma \cdot \nabla F(E[n])$ {$\nabla F(E[n])$ from (11)}
  $W[n+1], E[n+1] \leftarrow P(W'[n+1], E'[n+1])$ {P from (12)}
  $F[n+1] \leftarrow F(W[n+1], E[n+1])$
  $n \leftarrow n+1$
Return: $W^* = W[n], E^* = E[n]$

A. Datasets

We obtain our datasets from the MOOC provider Coursera. Since other MOOC platforms use the same forum structure, our methods are applicable to them as well.

1) Data collection: We coded crawling infrastructure that uses the `selenium` library in Python to collect data from a course's forum. We also wrote a parser that uses the `beautifulsoup` library in Python to extract the following information from each HTML page: the thread title, and for each post in the thread, the user ID, timestamp, and text created. The results were saved as text files.

2) Courses: We chose four MOOCs for analysis: "Machine Learning" (`ml`), "English Composition I" (`comp`), "Algorithms: Design and Analysis, Part 1" (`algo`), and "Shakespeare in Community" (`shake`). These were all MOOCs that, as of June 2015, were publicly-accessible and had passed the final exam date listed on the syllabus. Table I gives basic information on them;[9] each course has varying numbers of users, threads, and posts, but the sizes are all larger than the average MOOC [3].

B. Extracting Topics and Q&A Tendencies

Two of the key steps prior to optimization are (1) topic extraction and (2) inference of the topic-wise seeking and disseminating tendencies. Here, we briefly analyze the results from these steps before moving to efficiency evaluation.

1) Topics: We implemented LDA using collapsed Gibbs sampling, through the `lda` library in Python. We empirically varied the number of topics $|\mathcal{K}|$ for each dataset, and inspected (i) the highest constituent words $\arg\max_x \phi_{k,x}$ and (ii) the support $f_k = \sum_n \theta_{n,k}/|\mathcal{N}|$ across the resulting topics with each choice of $|\mathcal{K}|$. We found that $|\mathcal{K}| = 10$ obtained both a reasonably high support f_k across topics (*i.e.,* ensuring each topic is well represented across posts) and disparity among the top words (*i.e.,* ensuring each topic is different). As a sample, Table II gives a summary of the results for the `algo` dataset, with the three words having highest f_k shown for each k. We see that the topics (i) are representative of likely discussions for each course (*e.g.,* $k=3$ is about data types, $k=10$ is about graphs), and (ii) are reasonably non-overlapping, with the exception of ubiquitous course words (*e.g.,* "number").

[9]The URL Handle has prefix `www.coursera.org/course/`.

Name (URL Handle)	Start	Weeks	Users	Threads	Posts
ml (ml-003)	4/13	12	4263	4217	25,481
comp (composition-003)	9/14	13	3013	4656	16,276
algo (algo-004)	7/13	8	1862	1256	8255
shake (virtualshakespeare-001)	4/15	5	958	1389	7484

TABLE I: Basic statistics of the four datasets used, each corresponding to a different Coursera course session. The start (m/yy), duration (weeks), and number of users, threads, and posts are given for each.

Results for the other datasets can be found in our technical report [9], and are qualitatively similar.

2) Seeking and disseminating: Fig. 1 shows the distributions of $d_{u,k}$ and $s_{u,k}$ across users in algo (showing those $d_{u,k}, s_{u,k} > 0$) for the two topics that have highest f_k. For plots on the other datasets, see our technical report [9].

Considering the 40 topics across the courses, we can make a few observations. For one, while the $d_{u,k}$ values tend to be shifted to the right relative to the $s_{u,k}$, the distributions are on the same order. This indicates that while there is enough dissemination overall, it needs to be allocated intelligently to meet the seeking tendency. Our efficiency evaluation will quantify how well the observed SLN perform in this regard.

Finally, we remark that the topics and seeking/disseminating tendencies could serve as useful analytics for a course instructor in their own right. It would help the instructor to identify topics with highest disparity between seeking and disseminating, which could be used to devise necessary interventions.

C. Efficiency Evaluation

Parameters. Referring to (3), the marginal benefit α_u of teaching relative to learning for user u depends on various factors, and is likely user-dependent. We treat $\alpha_u \sim U(0, \alpha_m)$ as a uniform random variable over $(0, \alpha_m)$, where $\alpha_m \in [0, 1]$ is chosen so that learning benefit is at least as high as the teaching benefit. The values of α_m, and the smoothing factor σ, will be swept across suitable values to quantify their effect. We set the deficit penalty to a default of $\alpha_\epsilon = 0.1$; in Sec. IV-D3, we will justify this choice by showing that even $\alpha_\epsilon = 0$ (*i.e.,* no penalty) does not cause the optimization to produce distributions of local utilities that are less fair.

Implementation. In Algorithm 1, each step was coded de-novo in Python using the numpy package, with the exception of the projection (12) for which Gurobi [16] was called from within Python. The simulations were run across five machines, each with 8 cores and 16 GB RAM. Due to the random nature of α_u, each choice of parameters was averaged over multiple simulation runs. We fix $\gamma = 0.5$ and $T = 0.01$.

Results. Fig. 2(a) shows the two efficiency measures η^g and η^F from (4),(5) as α_m is varied (see our technical report [9] for a plot of the corresponding values of F and g). Here, we have set $\sigma = 0$ (*i.e.,* no smoothing). In Fig. 2(b), we consider the effect of the smoothing parameter σ from the definition of the social network in (6), plotting the efficiency measures as σ varies from 1E-4 to 1. Here, $\alpha_m = 0.4$. These graphs are the subject of the following discussion:

1) Low efficiency SLNs: Referring to Fig. 2(a), even without any teaching benefit ($\alpha_m = 0$), for each dataset we can see that *the observed SLNs have low efficiencies*, *i.e.,* substantially less global utility than the optimal. For η^g (efficiency with respect to global utility alone), ml has the highest of $\eta^g = 82.0\%$ ($\hat{g} = 4.50$, $g^* = 5.52$), followed by algo at 75.9% ($\hat{g} = 2.52$, $g^* = 3.32$), then shake at 74.9% ($\hat{g} = 11.6$, $g^* = 15.5$), and finally comp with 71.1% ($\hat{g} = 6.03$, $g^* = 8.48$). η^F (efficiency with the deficit penalty included) is ordered in the same way, with somewhat higher values (89.6%, 88.3%, 76.9%, and 73.3% for ml, algo, shake, and comp). Already, we see that much can be gained through optimization.

k	$f_k(\%)$	$\arg\max_x \phi_{k,x}$	k	$f_k(\%)$	$\arg\max_x \phi_{k,x}$
1	9.62	array sort element	6	12.0	time python run
2	9.02	hash heap function	7	7.60	number find min
3	4.84	int arr integ	8	9.67	log number time
4	15.1	test answer code	9	16.5	algorithm program problem
5	6.13	point group studi	10	9.52	edg node graph

TABLE II: Summary of the topics extracted by LDA with $|\mathcal{K}| = 10$ for the algo dataset. Given for each topic k are its support f_k and its highest three constituent words $\arg\max_x \phi_{k,x}$.

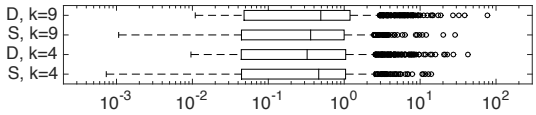


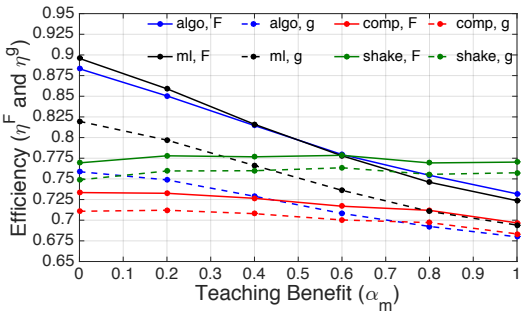
Fig. 1: Distributions of $s_{u,k}$ and $d_{u,k}$ inferred for the algo dataset, for the two topics $k = 4, 9$ with maximum support in Table II.

2) Lower efficiency SLNs for more teaching benefit: As α_m is increased in Fig. 2(a) to factor in teaching benefit, η^g and η^F are decreasing for each dataset, with the exception of shake for which they are roughly constant. This indicates that *the observed SLNs are less efficient with respect to a higher importance placed on teaching benefit*.

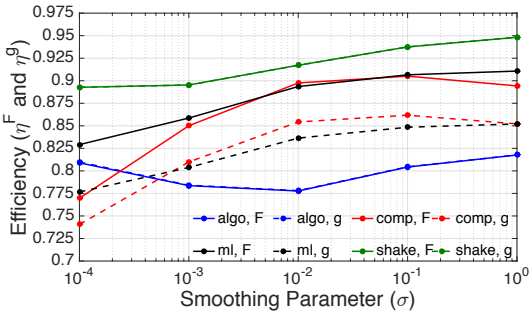
Notice that these decreases are particularly pronounced for algo and ml, for which η^g drops to 68.0% and 69.4% as $\alpha_m \to 1$. This implies that for these two datasets, there is generally less match between a user u's disseminating tendency $d_{u,k}$ and the seeking tendencies $s_{v,k}$ of those she tends to respond to (*i.e.,* those v with higher $w_{u,v}$) than between u's $s_{u,k}$ and the $d_{v,k}$ of those that respond to her (*i.e.,* those v with higher $w_{v,u}$). In other words, these SLN are formed around maximizing learning, rather than teaching, benefit. This difference can be attributed to the fact that people will tend to benefit from asking targeted questions in technical courses (algo, ml) rather than in humanities courses (comp, shake) where users would benefit from longer discussions.

Also, for each dataset, η^g is consistently lower than η^F. This indicates that if local utilities are ignored entirely, global utility can be pushed even higher. As we will see in Sec. IV-D3, local utilities are not substantially penalized through our framework anyway.

3) Higher efficiency SLNs for more smoothing: Referring to Fig. 2(b), we see that efficiency increases with the smoothing parameter σ (with the exception of algo). At $\sigma = 1$, η^g reaches 81.8%, 84.6%, 85.4%, and 94.8% for algo, comp, ml, and shake, respectively. Given that larger σ in (6) has the effect of making the weights $\hat{w}_{u,v}$ more uniform across v, this indicates that *SLNs where users respond impartially across neighbors tend to be more efficient*. However, across datasets except for shake, there is at least a 14% gap between the

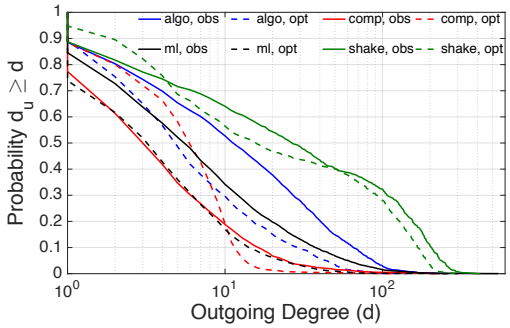


(a) Varying α_m.

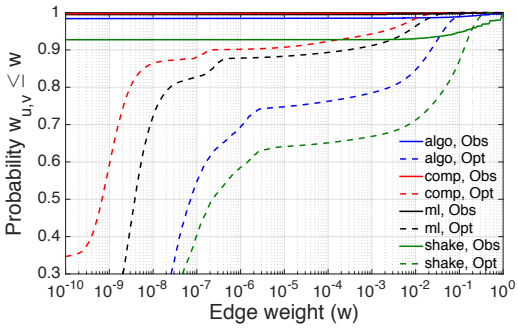


(b) Varying σ.

Fig. 2: Efficiency measures η^g and η^F as (a) the teaching benefit α_m and (b) the smoothing factor σ are varied. In (a), we see that even with $\alpha_m = 0$, the efficiencies are always below 1, highlighting the potential gains through optimization. Further, as α_m increases, the efficiency tends to drop, indicating that the networks are generally more formed around learning than teaching. In (b), we see that as σ increases, the networks tend to become more efficient, indicating that improvement can be obtained if users are more impartial in responding.



(a) Degree distributions, $P(d_u \geq d)$.



(b) Edge weight distributions, $P(w_{u,v} \leq w)$.

Fig. 3: Plots of (a) the outgoing degree distributions and (b) the edge weight distributions for observed and optimal networks. For each dataset, we can see that optimization makes the distributions more uniform, with (a) less users having large outgoing degrees and (b) many additional connections established between pairs of users.

smoothened SLN and the optimal with respect to g, *i.e.,* there is still substantial room for improvement through optimization.

D. Network Comparison

We perform an exploratory analysis to discover differences between the observed SLN and the optimal SLN. Here, we set $\sigma = 0$ for no smoothing and $\alpha_u = 0.4$.[10]

1) More uniform degree distributions: We first compare the degree distributions between the networks. To do so, we consider there to be a "link" from user u to v if and only if u is expected to respond to v at least once. Formally, with N_v as the number of times v posts, we define the adjacency matrix $A = [a_{u,v}]$, where $a_{u,v} = 1$ if $w_{u,v} \times N_v \geq 1$, else $a_{u,v} = 0$. With this, the (expected) outgoing degree of u is $d_u = \sum_v a_{u,v}$, *i.e.,* d_u is the number of unique users that u is expected to respond to.

Fig. 3(a) plots the degree distributions $P(d_u \geq d)$ across users for each network. Visually, we can see that *optimization tends to make the degrees more uniform*, reducing the number of users on the tail of the distribution. For example, take the proportion of users with $d_u \geq 20$: for `comp`, `ml`, `algo`, and `shake`, this fraction is reduced from 6.77% to 1.29%, 19.8% to 6.26%, 37.8% to 15.1%, and 54.3% to 46.7%, respectively.

2) More uniform edge weight distributions: The process of making the degree distributions more uniform involves adjusting the weights $w_{u,v}$ through optimization. In Fig. 3(b), we plot the CDF $P(w_{u,v} \leq w)$ of the edge weights for each dataset before and after this process.

Striking differences between the observed ($\hat{W}$) and optimized (W^*) SLN are apparent. In W^*, a vast amount of connections with $w^*_{u,v} > 0$ have been established between users, indicating that *optimization causes the edge weights to become more homogeneous*. For example, consider the case of `ml`. In $\hat{W}$, there are roughly 73K non-zero weights, which is only 0.40% of the potential user pairs in the network. Considering W^* on the other hand, 1.60M (8.83%) of the pairs are non-zero with $w^*_{u,v} \geq 0.001$.

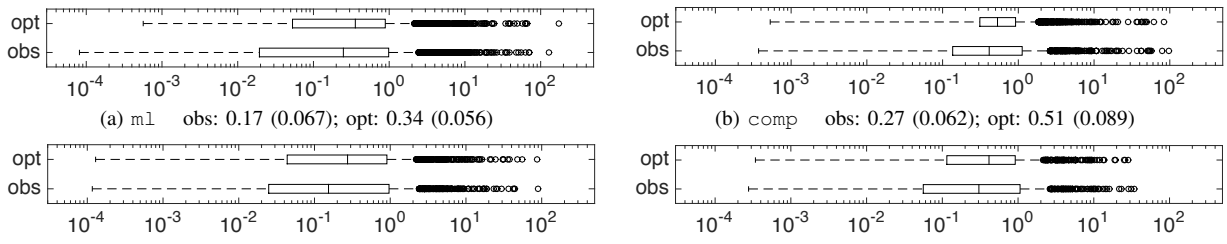
The distributions in Fig. 3 are also consistent with the finding in Fig. 2(b) that smoothing generally improves efficiency.

3) Optimization preserves fairness: As discussed in Sec. II-C, the deficit penalty α_ϵ in (3) controls a tradeoff between maximizing the global utility g and minimizing the effect on individual local utilities l_u. Here, we explore the effect of optimization on the l_u, by comparing the distributions of $r_u = l_u/g$ across users before and after.[11]

Fig. 4 gives boxplots of these values for each dataset with $\alpha_\epsilon = 0$, *i.e.,* the extreme case of no penalty for deficit. We can see that the distributions of the optimal are shifted to the right in each case, which indicates a tendency towards higher local utilities. To analyze the effect on the spread, we consider

[10]We observe the results to be qualitatively similar for other choices of α_u.

[11]Taking the ratio discounts the increase in global utility from optimization.



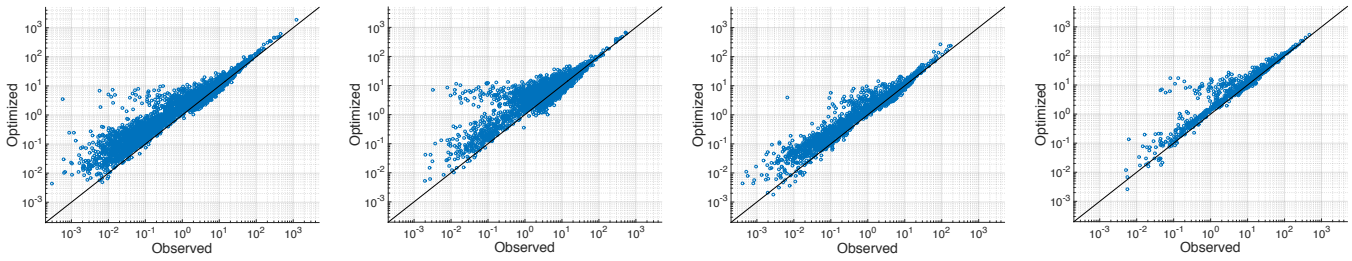
(a) `ml` obs: 0.17 (0.067); opt: 0.34 (0.056)

(b) `comp` obs: 0.27 (0.062); opt: 0.51 (0.089)

(c) `algo` obs: 0.10 (0.074); opt: 0.21 (0.072)

(d) `shake` obs: 0.17 (0.12); opt: 0.39 (0.16)

Fig. 4: Distribution of the ratio of local to global utility, $r_u = l_u/g$, for the observed (obs) and optimal (opt) SLN in each course, with $\alpha_\epsilon = 0$. The median (med) and Jain's Index (JI) of the plots are indicated in the caption, in the format: med (JI). Given that the JI do not change substantially, we conclude that the optimization at least preserves the fairness in local utility.



(a) `ml` (b) `comp` (c) `algo` (d) `shake`

Fig. 5: Plot of the local utility l_u for each user before (observed) versus after (optimized) optimization. The black line separates the regions of increased (above) and decreased (below) l_u. We can see that the majority of users have increased local utilities in each case.

the fairness of the r_u distributions through the standard Jain's Index (JI) metric.[12] The JI values are given in Fig. 4; we see that they do not change substantially after optimization for any of the datasets (actually increasing by 0.03 and 0.04 in `comp` and `shake`). Therefore, we conclude that while improving the global utility, *optimization also preserves fairness in the distribution of local utilities*.

4) Increases in local utilities: We are also interested in the differences between the l_u before (*i.e.,* $\hat{l}_u$) and after (*i.e.,* l_u^*) optimization, irrespective of g. In Fig. 5, we plot the effect of optimization on the local utilities, where each point is a user. Visually, it is apparent that *optimization improves local utility for the majority of users*. In particular, the percentage of users with $l_u^* \geq \hat{l}_u$ (*i.e.,* above the black line) is 85.5%, 82.2%, 87.2%, and 90.4% for `ml`, `comp`, `algo`, and `shake`. Also, the users who decrease to have larger $\hat{l}_u$ to begin with.

We also found that in general, users can obtain similar l_u in the optimized network with a smaller outgoing degree d_u. See our technical report [9] for a plot and more details.

E. Discussion and Future Work

Key observations. Large increases in global utility can be obtained by optimizing user participation (Fig. 2). Optimization also appears to not affect the spread of local utilities substantially (Fig. 4). The optimized networks have more homogeneous structures, with both outgoing degrees and edge weight distributions becoming more uniform (Fig. 3). The effect of this is more connected communities of users, causing the local utilities of the majority of users to increase (Fig. 5).

Realizing the optimal SLN. The focus of this work has been on modeling the optimal SLN. As discussed in Sec. I, the next step is to design and implement a mechanism to realize the optimized W^* networks in practice. Depending on the specific SLN application, there are a number of possibilities. Since most online forums already provide a news feed to direct user attention to new or popular posts, one way of influencing an SLN towards W^* would be to curate the news feed based on each user u's outgoing weights $w^*_{u,v}$ $\forall v$. This can be managed by updating u's news feed with a link to each new post created by v with probability $w^*_{u,v}$. Letting $\mathcal{C}_u = \{p_1, p_2, ...\}$ be the sequence of posts shown on u's page, Algorithm 2 shows one way the feeds $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ...\}$ can be updated from the W^* when a post q is made in thread r by user v at time t_c.[13] Here, each u has a maximum number of posts $c_{max}(u)$ to be displayed on her feed, and T is the maximum time q (created at $t(q)$) can be on the feed. The posts $p \in \mathcal{C}_u$ are prioritized according to $w^*_{u,v}$, where $v = \mu(p)$ is the creator of p.

Even with curated news feeds, users may not to follow the recommendations. It may be possible to design an incentive structure (*e.g.,* through awarding badges as in [2]) that rewards students who abide by their news feeds, or to automatically redirect the user to a post when the recommendation is made. In other SLN scenarios where engagement is compulsory (*e.g.,* in a classroom or in an enterprise social network [17]), it may be possible to require users to follow the recommendations.

Extensions to our model. Our work can be extended in several ways. First is regarding the definition of benefit in (1): this is assuming single-hop influence only, *i.e.,* from v directly to u; multi-hop influence could have the learning benefit in the form of $s_{u,k} \cdot f(\sum_v w_{v,u} i_{v,k})$ for user u, where $i_{v,k} = \sum_y w_{y,v} i_{y,k} + d_{v,k}$ is the (recursively defined) influence of user v on topic k. More generally, other types of data could be used to define a user's "benefit" in a SLN depending on

[12]The JI on n values varies between $1/n$ and 1. Higher JI is more fair.

[13]Given that the observed SLN evolves over time, the W^* can be recomputed at appropriate points (*e.g.,* once a day).

the application, *e.g.,* an improvement in exam performance (if available) would be a concrete way of measuring the benefit after optimizing the SLN. Second are the constraints in (3): (3b) could be modified to limit the decrease in local utility across users directly, and (3c) could be written in terms of a bound on the number of responses u is expected to provide. Third is the definition of the SLN: we have used the directed graph W formalized in Sec. III-A, but it is possible to work with *e.g.,* the user-thread graph instead if symmetry is valid. Fourth is investigating alternate inference algorithms in Sec. III-B, *e.g.,* learning-based methods for question detection [14].

V. Related Work

In recent years, there have been several studies proposing methods to help improve the quality of learning in MOOCs [3]. These include algorithms for *e.g.,* clickstream analysis and performance prediction to detect early dropouts [18], study partner recommendation [4], discussion thread ranking [3], and forum question recommendation [8]. Similar to [3], [4], [8], our work focuses on improving quality through MOOC forums. Specifically, we propose a framework for optimizing the allocation of a user's participation across the network; in this respect, our work is most related to [8], in which the authors propose a method for optimizing the allocation of users to questions, framed as a network flow problem. In their model, the specific content of each question is ignored, with the implicit assumption that participation implies expertise; our framework infers both question and answer tendencies for each user over a multidimensional topic space.

Other MOOC studies have focused specifically on the forum content [5] or graph structure [6] to gain insight into user behavior. Using the text of forum posts, [5] proposed an extension of non-negative matrix factorization to characterize students by learnt latent features. From a network perspective, [6] applied social network analysis techniques to identify significant interaction networks, detect communication vulnerability, and simulate the effect of information diffusion on an undirected user-user graph. Our work is different from these in that it takes a unified view of the content and structural aspects of MOOC forums, and views the flow of information as a directed graphical process (*i.e.,* $w_{u,v} \neq w_{v,u}$).

Regarding studies for online social networks in general, our work considers optimization of local and global utilities, as in [15] for recommender networks. Different from [15], our framework poses constraints specific to SLN, including multidimensional information spread. We also remark that the scale of our optimization (up to 18,000,000 variables) is much larger than those in existing work, posing unique computational challenges that are overcome in this paper.

VI. Conclusion

The proliferation of online (human) learning in recent years has made SLN an intriguing research area. We studied an important topic pertaining to SLN: the efficiency of information exchange between users. To do so, we proposed a framework which compares observed user benefit to that which can be obtained in an optimized, ideal SLN. Using our framework, we evaluated the efficiency of the discussion forums in four MOOC courses, in which we saw the potential gains that can be obtained through optimization. The main step for future work beyond the modeling presented here is to design mechanisms to enforce the optimized networks in practice.

Algorithm 2 Updating news feed based on the optimal SLN.

Input: v, r, $q \in \mathcal{P}_r$, $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ...\}$, c_{max}, t_c, T, W^*
for $u \in \mathcal{U} \setminus v$ **do**
  RM-LD($\mathcal{C}_u, t_c, T$) {RM-LD: remove any $p \in \mathcal{C}_u$ with $t_c - t(p) > T$}
  if Unif$(0,1) \geq w^*_{u,v}$ **then**
    APPEND($\mathcal{C}_u, q$) {append new post q to u's news feed $\mathcal{C}_u$}
    SORT($\mathcal{C}_u, w^*_u$) {sort $\mathcal{C}_u$ descending $\forall p \in \mathcal{C}_u$ based on $w^*_u(\mu(p))$}
    if $|\mathcal{C}_u| > c_{max}(u)$ **then**
      RM-ST($\mathcal{C}_u$) {RM-ST: remove the last (oldest) element from $\mathcal{C}_u$}
Return: $\mathcal{C}_u$ $\forall u$ {Updated news feed for each user u}

Acknowledgment

This work was in part supported by ARO grants W911NF-14-1-0190 and W911NF-11-1-0036. Additionally, we thank the reviewers and Liang Zheng for their valuable comments.

References

[1] C. G. Brinton and M. Chiang, "Social Learning Networks: A Brief Survey," in *CISS*. IEEE, 2014, pp. 1–6.
[2] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Engaging with Massive Online Courses," in *WWW*. ACM, 2014, pp. 687–698.
[3] C. G. Brinton, M. Chiang, S. Jain, H. Lam, Z. Liu, and F. M. F. Wong, "Learning About Social Learning in MOOCs: From Statistical Analysis to Generative Model," *IEEE Trans. Learning Technol.*, vol. 7, pp. 346–359, 2014.
[4] B. Xu and D. Yang, "Study Partners Recommendation for xMOOCs Learners," *Computational Intelligence & Neuroscience*, vol. 2015, 2015.
[5] N. Gillani, R. Eynon, M. Osborne, I. Hjorth, and S. Roberts, "Communication Communities in MOOCs," *arXiv:1403.4640*, 2014.
[6] N. Gillani, T. Yasseri, R. Eynon, and I. Hjorth, "Structural Limitations of Learning in a Crowd: Communication Vulnerability and Information Diffusion in MOOCs," *Scientific reports*, vol. 4, 2014.
[7] C. G. Cortese, "Learning Through Teaching," *Management Learning*, vol. 36, no. 1, pp. 87–115, 2005.
[8] D. Yang, D. Adamson, and C. P. Rosé, "Question Recommendation with Constraints for Massive Open Online Courses," in *RecSys*. ACM, 2014, pp. 49–56.
[9] Technical report. www.princeton.edu/~cbrinton/SLNopt_tech.pdf.
[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[11] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth, "Unsupervised Modeling for Understanding MOOC Discussion Forums: A Learning Analytics Approach," in *LAK*. ACM, 2015, pp. 146–150.
[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
[13] G. Zhou, J. Zhao, T. He, and W. Wu, "An Empirical Study of Topic-Sensitive Probabilistic Model for Expert Finding in Question Answer Communities," *Knowledge-Based Systems*, vol. 66, pp. 136–145, 2014.
[14] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun, "Finding Question-Answer Pairs from Online Forums," in *SIGIR*. ACM, 2008, pp. 467–474.
[15] F. M. F. Wong, Z. Liu, and M. Chiang, "On the Efficiency of Social Recommender Networks," in *INFOCOM*. IEEE, 2015, pp. 2317–2325.
[16] Gurobi Optimization Inc. (2014) Gurobi Optimizer Reference Manual. [Online]. Available: www.gurobi.com
[17] J. Cao, H. Gao, L. E. Li, and B. Friedman, "Enterprise Social Network Analysis and Modeling: A Tale of Two Graphs," in *INFOCOM*. IEEE, 2013, pp. 2382–2390.
[18] C. G. Brinton and M. Chiang, "MOOC Performance Prediction via Clickstream Data and Social Learning Networks," in *INFOCOM*. IEEE, 2015, pp. 2299–2307.