On the Efficiency of Online Social Learning Networks

Christopher G. Brinton, *Member, IEEE,* Swapna Buccapatnam, Liang Zheng, *Member, IEEE,* Da Cao, Andrew Lan, Felix M. F. Wong, *Member, IEEE,* Sangtae Ha, *Senior Member, IEEE,* Mung Chiang, *Fellow, IEEE,* and H. Vincent Poor, *Fellow, IEEE*

***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 method in which SLN efficiency is determined by comparing user benefit in the observed network to a benchmark of maximum utility achievable through optimization. Our method defines the optimal SLN through utility maximization subject to a set of constraints that can be inferred from the network, and given multiple solutions searches for the one closest to the observed network so as to require the least amount of change to user behavior in practice. Through evaluation on four MOOC discussion forum datasets and optimizing over millions of variables, we find that SLN efficiency can be rather low (from 76% to 90% 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 propose an algorithm for realizing the optimal network through curated news feeds in online SLN.**

I. Introduction

The term Social Learning Network (SLN) encapsulates a range of scenarios in which a number of people learn from one another through structured interactions. The proliferation of online communication has given rise to a number of SLN applications, including Question and Answer (Q&A) sites (*e.g.,* Quora), enterprise social networks (*e.g.,* Jive, Yammer), and platforms for online education which have in turn created learning networks among askers/answerers, employees, and students, respectively [2].

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 [3]. But they also suffer from low completion rates, often attributed factors such as low teacher-to-student ratios, a lack of face-to-face interaction, and asynchronous scheduling [4].

In an effort to alleviate some of these problems, MOOC platforms typically 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 [2]. While MOOC forums can be monitored by instructors and teaching staff, the large volume of students (*e.g.,* > 4.2K for one of the datasets in Table I) and posts made by students overall (*e.g.,* $>$25K for the same dataset) makes it infeasible for the staff to handle each individual 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 topical experts and those seeking information regarding the same topics [5]. It is unclear whether the SLN in MOOC forums tend to form in such an ideal manner, though, especially given that each individual student may only generate a handful of posts throughout the lifetime of a course [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?*

A. SLN Efficiency Modeling Methodology

To study our research questions, we propose a novel methodology for modeling SLN efficiency (Sec. II). The objective of our methodology is to compare the benefit obtained by users in the network as it exists presently (called the Observed SLN) to that which can theoretically be obtained through optimization (called the Ideal SLN). The key components of this methodology are outlined in Fig. 1. In what follows, we will introduce these components, highlighting the challenges involved and contributions made:

Observed SLN. The Observed SLN is gathered through network identification (Sec. II-A, III-A). The key challenge

C. Brinton and D. Cao are with Advanced Research at Zoomi Inc. e-mail: {chris.brinton, da.cao}@zoomiinc.com

S. Buccapatnam is with AT&T Labs Research. e-mail: sb646f@att.com.

A. Lan and V. Poor are with the Department of Electrical Engineering, Princeton University. e-mail: {andrew.lan, poor}@princeton.edu.

L. Zheng and F. Wong are independent researchers. emails: {liangz, mwthree}@princeton.edu.

S. Ha is with the Department of Computer Science, University of Colorado Boulder. email: sangtae.ha@colorado.edu

M. Chiang is with the College of Engineering at Purdue University. email: chiang@purdue.edu

Appendices A and B are available as online supplementary material.

This work was presented in part at the 2016 IEEE Conference on Computer Communications (INFOCOM) [1].

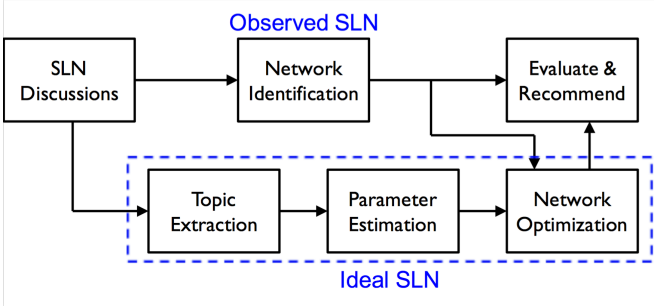


Fig. 1: High-level block diagram of the SLN optimization methodology developed in this paper.

here is quantifying the notion of a link between two users from their observed discussions, given that the number of posts made by individual users can be limited. We develop a probabilistic message-passing approach with smoothing for link weight inference.

Ideal SLN. To obtain the Ideal SLN, we first model each user individually as possessing certain levels of seeking (*i.e.,* question asking) and disseminating (*i.e.,* question answering) tendencies (Sec. II-A, III-B). A challenging consideration in this Parameter Estimation stage is how to infer whether a student is asking or answering a question. Further, this is ideally done on a topic-specific basis to account for the fact that students may have different needs on different topics [4]. As a result, our Topic Extraction process develops a set of latent educational topics for the course, using the student-generated discussion text as input.

With the user parameters and observed SLN in hand, Fig. 1 proceeds to Network Optimization. The purpose of this is to search for a social structure that maximizes global user benefit while at least preserving the benefits of each individual (Sec. II-C, III-C). In terms of knowledge transfer, we identify at least two ways that a user will benefit from an SLN: (i) by learning from others directly, and (ii) by explaining topics to others, *i.e.,* learning by teaching [6]. The components of our framework give us a natural way of quantifying these two benefits: (i) assess the match between a user's seeking tendency and the disseminating tendencies of her neighbors, and (ii) assess the match between the user's disseminating tendencies and the seeking tendencies of her neighbors. In an ideal setting, both of these would be as large as possible. Therefore, the optimization we develop 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 user will participate in the forums is constrained by her own resource limitations.

There is some existing work on studying the content of MOOC forums (*e.g.,* [7]) and some studying the graph structure (*e.g.,* [8]); our work considers a unified view of both components. Also, different from the optimization of users to questions proposed in [9], our method accounts for the difference between seeking and disseminating tendencies of users over a multidimensional topic space.

Algorithms for optimization. The Network Optimization poses two computational challenges that must be overcome in the implementation (Sec. III). First is one of *scalability*: it has several million variables corresponding to the weights in a directed user-to-user graph, making it intractable for standard convex optimization solvers [10]. As a result, we derive a projected gradient descent algorithm for this problem (Sec. III-C) in which the projection step becomes quadratic, and in turn derive a proximal alternating direction method of multipliers (ADMM) algorithm to perform the projection (in App. B, available as online supplementary material).

The second challenge is *non-uniqueness*: the optimization problem has non-unique solutions for realistic parameter values (proven in App. A, available as online supplementary material), posing the question of which solution is most useful and how that can be obtained. Intuitively, the network that is the shortest distance away from the observed is desirable because it requires the least amount of change in user behavior to realize in practice. As a result, we introduce a regularization term to the objective that forces the algorithm to converge to this particular solution.

B. Performance Evaluation

After formalizing our methodology, 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 can be rather low, ranging from 76% to 90% 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.

We will also discuss the implications of these results to MOOCs and social learning in general (Sec. IV, V). They indicate that SLN today tend to form in an inefficient manner, with substantial room for improvement through optimization without incurring the cost associated with adding additional instructors to each course. Given each user's finite capacity for contributing to the network, they should spread their participation more uniformly across questions they have expertise in, rather than focusing on those asked by particular users they have previously communicated with. The edge weights from our optimization indicate how to form these connections.

C. Improving SLN Efficiency

In general, there are three steps involved in improving the efficiency of an SLN: (i) defining the ideal SLN through optimization, (ii) solving for the optimal SLN, and (iii) implementing the optimized network in practice to observe the improvements. As the focus of this work is formulating and evaluating a model for efficiency, our main contributions are the first two steps, allowing us to quantify the gains that will be obtained when the modeling assumptions hold. In the final part of this paper (Sec. V), we outline the requirements

and challenges of the third step for future work, and propose an algorithm for realizing the optimized network by curating users' news feeds based on recommended interactions.

II. SLN Efficiency

To quantify the efficiency of an SLN, we pose the following question: *How much are users benefiting from the observed network topology relative to how much they could benefit from an optimized topology?* In this section, we formulate our efficiency methodology, 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 will 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. We 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. Note that $w_{u,v} \neq w_{v,u}$ in general, *i.e.,* the matrix is asymmetric.

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

Seeking and disseminating. A user will have some tendency towards disseminating information (*i.e.,* providing answers or facts about the material) and/or seeking information (*i.e.,* asking questions about material) on a topic-by-topic basis. 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 some SLN applications, it is reasonable to assume that only one of $s_{u,k}$ and $d_{u,k}$ will be positive for a given u, k pair (*e.g.,* on some Q&A sites, like Stackoverflow, there is a distinction between those who answer vs. ask questions on a topic). More generally, and for MOOC in particular, a user may both seek and disseminate on a topic; for example, a student may have a question she posted answered and then herself answer someone's similar question later. As a result, we do not impose any such restrictions on S and D.

Will discuss our algorithms for inferring W, $\mathcal{K}$, S, and D from SLN data in Sec. III.

B. Utility Modeling of SLN (B, Φ, Ψ)

We now formalize user benefit and utility in an SLN.

Benefit. We identify two types of user benefit:

(i) 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 $f(\cdot)$ 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.

(ii) Teaching benefit. In peer-to-peer learning, users also draw benefit from acting as teachers to others, *i.e.,* from learning by teaching [2], [6]. 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 $f(\cdot)$ 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.

Now, 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; we will discuss the approach we take for setting α_u in Sec. IV. We choose $f(x) = \log(1 + x)$ because it is a standard function used to capture diminishing marginal utility.

SIDR and DISR. For each u, k pair, we also define the Seeking to Incoming Disseminating Ratio (SIDR)

$$\phi_{u,k} = \frac{s_{u,k}}{\sum_v w_{v,u} d_{v,k}} \tag{2}$$

and the Disseminating to Incoming Seeking Ratio (DISR)

$$\psi_{u,k} = \frac{d_{u,k}}{\sum_v w_{u,v} s_{v,k}}, \tag{3}$$

with $\Phi = [\phi_{u,k}]$ and $\Psi = [\psi_{u,k}]$. A smaller SIDR $\phi_{u,k}$ implies that u's seeking tendency on topic k has higher satisfaction from the incoming disseminating tendencies of her neighbors. A smaller DISR $\psi_{u,k}$ implies that u's disseminating tendency on k is being used to satisfy more of her neighbor's seeking tendency. These will be used as constraints in Sec. II-C.

Utility. We quantify two different types of utility:

(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

$$l_u = \sum_k b_{u,k}. \tag{4}$$

(ii) Global utility: The global utility g is defined as the average local utility across users. From (1),

$$g_{\alpha_u} = \frac{1}{|\mathcal{U}|} \sum_{u,k} b_{u,k}. \tag{5}$$

C. Optimizing SLN

From the definitions in Sec. II-A and II-B, our optimization seeks the combination of weights W in the SLN that will (i) maximize the global utility g of the SLN while (ii) minimizing

the impact on – and potentially improving – the benefits that are already provided to specific user-topic pairs from the observed network $\hat{W}$. Formally, for fixed seeking S and disseminating D tendencies, our optimization over W is given as follows:

$$\underset{W}{\text{maximize}} \quad g_{\alpha_u}(W) \tag{6a}$$
$$\text{subject to} \quad \Phi(W) \leq C_s\hat{\Phi} \tag{6b}$$
$$\Psi(W) \geq C_d\hat{\Psi} \tag{6c}$$
$$\mathbf{0} \leq W \leq \mathbf{1},\ \text{diag}(W) = \mathbf{0} \tag{6d}$$

There are two linear constraints (besides bounds):
(i) Preserving incoming information (6b): $\Phi(W) = [\phi_{u,k}(W)]$ denotes the SIDR resulting from a given W for each u, k pair. On the right, $\hat{\Phi} = [\hat{\phi}_{u,k}]$ is the matrix of observed SIDR from $\hat{W}$, *i.e.,* $\hat{\phi}_{u,k} = \phi_{u,k}(\hat{W})$. If $\phi_{u,k}(W) < \hat{\phi}_{u,k}$, this means that the amount of information transferred to user u on topic k (*i.e.,* $\sum_v w_{v,u}d_{v,k}$) in W is larger than it was in $\hat{W}$; if it holds $\forall k$, then the local utility l_u in (4) will increase. $C_s > 0$ is a tightness parameter enforcing that the SIDR after optimization cannot exceed C_s times what it was before. If $C_s < 1$, we are requiring a tighter bound than what was observed, whereas if $C_s > 1$, we allow any particular SIDR to rise if needed.

A direct lower bound on the benefits $b_{u,k}$ (or local utilities l_u) may appear to be a more natural form for (6b). This, however, would introduce a concave constraint to (6), which would necessitate the use of interior point methods that are not generally scalable to the millions of variables we consider here [10]. The linear constraint set will allow us to derive a scalable projected gradient descent procedure in Sec. III-C to solve (6) with convergence guarantees. With (6b) in its proposed form, we will investigate the impact of optimization on the local utilities through experimentation in Sec. IV-D.
(ii) Balancing load (6c): By (1), a higher incoming seeking score $\sum_v w_{u,v}s_{v,k}$ leads to a higher teaching benefit $b_{u,k}$. But each user also 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 this constraint, DISR is restricting the amount of seeking tendency a user is addressing (*i.e.,* $\sum_v w_{u,v}s_{v,k}$) with her dissemination (*i.e.,* $d_{u,k}$) to not exceed $C_d > 0$ times what it was observed to be already. $\hat{\Psi} = [\hat{\psi}_{u,k}]$ is the matrix of DISR from the observed network, *i.e.,* $\hat{\psi}_{u,k} = \psi_{u,k}(\hat{W})$. If $C_d > 1$, we are requiring users to participate less than what was observed. If $C_d < 1$, on the other hand, we allow any particular DISR to drop (*i.e.,* user participating more) if needed. $C_d = 1$ would be a conservative selection, because under the optimized network, we can expect that users will be incentivized to participate more.

Note that an upper bound on the sum of the outgoing weights, *i.e.,* $\sum_v w_{u,v} \leq \bar{w}_u$, would not exactly capture the participation capacity for (6c): each user v demands a different teaching load on a particular topic, quantified by $s_{v,k}$. Varying C_d will also allow us to evaluate the effect of potential errors in estimating $\hat{\Psi}$, *i.e.,* having underestimated ($C_d < 1$) or overestimated ($C_d > 1$) user load from the data.

Definition 1 (SLN efficiency)**.** *Let $g^\star_{\alpha_u,C_s,C_d}$ be the value that (6a) takes for an optimal solution $W^\star$ of (6) for fixed S, D, and parameters α_u, $C_s \geq 1$, $C_d \leq 1$.*[1] *The efficiency of the SLN for its observed matrix $\hat{W}$ is quantified as*

$$\eta^g_{\alpha_u,C_s,C_d} = g_{\alpha_u}(\hat{W})/g^\star_{\alpha_u,C_s,C_d}. \tag{7}$$

In other words, η^g is the fraction of the global utility achievable in the optimized network that is already obtained by the observed network $\hat{W}$.

Nonuniqueness. Note that in (6), the objective is concave and the constraints are linear in W, making this a convex optimization problem. But in App. A, we prove that (6a) is not a strictly convex function, and that any optimal solution $W^\star$ to (6) is not unique under realistic conditions on S and D. As a result, the algorithm that we propose to solve this optimization in Sec. III will find the optimal solution $W^\star_I$ closest to $\hat{W}$ so as to induce the least amount of change in user behavior from the observed network.

III. Inference and Optimization Algorithms

To compute the efficiency (7), we need to determine the observed social network ($\hat{W}$), solve (6) to obtain an optimized SLN (W^*), and find the global utilities $g(\hat{W})$ and $g^\star$. To solve the optimization and find the utilities, we must also infer the seeking (S) and disseminating (D) tendencies, which lead to the observed SIDR ($\hat{\Phi}$) and DISR ($\hat{\Psi}$) matrices. In this section, we will describe how we infer these quantities and solve (6).
Forum structure. We first develop terminology for the structure of forums on MOOC platforms. Typically, each course has a single forum comprised 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; for our purposes, we do not distinguish between posts and comments, and refer to them both as posts. If comments were always written in response to posts, then the relationship between them could be useful for inferring the observed SLN in Sec. III-A and Q&A tendencies in Sec. III-B2; however, MOOC users do not abide to this structure consistently [4].

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 $u(p)$, creation time $t(p)$, and text $x(p)$ written by $u(p)$. 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 ($\hat{W}$)

The first component of the SLN 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 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 $\sum_r \min(|\mathcal{P}_{r,u}|, |\mathcal{P}_{r,v}|)$ [8].

[1] $\hat{W}$ is only in the feasible region of (6) for these ranges of C_s and C_d.
[2] We will drop subscripts like r when the context makes it clear.

Name	Title	URL Handle	Type	Start	Weeks	Users	Threads	Posts
ml	Machine Learning	ml-003	T	4/29/13	12	4263	4217	25,481
comp	English Composition I	composition-003	H	9/22/14	13	3013	4656	16,276
algo	Algorithms: Design and Analysis I	algo-004	T	7/01/13	8	1862	1256	8255
shake	Shakespeare in Community	virtualshakespeare-001	H	4/22/15	5	958	1389	7484

TABLE I: Basic statistics of the four datasets used, each corresponding to a different Coursera course session. The title, URL handle, type (Technical (T) or Humanities (H)), start date (m/dd/yy), duration (weeks), and number of users, threads, and posts are given for each.

But analyzing the user-thread bipartite graph directly leads to a symmetric $\hat{W}$: while this may be a valid assumption for friendship networks [4], 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 message-passing formulation: *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.

1) Computing $\hat{w}_{u,v}$: Formally, let $n_{u,v}$ be the number of times that u posts after v, with $n_{u,u} = 0$ (our algorithm for obtaining $n_{u,v}$ is given next). 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 will be diverse among forum users, giving each u varying opportunities to respond to v in the first place, we apply a standard shrinkage estimator [11], [12] to smoothen the $\hat{w}_{u,v}$ towards u's overall response rate $\sum_j n_{u,j} / \sum_j N_j$:

$$\hat{w}_{u,v}(\sigma) = \frac{n_{u,v} + \sigma(\sum_j n_{u,j} / \sum_j N_j) N_{max}}{N_v + \sigma N_{max}}, \qquad (8)$$

where σ is the smoothing parameter and $N_{max} = \max_i N_i$ is the maximum number of times a user posted.

Note that (8) with $\sigma = 0$ gives the non-smoothened version $\hat{w}_{u,v} = n_{u,v}/N_v$, i.e., the fraction of posts u was observed to write in reply to v. All else constant, as σ is increased, a user is expected to spread his/her overall response rate more uniformly among the other users in the SLN. We will consider the effect of smoothing in Sec. IV, retaining $\sigma = 0$ as the default value corresponding to the observed SLN.

2) Computing $n_{u,v}$: In computing $n_{u,v}$, the key is to ensure that within a thread r, (i) u is counted as responding at most once to each post made by v, and (ii) each post made by u is counted as a response to v at most once. Let $\mathcal{I}^r_{u,v}$ be the set of post-response pairs (from u to v) in thread r. Starting with $\mathcal{I}^r_{u,v} = \emptyset$, for each $q \in \mathcal{P}_{r,v}$, (p, q) is added to $\mathcal{I}^r_{u,v}$ if the following conditions are satisfied: $\mu(q) = v$, $t(p) > t(q)$, $(y, q) \notin \mathcal{I}^r_{u,v}$ $\forall y \in \mathcal{P}_{r,u}$, and $(p, z) \notin \mathcal{I}^r_{u,v}$ $\forall z \in \mathcal{P}_{r,v}$. These conditions ensure that each of p and q occurs only once in $\mathcal{I}^r_{u,v}$, *i.e.,* u responds at most once to each $q \in \mathcal{P}_{r,v}$, and each $p \in \mathcal{P}_{r,u}$ is counted as a response to v at most once. With this, $n_{u,v} = \sum_r |\mathcal{I}^r_{u,v}|$.

With this specification of $n_{u,v}$, the requirement for post p to be counted as a response to post q is less stringent than *e.g.,* viewing p as a response to q only if it is an explicit comment to q. Indeed, as stated, MOOC users do not abide by the structure of posts vs. comments consistently [4]. Also, by taking this approach, the number of counted responses from u becomes larger than the number of posts made by u, i.e., $\sum_v n_{u,v} > N_u$; since the average number of posts per user tends to be small in MOOC, including in our datasets (see Table I), this improves the sample size for parameter estimation.

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

Another component of SLN is the seeking $S = [s_{u,k}]$ and disseminating $D = [d_{u,k}]$ tendencies.[3] 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 (1) and (2).

1) Topic extraction: We employ Latent Dirichlet Allocation (LDA), a popular generative model for topic extraction from a collection of documents [14]. LDA has been applied to discussion forums in several studies, *e.g.,* in [5], [15].

Formally, consider a collection of documents $\mathcal{N}$, where each $n \in \mathcal{N}$ is written as a series of word indices $d_n = (d_{n,1}, d_{n,2}, ...)$, $d_{n,j}$ being an index into the dictionary $\mathcal{X}'$ (we will discuss the choice of n and $\mathcal{X}'$ further below). Under LDA [14], 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 $\beta = [\beta_{k,x}] \in [0,1]^{|\mathcal{K}| \times |\mathcal{X}'|}$ are such that $\beta_{k,x}$ 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 $\beta_k = \{\beta_{k,1}, ..., \beta_{k,|\mathcal{X}'|}\}$.[4]

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 [15]) since there can 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,[5] (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.

[3] These can be inferred independent of specific k, similar to in [9], but this is undesirable because the topics discussed in MOOC are diverse [4], [13].

[4] Note that the multinomials here are single trials, as each $w_{n,j}$ is generated from a single topic k.

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

2) Question/answer tendency: With the post-topic distributions θ, the next step in 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 $\theta_{p,k}$:

$$q_{u,r,k} = \frac{\sum_{p \in \mathcal{P}_{r,u}} \theta_{p,k} \cdot Q(p)}{\sum_{p \in \mathcal{P}_{r,u}} \theta_{p,k}}. \tag{9}$$

In general, question detection algorithms can be divided into two groups: rule-based methods, *e.g.,* whether a question mark is present [5], and learning-based methods, *e.g.,* classifiers analyzing sequences of parts of speech [16]. In our work, we apply a series of rule-based methods, as some of them have demonstrably high quality; for example, in [16], question-mark detection had an F1-score (F1) of roughly 85% on two datasets.[6] 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 $\in x(p)$".[7] $Q(p)$ 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; for all other posts ($p \neq 1$), we required $?_p$ to be true, and at least one question-type word to protect against false positives.

Small experiment. To test our intuitions, we obtained human generated labels on some posts to compare with our $Q(p)$. To do so, we gathered all threads from our datasets in Sec. IV that had between 10 and 25 posts, and chose 50 threads randomly from this set. This yielded a total of 749 posts. We then recruited three individuals to label each post as either seeking information, denoted $Q_o(p) = 1$, or providing information, denoted $Q_o(p) = 0$. For each p, we took the majority vote among the three labels as the true $Q_o(p)$.

We make two observations on the results: First, only 19.6% of the 749 total posts had $Q_o(p) = 1$, whereas 52.0% of the 50 posts with $p = 1$ had $Q_o(p) = 1$. This suggests that while a first thread post is a question only roughly half of the time, these posts have significantly higher chance of $Q_o(p) = 1$ than do those with $p \neq 1$. Second, in comparing the $Q(p)$ and $Q_o(p)$ $\forall p$, our method obtains an accuracy of 0.86 and an F1 of 0.65. This accuracy is quite high, but the F1 is lower than those cited in *e.g.,* [16] for other methods, which emphasizes the importance of averaging in (9) to reduce noise.

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

$$d_{u,k} = \sum_r (1 - q_{u,r,k}) \cdot \log(1 + \sum_{p \in \mathcal{P}_{r,u}} \theta_{p,k} \cdot |x'_p|), \tag{10}$$

[6]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 [11].

[7]5W1H are standard question words. We observed that urgency/gratitude (UG) words tend to appear frequently in question posts too.

Algorithm 1 Projected gradient descent algorithm to solve (6).

Input: $\hat{W}$, S, D, $\hat{\Phi}$, $\hat{\Psi}$, α_u $\forall u$, C_s, C_d, $N = |\mathcal{U}|$, T
Initialize: $\tilde{g}[-1] \leftarrow -\infty$, $W[0] \leftarrow \hat{W}$, $n \leftarrow 0$, γ
$\tilde{g}[0] \leftarrow \tilde{g}(W[0])$
while $(\tilde{g}[n] - \tilde{g}[n-1])$ / $|\tilde{g}[n-1]| \geq T$ **do**
  $W'[n+1] \leftarrow W[n] + \gamma[n] \cdot \nabla\tilde{g}(W[n])$ {$\nabla\tilde{g}(W[n])$ from (13)}
  $W[n+1] \leftarrow P(W'[n+1])$ {P from (14)}
  $F[n+1] \leftarrow F(W[n+1])$
  $n \leftarrow n + 1$
Return: $W^* = W[n]$

$$s_{u,k} = \sum_r q_{u,r,k} \cdot \log(1 + \sum_{p \in \mathcal{P}_{r,u}} \theta_{p,k} \cdot |x'_p|), \tag{11}$$

where $q_{u,r,k}$ is from (9) and x'_p, $\theta_{p,k}$ are the sequence of words and post-topic distributions from Sec. III-B1. The inclusion of text length $|x'_p|$ here captures the fact that longer posts tend to contain more information; despite the fact that MOOC users make only 5-10 posts each on average [4], these estimators can still reveal substantial differences in tendencies between users, as we will see in Sec. IV-B. 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.

SIDR and DISR ($\hat{\Phi}$ and $\hat{\Psi}$): Out of the quantities needed in (6) and (7), we now have methods to infer S, D, and $\hat{W}$. Only $\hat{\Phi}$ and $\hat{\Psi}$ remain, which can now be obtained from (2) and (3) using $\hat{W}$, S, and D.

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

The final component to develop is the algorithm to solve (6) for W^*. As (6) is convex, it can be solved numerically in theory by standard algorithms such as interior point methods. We approach the solution otherwise for two reasons. First is an issue of *scalability*: the number of variables in our problem is $|\mathcal{U}| \times (|\mathcal{U}| - 1)$; with just 1K users (which is on the order of the smallest dataset in Table I), there are already almost 1M variables, which makes these standard methods computationally intractable [10]. Second is *non-uniqueness*: the problem can have multiple optimal solutions, as discussed in Sec. II-C. We desire a method that will obtain the $W_I^\star$ closest to $\hat{W}$, to minimize the impact on user behavior.

To force $W_I^\star$, we replace the objective in (6) with the following $\tilde{g}_{\alpha_u}(W)$, introducing a regularization term:

$$\tilde{g}_{\alpha_u}(W) = g_{\alpha_u}(W) - \lambda \|W - \hat{W}\|_F, \tag{12}$$

where $\|W - \hat{W}\|_F$ can be regarded as a convex loss function which penalizes solutions that are far from the observed user-to-user network $\hat{W}$.[8] Then, for a scalable solver, we derive a projected gradient descent method for the optimization problem. In this method, three steps are repeated in sequence: Gradient, Projection, and Objective. The pseudocode is given in Algorithm 1, and the individual steps are as follows:

[8]Other standard convex loss functions are also possible; the effect is on the step size in gradient descent, which must be chosen for convergence [17].

k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$	k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$
1	5.72	grad theta1 theta2	6	8.84	train set data
2	8.48	time data learn	7	12.2	vector matrix loop
3	9.84	octav file error	8	14.8	code problem work
4	8.12	theta function sum	9	13.4	learn machin class
5	10.6	featur data regress	10	8.03	cost theta function

(a) `ml`

k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$	k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$
1	14.8	write writer school	6	12.7	project feedback submit
2	3.93	imag expertis pictur	7	7.49	practic coyl talent
3	10.8	argument paragraph expertis	8	5.36	live world love
4	10.3	write time idea	9	12.7	english write languag
5	16.4	write read writer	10	5.57	work peopl educ

(b) `comp`

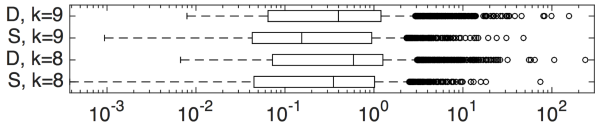
k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$	k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$
1	9.62	array sort element	6	12.0	time python run
2	9.02	hash heap function	7	7.60	number find min
3	4.84	int arr integ	8	9.67	log number time
4	15.1	test answer code	9	16.5	algorithm program problem
5	6.13	point group studi	10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

(c) `algo`

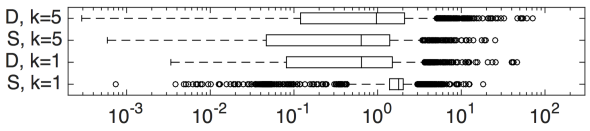
k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$	k	$f_k(\%)$	$\arg\max_x \beta_{k,x}$
1	16.2	shakespear read play	6	14.6	read shakespear post
2	10.6	romeo juliet love	7	4.26	beatric benedick strong
3	7.39	shakespear hamlet play	8	17.1	play word shakespear
4	7.19	love hermia play	9	6.25	light night scene
5	9.6	play version film	10	6.84	dream word love

(d) `shake`

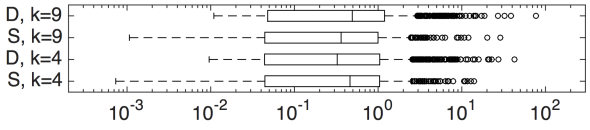
TABLE II: Summary of the topics extracted by LDA for each course, with $|\mathcal{K}| = 10$. Given for each k are the support f_k and the highest three constituent words x. We see that the topics are representative of likely discussions given the course context, and that they tend to be non-overlapping, with the exception of certain, obvious words.



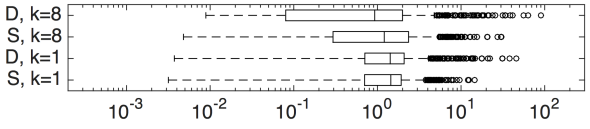
(a) `ml`



(b) `comp`



(c) `algo`



(d) `shake`

Fig. 2: Distributions of seeking ($s_{u,k}$) and disseminating ($d_{u,k}$) tendencies inferred for each dataset, for the two topics k with maximum support for each course (see Table II). In each box, we only consider non-zero values of $d_{u,k}$ and $s_{u,k}$. From this sample, we can see that the $d_{u,k}$ tend to be slightly larger than the $s_{u,k}$, but that they are on the same order, implying there is typically sufficient disseminating tendency to match the questions posted on the topics, if it is allocated efficiently.

1) Gradient step: Here, the gradient of (12) must be computed with respect to each $w_{u,v}$. It is easy to show that

$$\frac{\partial \tilde{g}}{\partial w_{u,v}} = \frac{1}{|\mathcal{U}|} \sum_k \left(\frac{d_{u,k} s_{v,k}}{1 + \sum_i w_{i,v} d_{i,k}} + \frac{\alpha_u d_{u,k} s_{v,k}}{1 + \sum_j w_{u,j} s_{j,k}} \right) - \lambda \frac{w_{u,v}}{\|W - \hat{W}\|_F}. \tag{13}$$

In Algorithm 1, the procedure moves in the direction of the gradient $\nabla \tilde{g}$ in each iteration, by the step size $\gamma[n]$, which is selected via backtracking line search [17].

2) Projection step: The solution from the gradient update is then projected onto the feasible region of (6). Since the constraints are affine, this problem can be cast as a linearly-constrained quadratic program. Formally, with W' as the matrix of variables before projection, we solve:

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & ||W - W'||_F^2, \\ \text{subject to} \quad & \text{Constraints (6b)-(6d).} \end{aligned} \tag{14}$$

We derive an alternating direction method of multipliers (ADMM) algorithm to solve (14); details of that are given in App. B available as online supplementary material. Note that since the constraints in (6) are linear, ADMM has convergence guarantees [18], whereas it would not if we had bounded the concave benefit terms $b_{u,k}$ directly in (6b) as discussed in Sec. II-C. In Algorithm 1, the function P refers to solving (14).

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

IV. Datasets and Results

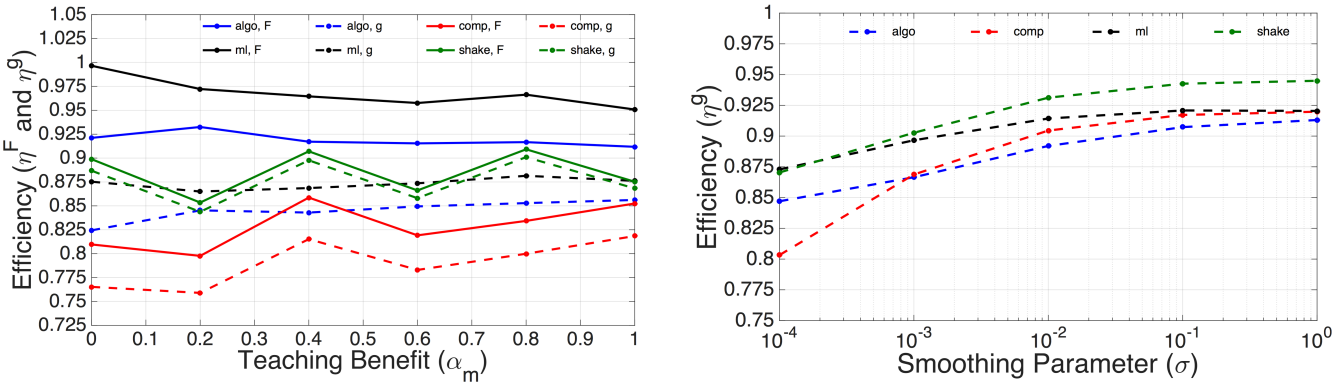
In this section, we evaluate the efficiency of four MOOCs, and compare the properties of the observed and optimal SLN.

A. Datasets

We obtain our datasets from the MOOC provider Coursera. Since other MOOC platforms use the same forum structure, our methods are generally 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.

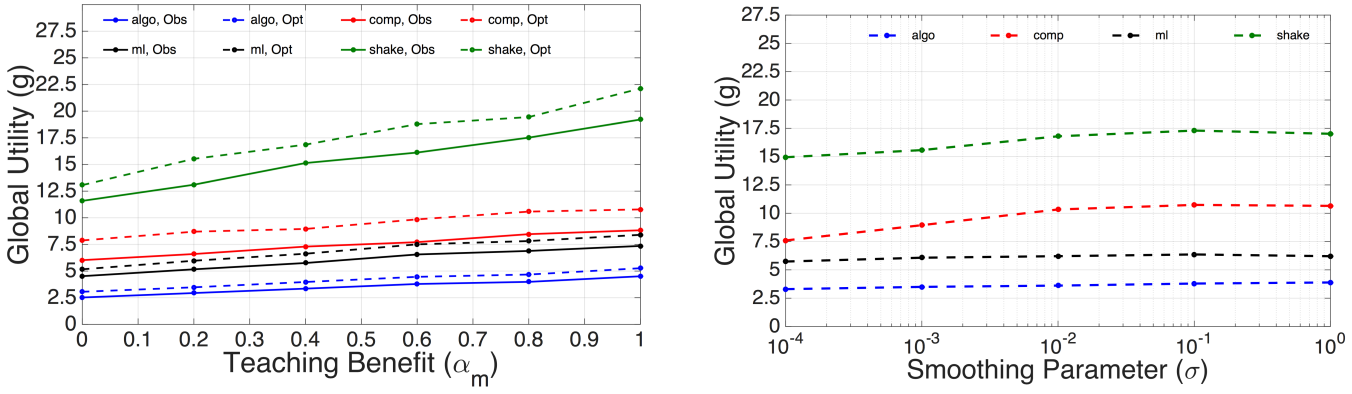
2) Courses: We chose four MOOCs for analysis: "Machine Learning" (`ml`), "English Composition I" (`comp`), "Algorithms: Design and Analysis, Part 1" (`algo`), and "Shakespeare in Community" (`shake`). We picked two courses that are technical in nature (`ml` and `algo`) and two on the humanities side (`comp` and `shake`), to obtain a diverse



(a) Varying α_m.

(b) Varying σ.

Fig. 3: Efficiency measures as (a) the teaching benefit α_m and (b) the smoothing factor σ are varied. In (a), we see that the η^g efficiencies are always below 0.9, highlighting the potential gains through optimization. This holds despite there being some variations in efficiency depending on what is taken as the true value of α_m. In (b), we see that as σ increases, the networks tend to become more efficient, indicating that improvement can be obtained in global utility if users are more impartial in responding.



(a) Varying α_m.

(b) Varying σ.

Fig. 4: Global utilities as (a) the teaching benefit and (b) the smoothing factor vary. The ratios between the observed $\hat{g}$ and optimized $g^\star$ global utilities in (a) correspond to the plots of η^g in Fig. 3(a). α_m and σ both have a more profound impact on global utility in the humanities courses, which indicates that learners benefit from sharing information across broader networks of peers in this type of course.

sample with respect to subject matter. The sessions of these MOOCs that we used were all publicly-accessible and had passed the final exam date listed on the syllabus in June 2015. Table I gives basic information on them including the number of users, threads, and posts; the large total numbers of posts but relatively small average number of posts per user – ranging from 4.4 to 7.8 – are typical of MOOCs [4]. As discussed in Sec. I, small participations of individual users is one of the challenges to optimizing SLN efficiency in MOOCs.

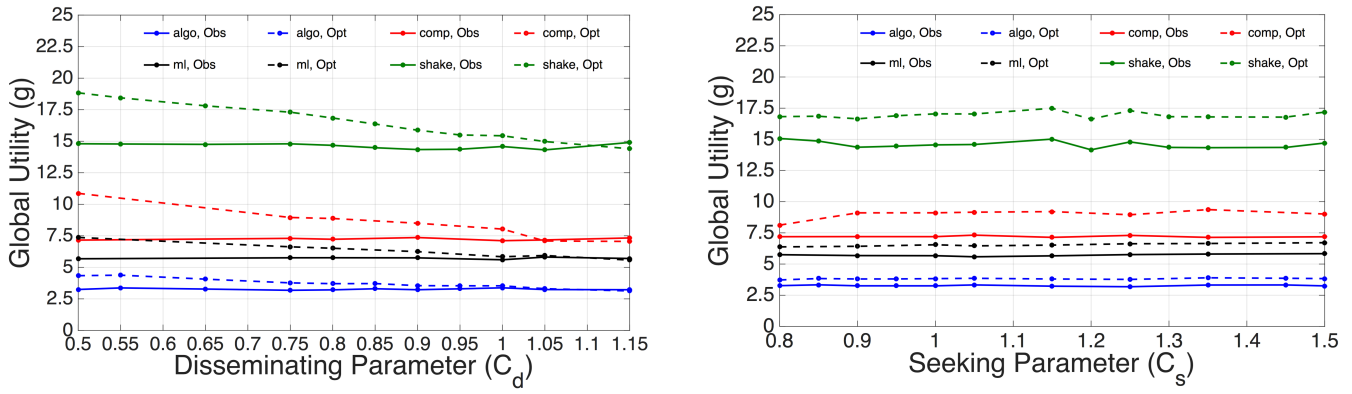
B. Extracting Topics and Q&A Tendencies

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 $\mathcal{K}$: We implemented LDA using collapsed Gibbs sampling, through the `lda` library in Python. We empirically varied the number of topics for each dataset, and inspected (i) the highest constituent words $\arg\max_x \beta_{k,x}$ 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 reasonable disparity among the top words (*i.e.,* ensuring each topic is different).

Table II gives a summary of the results for each dataset, with the three words having highest f_k shown for each k. From the top three words, we see that the topics (i) are representative of likely discussions for each course (*e.g.,* $k = 2, 3, 7$ in `shake` are about specific Shakespeare plays, and $k = 3, 10$ in `algo` are about data types and graphs, respectively), and (ii) are reasonably non-overlapping, with the exception of ubiquitous course words (*e.g.,* "write" in `comp`, "number" in `algo`).

2) Seeking S and disseminating D tendencies: With the topics $\mathcal{K}$ identified and the $q_{u,r,k}$ computed as in Sec. III-B2, we can infer the $d_{u,k}$ and $s_{u,k}$ from (10)-(11). As a sample, in Fig. 2, we plot the distributions across users for the two topics in each course that have highest f_k (see Table II), considering the non-zero values only. For the 40 topics across the courses (8 shown), we make a few observations. For one, we notice that the $d_{u,k}$ values tend to be shifted to the right relative to the $s_{u,k}$; in particular, the median is higher in $29/40$ cases, and in $5/8$ of the cases in Fig. 2. This indicates that there is higher disseminating tendency overall, consistent with the observation in Sec. III-B2 that there are more answer posts



(a) Varying C_d.

(b) Varying C_s.

Fig. 5: Global utilities as (a) the disseminating parameter and (b) the seeking parameter vary. By dropping C_d, users are able to participate more, and we see in (a) that the optimization is able to leverage that additional dissemination to obtain more improvement. By increasing C_s, users may get more responses to their questions than before, but we see in (b) that this does not affect the global utility. Having a few, high quality sources is sufficient.

than question posts. However, $s_{u,k}$ and $d_{u,k}$ do tend to be on the same order; the ratio of the medians is less than 2 in $31/40$ cases. 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.

Before proceeding, we remark that the inferred topics (*e.g.,* in Table II) and distributions of seeking/disseminating tendencies (*e.g.,* in Fig. 2) could serve as useful analytics for a course instructor in their own right. It would allow the instructor to see which topics in his/her course have the highest disparity between disseminating and seeking tendencies, and which discussion words tend to make up these topics. With this, the instructor could devise interventions for the course that would benefit the students.

C. Efficiency Evaluation

We now move to the optimization evaluation and results.

Parameters. Referring to (6) and (7), there are four parameters: α_u, C_s, C_d, and σ. α_u, the marginal benefit of teaching relative to learning for user u, depends on several factors and is likely user-dependent. As a result, 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. We set $\alpha_m = 0.4$, the smoothing factor $\sigma = 0$, and the tightness parameters $C_s = 1.25$ and $C_d = 0.75$ by default; each of these values will be swept across suitable values in the evaluation to analyze their effects.

Implementation. In Algorithm 1, each step was coded de-novo in Python. The simulations were run across six machines, each with 12 cores and 32 GB RAM. Due to the random nature of α_u, each choice of parameters was averaged over multiple simulation runs. We fix $\lambda = 0.1$ and $T = 0.01$.

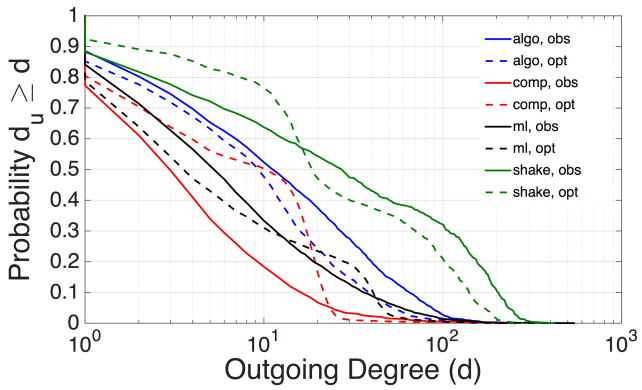
Results. Fig. 3(a) shows two efficiency measures, η^g and η^F, as α_m is varied. η^g is the actual efficiency based on global utility g from (7), while η^F is the ratio based on the full objective function (12), given for completeness. Fig. 4(a) plots the corresponding observed and optimized values of g. Fig. 3(b) shows how the efficiency η^g of the observed network $\hat{W}(\sigma)$ varies with σ, and Fig. 4(b) gives the corresponding global utilities. Finally, Figs. 5(a) and (b) show how varying C_d and C_s affect the global utility in the optimized network.[9] These graphs are the subject of the following discussion.

1) Low efficiency SLNs: Referring to Fig. 3(a) and Fig. 4(a), for each dataset we can see that *the observed SLNs have low efficiencies*, *i.e.,* they obtain substantially less global utility than the optimal. This is true regardless of what is taken as the true teaching benefit α_m for each course: the highest of $\eta^g = 0.90$ is obtained by `shake` with $\alpha_m = 0.8$, while the lowest of $\eta^g = 0.76$ is obtained by `comp` with $\alpha_m = 0.2$. From these results, we see that much can be gained through optimization; in Sec. IV-D3 we will see that local utilities are not substantially penalized in the process either. Also, η^F is consistently higher than η^g in Fig. 3(a), which is consistent with the regularization parameter λ in (12) being 0 at $W = \hat{W}$; deviations from W are penalized in the objective, thus giving insight into how far $W^\star$ is from $\hat{W}$.

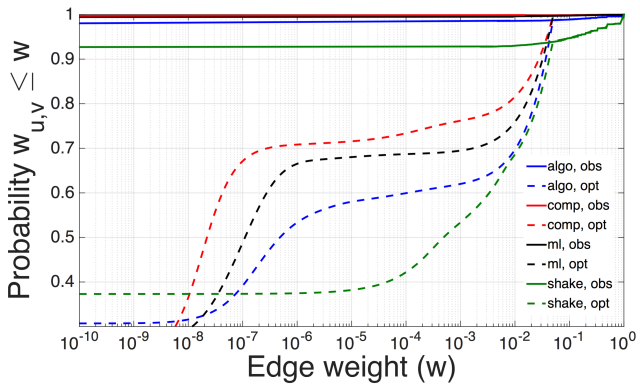
2) Higher efficiency SLNs for more smoothing: Referring to Fig. 3(b), we see that as the smoothing parameter σ increases, the SLNs in each course gain in efficiency. Once $\sigma = 1$, η^g has reached between 0.91 (for `algo`) and 0.94 (for `shake`). Given that larger σ in (8) has the effect of spreading the observed, overall response rate of user u more uniformly across other users v (i.e., equalizing the $\hat{w}_{u,v}$ 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 an 8% gap between the smoothened SLNs and the optimal solution, indicating there is still substantial room for improvement through optimization.

3) Utility gains in humanities courses: As α_m is increased in Fig. 4(a) to factor in teaching benefit, the global utilities – both $\hat{g}$ and $g^\star$ – increase as well, as expected from (1). This increase is more pronounced for `shake` and `comp` than for `algo` and `ml`, though, especially for `shake` where $g^\star$ rises from 13 to 22. This implies that the "learning by teaching" factor in (1) tends to be larger for the humanities than for the

[9]The minor variations in global utility for the observed network here are due to the random samplings for α_u.

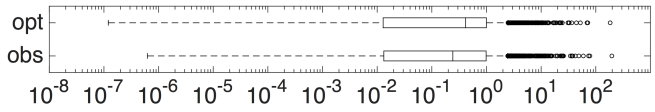


(a) Degree distributions, $P(d_u \geq d)$.

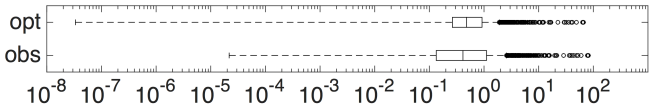


(b) Edge weight distributions, $P(w_{u,v} \leq w)$.

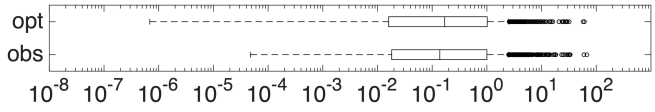
Fig. 6: Plots of (a) the outgoing degree distributions and (b) the edge weight distributions for observed and optimal networks. For each dataset, we can see that optimization makes the distributions more uniform, with (a) less users having large outgoing degrees and (b) many additional connections established between pairs of users.



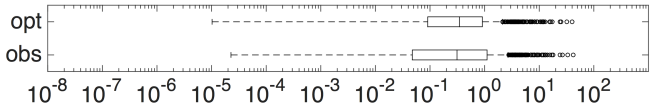
(a) `ml` obs: 0.24 (0.056); opt: 0.41 (0.061)



(b) `comp` obs: 0.41 (0.074); opt: 0.48 (0.089)



(c) `algo` obs: 0.14 (0.090); opt: 0.17 (0.097)



(d) `shake` obs: 0.31 (0.136); opt: 0.35 (0.132)

Fig. 7: 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. 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 of local utility.

technical courses, *i.e.,* there is a higher match between users' seeking tendencies $s_{u,k}$ and the disseminating tendencies $d_{v,k}$ of those responding. A similar trend is seen for the smoothing factor in Fig. 4(b), where $\hat{g}$ noticeably increases with σ in the humanities but not the technical courses. Together, these findings imply that *learners benefit from sharing information across broader networks of peers in the humanities courses*. This may be explained by the discussion-oriented nature of this course type, as opposed to technical courses where learners would tend to ask targeted questions with objective answers.

4) SLNs can leverage additional dissemination: In Fig. 5(a), we see that as the disseminating parameter C_d drops, the gap in global utility between the observed and optimized networks gets considerably larger, *i.e.,* there is more room for improvement. Recall that lower C_d in (6) allows users' DISRs to drop, which simulates the case that they can take on a larger load of questions or that their individual capacities were underestimated. This implies that *the optimized networks can take advantage of additional disseminating capacity, especially in the humanities courses*. In Fig. 5(b), on the other hand, we see that the seeking parameter C_s does not affect the gap over this range of values. Recalling that a higher C_s allows users' SIDRs to drop if needed, *i.e.,* users' may receive less answers to questions, this implies that *having more dissemination to match the same seeking tendency does not tend to benefit global utility*. Having the strongest few answers is sufficient.

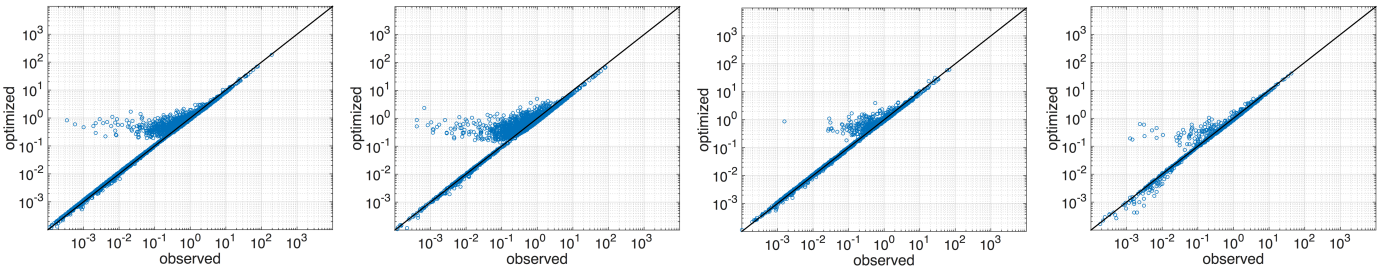
D. Network Comparison

Equipped with an understanding of efficiency in our datasets, we now perform an exploratory analysis to discover differences between the observed and optimal SLN. All parameters are set to the defaults stated at the beginning of Sec. IV-C.[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 user 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 $d_u = \sum_v a_{u,v}$; in other words, d_u is the number of unique users that u is expected to respond to.

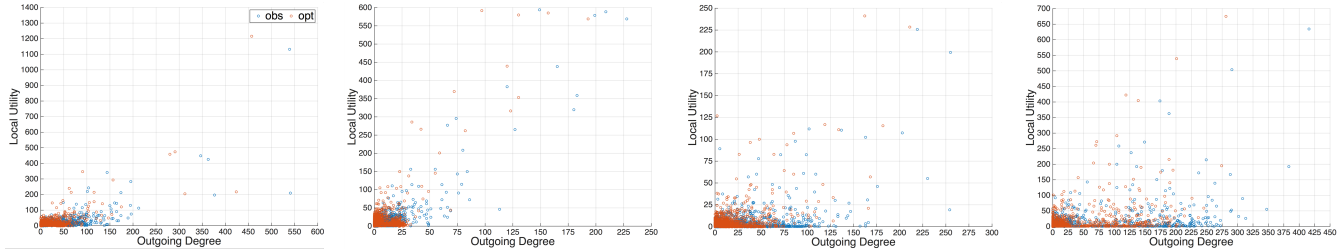
Fig. 6(a) plots the degree distributions $P(d_u \geq d)$ across users for each network. Visually, we can see that *optimization tends to make the degrees more uniform*, reducing the number of users on the tail of the distribution. For example, in `algo`, the proportion of users with $d_u \geq 30$ is reduced from 28% to 15%, and in `ml` the proportion with $d_u \geq 50$ is reduced from 6.4% to 2.9%. After optimization, there are more users with $d_u \leq 20$ for `comp`, $d_u \leq 15$ for `shake`, and $d_u = 1$ for both `ml` and `algo` than there were before.

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



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

Fig. 8: 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 the same or higher local utilities in each case.



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

Fig. 9: Plot of local utility l_u against (expected) outgoing degree d_u for each of the datasets, before and after optimization. In each case except comp, we can see that learners are on average able to obtain comparable l_u for a lower d_u.

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. 6(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*. Considering $\hat{W}$, there are roughly 73K and 66K non-zero weights for ml and shake, which is only 0.40% and 7.2% of the potential user pairs in the network. Considering $W^\star$ on the other hand, 5.54M (31%) and 427K (47%) of the pairs are non-zero with $w^\star_{u,v} \geq 0.001$.

The distributions in Fig. 6 are also consistent with the finding in Fig. 3(b) that smoothing generally improves efficiency.

3) Optimization preserves fairness: Setting the tightness parameters $C_s = 1.25$ and $C_d = 0.75$ in (6) leaves the potential of sacrificing individual local utilities l_u at the expense of maximizing global utility g. 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; the ratio is taken to account for the increase in global utility from optimization.

Fig. 7 gives boxplots of these values for each dataset. 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 the fairness of the r_u distributions through the standard Jain's Index (JI) metric.[11] The JI values are given in Fig. 7; we see that they do not change substantially after optimization for any of the datasets (and actually increase by around 0.01 in comp, ml and algo). Therefore, we conclude that while improving the global utility, *optimization also preserves fairness in the distribution of local utilities*. This also verifies that our choice of constraint (6b) to preserve incoming information rather than bound local utility did not result in a negative impact on individual users after optimization.

4) Increases in local utilities: We are also interested in the differences between the local utilities l_u before (*i.e.,* $\hat{l}_u$) and after (*i.e.,* $l^\star_u$) optimization, irrespective of g. In Fig. 8, we plot the effect of optimization on the local utilities, where each point is a user. Visually, it is apparent that *optimization preserves or improves local utility for the majority of users*. The percentage of users with $l^\star_u \geq \hat{l}_u$ (*i.e.,* at or above the black line) is 74% for ml and 60% for comp. Only in shake is it under 50%, but 52% of cases increase or drop by at most 4%.

In Fig. 6(a), we saw that optimization tends to make the expected outgoing degree d_u more uniform. In Fig. 9, we plot the local utility l_u against d_u for each of the datasets, comparing the observed and optimized SLNs in each case. In ml, algo, and shake, we see visually that users are on average able to obtain the same l_u in the optimized network with a smaller d_u. The average user in comp, however, obtains $l_u = 9.0$ with $d_u = 7.5$ after optimization, as opposed to a lower $l_u = 7.3$ from a lower $d_u = 4.6$ before optimization.

V. Discussion and Forum Implementation Algorithm

From the evaluation in Sec. IV, it is apparent that large increases in global utility can be obtained by optimizing user participation (Fig. 3). Importantly, this can be done without affecting the spread of local utilities substantially, meaning that fairness is preserved, and even improved (Fig. 7). To

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

obtain these gains, the optimized network will take a more homogeneous structure, with both the outgoing degree and edge weight distributions becoming more uniform (Fig. 6). The effect of this is a more connected community of users with a more distributed workload, causing the local utilities of the majority of users to increase (Fig. 8), and giving learners the ability to obtain the same or higher local utility with lower outgoing degree (Fig. 9).

These results imply that substantial improvements in learning efficacy can be gained in MOOCs – and online learning/education more generally – through SLN optimization. This provides a means for combating the high attrition rates and other adverse outcomes associated with scaling up learning [4], without incurring costs associated with adding additional instructors to each course. Certain analytics provided by our methodology, such as the course topics (Table II) and seeking/disseminating tendencies (Fig. 2), can also be used by the instructors to themselves devise interventions for students.

As outlined in Sec. I, there are three steps involved in improving SLN efficiency: (i) defining the ideal network, (ii) solving for the optimal SLN, and (iii) realizing the optimized network in practice. In presenting and evaluating our efficiency methodology, the focus of this work has been on the first two steps. The third step, which we leave mostly to future work, can be broken down into the following main parts: (a) designing an algorithm to recommend/enforce the optimized interaction structure in an SLN, (b) adding the appropriate UI/UX functionality to the web application hosting the SLN, and (c) obtaining a large volume of users for experimentation from an existing MOOC provider to measure the improvements in learning outcomes from the optimization. While (b) and (c) are out of scope here, we will next propose a solution for (a) based on our optimization methodology; our choosing of the solution closest to the observed network in step (ii) also makes this less disruptive for users overall.

Since most online forums already provide a news feed to direct user attention in an SLN to new or popular posts, we propose 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. 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. Also, given that the observed SLN evolves over time, the $W^\star$ can be re-computed at appropriate points (*e.g.,* once a day).

A key challenge here is encouraging/ensuring users follow the recommendations. It may be possible to design an incentive structure (*e.g.,* through awarding badges as in [3]) 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 [19]), it may be possible to force users to follow the recommendations by enforcing consequences for those who do not.

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 post $p \in \mathcal{C}_u$ with $t_c - t(p) > T$, *i.e.,* outdated posts in $\mathcal{C}_u$}
  if $U(0,1) \geq w^*_{u,v}$ **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 $w^*_u(\mu(p))$, *i.e.,* from highest to lowest $w^*_{u,v}$}
    if $|\mathcal{C}_u| > c_{max}(u)$ **then**
      RM-ST($\mathcal{C}_u$) {RM-ST: remove the last (*i.e.,* least relevant) element from $\mathcal{C}_u$}
Return: $\mathcal{C}_u\ \forall u$ {Updated news feed for each user u}

VI. Related Work

Several studies on MOOCs have emerged in recent years. Many of these have aimed to codify the learning process through data-driven methodologies. Researchers have proposed algorithms for clickstream data analysis [20], [21], performance prediction [22], [23], community detection [24], study partner recommendation [5], [25], and forum question recommendation [9]; see also [2] for a survey of earlier works.

In this paper, we focus specifically on the discussion forum aspect of MOOCs. Some prior work has analyzed the content of discussions [4], [7] while others have considered the graph structure [8], [26] of the forums to gain insight into user behavior. More specifically, [7] proposed an extension of non-negative matrix factorization to characterize students by learnt latent features using the text of forum posts. [8] used social network analysis to identify significant interaction networks among students, detect communication vulnerability, and simulate the effect of information diffusion on an undirected user-user graph. [26] provided a socio-semantic analysis on users' roles in an SLN according to their information-giving relations, using bipartite graph modeling to identify user similarities. Different from these works, our methodology takes a unified view of the topic-specific content and structural aspects of MOOC forums, and models the flow of information between users as a directed graphical process.

Our work is also unique in that we propose methodology to optimize student interactions. Recent empirical analysis [27] has highlighted the potential for improving MOOC learning efficacy from a network perspective. Also, the analysis in [28] observed the uneven distribution of interactions between core participants and other users in an SLN; our work indicates that these uneven distributions are indeed suboptimal. The methodology we propose is perhaps most related to that in [9], in which the authors propose a method for optimizing the allocation of users to questions, but ignore the specific content of each question and make the implicit assumption that a user's participation implies expertise. The methodology we develop, by contrast, infers both question and answer tendencies of each user over a multidimensional topic space. Further, we choose to discover topic distributions through natural language

processing, in light of works [4], [29] showing that human identification of topics may fail to capture important discussions.

A plethora of studies exist for Online Social Networks (OSN) more generally. Information propagation and efficiency in OSNs has been studied in *e.g.,* [30], [31] for public social networking/blogging, [19], [32] for enterprise social networks, [10], [33], [34] for recommender networks, and [35], [36] for human learning networks. As in [10], our work considers optimization of local and global utilities, but considers constraints specific to SLN such as multidimensional information spread. The scale (up to 18M variables) and non-uniqueness of our optimization also pose unique computational challenges overcome in this work, through projected gradient descent/ADMM and Frobenius norm regularization, respectively.

VII. Conclusion

The proliferation of online (human) learning in recent years has rendered 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 methodology which compares the observed user benefit to that which can be obtained in an optimized, ideal SLN. Through our method, each user is modeled as possessing a certain level of seeking (*i.e.,* question asking) and disseminating (*i.e.,* question answering) tendency on a set of latent topics forming the educational context of the SLN. We evaluated efficiency on the discussion forums from four MOOC courses, in which we compared the observed and optimal SLN along a number of dimensions. For one, we saw that the efficiency of the SLN is rather low, with much to be gained through optimization. Also, in addition to improving global utility, the optimal network surprisingly does not penalize the fairness in the distribution of local utilities. The main step for future work beyond the modeling and optimization presented here is the implementation of a mechanism to enforce the optimized network in a discussion forum during a course.

Acknowledgment

This work was supported by Zoomi Inc and ARO grants W911NF-16-1-0448, W911NF-14-1-0190, and W911NF-11-1-0036. The authors thank the reviewers for their comments.

References

[1] C. G. Brinton, S. Buccapatnam, F. Wong, M. Chiang, and H. V. Poor, "Social Learning Networks: Efficiency Optimization for MOOC Forums," in *Proc. of IEEE INFOCOM*, 2016.

[2] C. G. Brinton and M. Chiang, "Social Learning Networks: A Brief Survey," in *Proc. of IEEE CISS*, 2014.

[3] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Engaging with Massive Online Courses," in *ACM WWW*, 2014.

[4] 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.

[5] B. Xu and D. Yang, "Study Partners Recommendation for xMOOCs Learners," *Computational Intelligence & Neuroscience*, vol. 2015, 2015.

[6] C. G. Cortese, "Learning Through Teaching," *Management Learning*, vol. 36, no. 1, pp. 87–115, 2005.

[7] N. Gillani, R. Eynon, M. Osborne, I. Hjorth, and S. Roberts, "Communication Communities in MOOCs," *arXiv:1403.4640*, 2014.

[8] 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.

[9] D. Yang, D. Adamson, and C. P. Rosé, "Question Recommendation with Constraints for Massive Open Online Courses," in *RecSys*, 2014, pp. 49–56.

[10] F. M. F. Wong, Z. Liu, and M. Chiang, "On the efficiency of social recommender networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2512–2524, 2016.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006.

[12] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero, "Shrinkage Algorithms for MMSE Covariance Estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5016–5029, 2010.

[13] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth, "Unsupervised Modeling for Understanding MOOC Discussion Forums: A Learning Analytics Approach," in *LAK*, 2015, pp. 146–150.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[15] 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.

[16] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun, "Finding Question-Answer Pairs from Online Forums," in *SIGIR*, 2008.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[19] J. Cao, H. Gao, L. E. Li, and B. Friedman, "Enterprise Social Network Analysis and Modeling: A Tale of Two Graphs," in *Proc. of IEEE INFOCOM*, 2013.

[20] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg, "Your Click Decides your Fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions," in *EMNLP*, 2014.

[21] C. G. Brinton, S. Buccapatnam, M. Chiang, and H. V. Poor, "Mining MOOC Clickstreams: Video-Watching Behavior vs. In-Video Quiz Performance," *IEEE Trans. Signal Proc.*, vol. 64, no. 14, pp. 3677–3692, 2016.

[22] C. G. Brinton and M. Chiang, "MOOC Performance Prediction via Clickstream Data and Social Learning Networks," in *Proc. of IEEE INFOCOM*, 2015.

[23] T.-Y. Yang, C. G. Brinton, C. Joe-Wong, and M. Chiang, "Behavior-Based Grade Prediction for MOOCs via Time Series Neural Networks," *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[24] A. Ochirbat, M. Namsraidorj, and W. Y. Hwang, "Small K-Teams Recommendation in Social Learning Networks," in *2014 7th International Conference on Ubi-Media Computing and Workshops*, 2014.

[25] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, "E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style Identification," *Computers & Education*, vol. 56, no. 3, pp. 885–899, 2011.

[26] T. Hecking, I.-A. Chounta, and H. U. Hoppe, "Investigating Social and Semantic User Roles in MOOC Discussion Forums," in *Proc. of ACM LAK*, 2016.

[27] J. Zhang, M. Skryabin, and X. Song, "Understanding the Dynamics of MOOC Discussion Forums with Simulation Investigation for Empirical Network Analysis (SIENA)," *Distance Education*, vol. 37, no. 3, pp. 270–286, 2016.

[28] Y. Hu and F. Zhao, "A Social Network Analysis of Online Collaborative Learning Aspects in an Online Course," in *2016 International Symposium on Educational Technology*, 2016.

[29] J. Cruz-Benito, O. Borrás-Gené, F. J. García-Peñalvo, Á. F. Blanco, and R. Therón, "Learning Communities in Social Networks and Their Relationship With the MOOCs," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 12, no. 1, pp. 24–36, 2017.

[30] M. Cha, A. Mislove, and K. P. Gummadi, "A Measurement-Driven Analysis of Information Propagation in the Flickr Social Network," in *ACM WWW*, 2009.

[31] D. Wang, H. Park, G. Xie, S. Moon, M.-A. Kaafar *et al.*, "A Genealogy of Information Spreading on Microblogs: A Galton-Watson-Based Explicative Model," in *Proc. of IEEE INFOCOM*, 2013.

[32] I. Guy, I. Ronen, N. Zwerdling, I. Zuyev-Grabovitch, and M. Jacovi, "What is Your Organization'Like'?: A Study of Liking Activity in the Enterprise," in *Proc. of ACM CHI*, 2016.

[33] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent Recommender Networks," in *Proc. of ACM WSDM*, 2017.

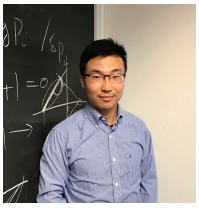
[34] X. Liu and K. Aberer, "SoCo: A Social Network Aided Context-Aware Recommender System," in *Proceedings of ACM WWW*, 2013.
[35] C. Haythornthwaite and M. De Laat, "Social Networks and Learning Networks: Using Social Network Perspectives to Understand Social Learning," in *Proc. of the 7th international conference on networked learning*, 2010.
[36] N. M. Dowell, O. Skrypnyk, S. Joksimovic, A. C. Graesser, S. Dawson, D. GaĹevic, T. A. Hennis, P. de Vries, and V. Kovanovic, "Modeling Learners' Social Centrality and Performance through Language and Discourse," *International Educational Data Mining Society*, 2015.

Christopher G. Brinton (S'08, M'16) is the Head of Advanced Research at Zoomi Inc, a learning technology company he co-founded in 2013, and a Lecturer in Electrical Engineering at Princeton University. His research focus is data science for social networks, with a particular emphasis on predictive learning analytics, social learning networks, and personalized learning for education. Chris co-authored the book *The Power of Networks: Six Principles that Connect our Lives*, and has reached over 250,000 students through MOOCs based on his book. A recipient of the 2016 Bede Liu Best Dissertation Award in Electrical Engineering, Chris received his PhD from Princeton in 2016, his Masters from Princeton in 2013, and his BSEE from The College of New Jersey (valedictorian and summa cum laude) in 2011, all in Electrical Engineering.

Swapna Buccapatnam is a Principal Inventive Scientist at AT&T Labs, Inc. Prior to this, she was a postdoctoral researcher at the IBM T.J. Watson Research Center, Yorktown Heights, NY, USA and a postdoctoral research associate in the Department of Electrical Engineering at Princeton University. She received her Ph.D. in Electrical and Computer Engineering from the Ohio State University in 2014 and her undergraduate degree in Electrical Engineering from the Indian Institute of Technology, Madras, India, in 2008. Her research interests lie in stochastic modeling and analysis, machine learning, data analytics, and optimization.

Liang Zheng was a postdoctoral research associate at Department of Electrical Engineering of Princeton University. She received the bachelor's degree in software engineering from Sichuan University, Chengdu, China, in 2011, and the Ph.D. in computer science from the City University of Hong Kong, Hong Kong, in 2015. Previously, she was a Visiting Student Research Collaborator at Princeton University, Princeton, NJ, USA, in 2014. Her research interests are primarily in understanding user behavior in computing systems, particularly from an economic perspective.

Da Cao is a Research Algorithm Engineer at Zoomi Inc, an innovative learning technology company. Da's work involves software development, data processing, feature extraction and exploring machine learning algorithms for behavioral analysis. He received his Bachelor of Science in Electrical Engineering from University of Wisconsin - Madison and his M.S. in Electrical Engineering from University of Pennsylvania.



Andrew S. Lan is a postdoctoral research associate in the EDGE Lab at the Department of Electrical Engineering, Princeton University since Feb. 2017. From June 2016 to Jan. 2017, he was a postdoctoral research associate in the Digital Signal Processing (DSP) group and OpenStax at Rice University. He received his M.S. and Ph.D. degrees in May 2014 and May 2016, respectively, in the Rice DSP group. His research focuses on the development of human-in-the-loop machine learning methods to enable scalable, effective, and fail-safe personalized learning in education, by collecting and analyzing massive and multi-modal learner and content data.

Felix Wong (S'10, M'15) received the B.Eng. in computer engineering from the Chinese University of Hong Kong, Hong Kong, in 2007, the M.Sc. in computer science from the University of Toronto, Toronto, ON, Canada, in 2009, and the Ph.D. in electrical engineering from Princeton University, Princeton, NJ, USA, in 2015. He is currently a Software Engineer at Google, Inc., Mountain View, CA, USA.

Sangtae Ha (S'07, M'09, SM'12) is an Assistant Professor in the Department of Computer Science at the University of Colorado Boulder. He received his Ph.D. in Computer Science from North Carolina State University. He co-founded the Princeton EDGE Lab as its first Associate Director in 2009 and led its research team as an Associate Research Scholar at Princeton University from 2010 to 2013. His research focuses on building and deploying practical network systems. He is a Co-Founder and the founding CTO/VP Engineering of DataMi, a startup company on mobile networks. He also co-founded Zoomi, an artificial intelligence-based learning analytics company. He received the INFORMS ISS Design Science Award in 2014. He serves as an Associate Editor for the IEEE Internet of Things Journal.

Mung Chiang (S'00, M'03, SM'08, F'12) is the John A. Edwardson Dean of the College of Engineering at Purdue University. Previously he was the Arthur LeGrand Doty Professor of Electrical Engineering at Princeton University. His research on networking received the 2013 Alan T. Waterman Award, the highest honor to US young scientists and engineers. His textbook *Networked Life*, popular science book *The Power of Networks*, and online courses reached over 250,000 students since 2012. He founded the Princeton EDGE Lab in 2009, which bridges the theory-practice gap in edge networking research by spanning from proofs to prototypes. He also co-founded a few startup companies in mobile data, IoT and AI, and co-founded the global nonprofit Open Fog Consortium.

H. Vincent Poor (S'72, M'77, SM'82, F'87) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of Electrical Engineering. During 2006 to 2016, he served as Dean of Princeton?s School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory and signal processing, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the recent book *Information Theoretic Security and Privacy of Information Systems* (Cambridge University Press, 2017). Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the Marconi and Armstrong Awards of the IEEE Communications Society in 2007 and 2009, respectively. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal, Honorary Professorships at Peking University and Tsinghua University, both conferred in 2017, and a D.Sc. honoris causa from Syracuse University also awarded in 2017.

APPENDIX A
PROOF OF NON-STRICT-CONCAVITY OF (6A) AND NON-UNIQUENESS OF SOLUTIONS TO (6)

To prove that (6a) is not strictly concave, we seek to find conditions under which, for two matrices $X = [x_{u,v}]$ and $Y = [y_{u,v}]$ in the feasible region of (6), the concavity inequality

$$g(\mu X + (1-\mu)Y) \geq \mu f(X) + (1-\mu) f(Y), \;\; \mu \in [0,1]$$

holds with equality. We have

$$\begin{aligned}
&g\left(\mu X + (1-\mu)Y\right) \\
&= \sum_{u,k} s_{u,k} \log\left(1 + \sum_v \left(\mu x_{v,u} + (1-\mu) y_{v,u}\right) d_{v,k}\right) \\
&\quad + \sum_{u,k} \alpha_u d_{u,k} \log\left(1 + \sum_v \left(\mu x_{u,v} + (1-\mu) y_{u,v}\right) s_{v,k}\right) \\
&= \log \prod_{u,k} \left(\mu(1 + \sum_v x_{v,u} d_{v,k}) + (1-\mu)(1 + \sum_v y_{v,u} d_{v,k})\right)^{s_{u,k}} \\
&\quad \times \left(\mu(1 + \sum_v x_{u,v} s_{v,k}) + (1-\mu)(1 + \sum_v y_{u,v} s_{v,k})\right)^{\alpha_u d_{u,k}},
\end{aligned}$$

and

$$\begin{aligned}
&\mu g(X) + (1-\mu) g(Y) \\
&= \mu \sum_{u,k} \left(s_{u,k} \log(1 + \sum_v x_{v,u} d_{v,k}) + \alpha_u d_{u,k} \log(1 + \sum_v x_{u,v} s_{v,k})\right) \\
&\quad + (1-\mu) \sum_{u,k} \left(s_{u,k} \log(1 + \sum_v y_{v,u} d_{v,k})\right. \\
&\quad \left. + \alpha_u d_{u,k} \log(1 + \sum_v y_{u,v} s_{v,k})\right) \\
&= \log \prod_{u,k} \left((1 + \sum_v x_{v,u} d_{v,k})^{\mu} \cdot (1 + \sum_v y_{v,u} d_{v,k})^{1-\mu}\right)^{s_{u,k}} \\
&\quad \times \left((1 + \sum_v x_{u,v} s_{v,k})^{\mu} \cdot (1 + \sum_{v,k} y_{u,v} s_{v,k})^{1-\mu}\right)^{\alpha_u d_{u,k}}.
\end{aligned}$$

By the weighted AM-GM inequality, the inequalities

$$\begin{aligned}
&\mu(1 + \sum_v x_{v,u} d_{v,k}) + (1-\mu)(1 + \sum_v y_{v,u} d_{v,k}) \\
&\geq (1 + \sum_v x_{v,u} d_{v,k})^{\mu} \cdot (1 + \sum_v y_{v,u} d_{v,k})^{1-\mu},
\end{aligned}$$

and

$$\begin{aligned}
&\mu(1 + \sum_v x_{u,v} s_{v,k}) + (1-\mu)(1 + \sum_v y_{u,v} s_{v,k}) \\
&\geq (1 + \sum_v x_{u,v} s_{v,k})^{\mu} \cdot (1 + \sum_v y_{u,v} s_{v,k})^{1-\mu}
\end{aligned}$$

hold with equality if (and only if) $\sum_v x_{v,u} d_{v,k} = \sum_v y_{v,u} d_{v,k}$ and $\sum_v x_{u,v} s_{v,k} = \sum_v y_{u,v} s_{v,k} \;\forall u, k$, respectively. Thus, g is not strictly concave.

We then prove that if $X^\star$ is an optimal solution, there exists another optimal solution $Y^\star$. We suppose two vectors a and c such that $a^T c = 0$ (i.e., $\mathrm{diag}(ac^T) = \mathbf{0}$), $D^T a = \mathbf{0}$, and $S^T c = \mathbf{0}$, and we show that there exist such a and c. By observation, a and c are left nullspaces of D and S with dimensions $N - \mathrm{rank}(D)$ and $N - \mathrm{rank}(S)$ respectively, i.e., both are $N - K > 2$. Thus, there are more than one vectors in a basis of each of their nullspaces. Supposing $\mathcal{N}(D^T) \in \mathbb{R}^{N \times (N-K)}$ and $\mathcal{N}(S^T) \in \mathbb{R}^{N \times (N-K)}$ are bases of left nullspaces of D and S, respectively, we define two vectors

$$a' = \sum_{i=1}^{N-K} \lambda_i \mathcal{N}_i(D^T) = \mathcal{N}(D^T)^T \lambda$$

and

$$c' = \sum_{i=1}^{N-K} \nu_i \mathcal{N}_i(S^T) = \mathcal{N}(S^T)^T \nu$$

with λ, $\nu \in \mathbb{R}^{N-K}$. We then have

$$a'^T c' = \lambda^T \mathcal{N}(D^T) \mathcal{N}(S^T)^T \nu.$$

Here, note that $\mathrm{rank}(\mathcal{N}(D^T)\mathcal{N}(S^T)^T) = N - K$. The $\mathcal{N}(D^T)\mathcal{N}(S^T)^T$ has the singular decomposition with

$$\mathcal{N}(D^T)\mathcal{N}(S^T)^T = U \Sigma V^T,$$

where Σ is a diagonal matrix and U and V are both unitary matrix, i.e., $U^T U = V^T V = I$. We also define $\lambda = U\lambda'$ and $\mu = V\mu'$, satisfying $\lambda'_i = 0$ if $\nu'_i \neq 0$, $\lambda'_i \neq 0$ if $\nu'_i = 0$, $\|\lambda\|_0 \geq 1$ and $\|\nu\|_0 \geq 1$ so that $a'^T c' = \lambda'^T \mu' = 0$. Now, to ensure $Y^\star = X^\star + ac^T \in [0,\,1]^{|\mathcal{U}| \times |\mathcal{U}|}$, ac^T is given by

$$ac^T = \frac{1}{\max\limits_{a'_i c'_i \neq 0, X_{ij} \neq 0,1} \frac{a'_i c'_i}{\min\{1 - X_{ij},\, |X_{ij}|\}}} a' c'^T$$

Since any $Y^\star = X^\star + ac^T \in [0,\,1]^{|\mathcal{U}| \times |\mathcal{U}|}$ would satisfy $\mathrm{diag}(Y^\star) = \mathbf{0}$, $Y^{\star T} d = X^{\star T} d \geq s$, and $Y^\star s = X^\star s \leq d$, $Y^\star$ is another optimal solution.

APPENDIX B
ALGORITHM FOR THE PROJECTION STEP

We now detail our proximal gradient-within-ADMM algorithm to perform the projection step, *i.e.,* to solve the proximal problem defined in (14). The problem can be written equivalently as

$$\begin{aligned}
\underset{W}{\text{minimize}} \quad & \frac{1}{2}\|W - \hat{W}\|_F^2, \\
\text{subject to} \quad & -D^T W + P \leq \mathbf{0}, \\
& WS - Q \leq \mathbf{0}, \\
& \mathbf{0} \leq W \leq \mathbf{1},\; \mathrm{diag}(W) = \mathbf{0},
\end{aligned}$$

where the $|\mathcal{K}| \times |\mathcal{U}|$ matrix P is given by $P = S/(C_s \hat{\Phi})$, and the $|\mathcal{U}| \times |\mathcal{K}|$ matrix Q is given by $Q = D/(C_d \hat{\Psi})$. The inequalities and the division operators operate entry-wise on the corresponding matrices.

Since there are multiple constraints on W, we resort to the ADMM method [18], which enables us to keep multiple copies of variables in order to reduce each sub-problem to an easier problem with a single set of constraints. Concretely, we rewrite

the above optimization problem as the following

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \frac{1}{2}\|W-\hat{W}\|_F^2, \\ \text{subject to} \quad & -D^TW+P=Z_1, WS-Q=Z_2, \\ & Z_1 \leq \mathbf{0}, Z_2 \leq \mathbf{0}, \\ & \mathbf{0} \leq W \leq \mathbf{1},\ \mathrm{diag}(W)=\mathbf{0}. \end{aligned}$$

The augmented Lagrangian of this equivalent problem is given by

$$\begin{aligned} L(W,Z_1,Z_2,\Lambda_1,\Lambda_2) &= \frac{1}{2}\|W-\hat{W}\|_F^2 \\ &+\frac{\rho}{2}\|-D^TW+P-Z_1+\Lambda_1\|_F^2+\frac{\rho}{2}\|WS-Q-Z_2+\Lambda_2\|_F^2, \end{aligned}$$

where Λ_1 and Λ_2 are the Lagrange multiplier variables for the two inequality constraints, and $\rho > 0$ is a (suitably chosen) scaling parameter. We start by initializing W as $\hat{W}$, Z_1, Z_2, Λ_1, and Λ_2 as all-zero matrices, and in each ADMM iteration, we perform the following updates for each variable, until convergence:

a) W update: We solve the following sub-problem

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & h(W)=\frac{1}{2}\|W-\hat{W}\|_F^2 \\ & +\frac{\rho}{2}\|-D^TW+P-Z_1+\Lambda_1\|_F^2 \\ & +\frac{\rho}{2}\|WS-Q-Z_2+\Lambda_2\|_F^2, \\ \text{subject to} \quad & \mathbf{0} \leq W \leq \mathbf{1},\ \mathrm{diag}(W)=\mathbf{0}. \end{aligned}$$

We solve this problem using a proximal gradient algorithm [17], i.e., in each inner iteration, we perform a gradient step followed by a projection step, until convergence. The gradient step given by

$$W \leftarrow W-\tau\nabla_W h(W),$$

where $\nabla_W h(W) = W-\hat{W}+\rho D(D^TW-P+Z_1-\Lambda_1)+\rho(WS-Q-Z_2+\Lambda_2)S^T$. The projection step is given by

$$W \leftarrow \max\{\min\{W-\mathrm{diag}(\mathrm{diag}(W)),0\},1\},$$

where $\max$ and $\min$ denotes element-wise maximum and minimum operators. We select the step-size τ using backtracking line search [17].

b) Z update: The subproblems for Z_1 and Z_2 are trivial; the updates are given by

$$Z_1 \leftarrow \max\{-D^TW+P+\Lambda_1,0\},$$

and

$$Z_2 \leftarrow \max\{WS-Q+\Lambda_2,0\}.$$

c) Λ update: The Lagrangian multiplier matrices Λ_1 and Λ_1 updates are given by

$$\Lambda_1 \leftarrow \Lambda_1-D^TW+P-Z_1,$$

and

$$\Lambda_2 \leftarrow \Lambda_2+WS-Q-Z_2.$$

Note that this ADMM algorithm has convergence guarantees since the constraint set is linear [18].