# On the Efficiency of Social Recommender Networks

Felix W.

***Abstract*—We study a fundamental question that arises in social recommender systems: whether individuals' benefits from using social networks are compatible with the efficiency of the network in disseminating information/recommendations. To tackle this question, our studies consist of three components. First, we introduce a stylized stochastic model for recommendation diffusion. Such a model allows us to highlight the connection between user experience at the individual level and network efficiency at the global scale. We also propose a set of metrics for quantifying both user experience and the network efficiency. Second, based on these metrics, we extensively study the tradeoff between the two factors in a Yelp dataset, concluding that Yelp's social network is surprisingly efficient, though not optimal. Finally, we design a friend recommendation algorithm that can simultaneously address individuals' need to connect to high quality friends, and service providers' need to improve network efficiency in information propagation.**

## I. Introduction

High-quality recommender systems are the hallmark of many online services, such as Netflix, Yelp and Amazon. They are also critical to many organizations in traditional sectors with a strong online presence, including the New York Times, CNN and the PlayStation Plus gaming network. When a recommender system is effective in showing users relevant and useful items (*e.g.,* a book from Amazon, a piece of news from CNN, or a game from PS Plus), users are much more likely to make purchases and/or re-visit the recommender system websites, which can result in a substantial increase in revenue and site traffic.

A trend in the recent design of recommender systems is the introduction of the "social feature." For example, Yelp's ranking algorithms take friends' opinions into account in recommending restaurants. Netflix shows a user what her friends have recently watched. Amazon allows users to share their purchases on Facebook or Twitter.

Some benefits of such features are immediate: it is usually more fun to interact with friends, and hopefully, people will be able to influence each other to make purchases, or visit restaurants, which will further increase a website's revenue. In addition, social features can also impact the entire recommender system in some other less visible, but perhaps more important ways. Specifically, (1) at the individual level, friends are more trusted and tend to share common interests. Thus, individuals are are more likely to adopt friends' opinions or recommendations [1], [2]. And (2) at the macro-level, information/recommendations are more likely to be propagated through social ties. This diffusion mechanism plays an important role in promoting less well-known but excellent businesses/products in a recommender system [3], [4].

In other words, the social network in a recommender system plays at least two important roles: (1) it helps individuals improve their experience. Let us refer to this benefit as the *local utility* of the social network. (2) It also helps the recommender system to disseminate information or recommendations more efficiently. Similarly, let us refer to this benefit as the *global utility* of the social network.

Here, an important question arises:

*Are local utilities and global utilities of the social network in a recommender system compatible?*

More specifically, if the social network is constructed in such a way that user experience is optimized, does it come at the cost of being less effective in transmitting information? And vice versa: if we optimize the social network for information propagation, will user experience be sacrificed?

Such a question becomes more interesting as the social networks are formed in a decentralized manner: two people can establish a link so long as both parties agree to do so. They often do not have a clear idea how the entire network is formed. They also usually do not make the connection for the sole purpose of sharing opinions and recommendations. It is also not always clear to an individual whether the potential new friend will have high-quality recommendations. Thus, it would be quite surprising to find that this crude network formation mechanism can optimize local or global utilities.

**Our work and contributions** This paper presents an extensive study of the local and global utilities tradeoffs in a recommendation system's social network. We shall focus on a recently-released data set from Yelp. Our contribution can be summarized as follows.

*1. Proposal of a stochastic model for recommendation diffusion.* We propose a simple but natural stochastic model for the diffusion of recommendations in social networks. Based on this model, we propose a set of statistically-grounded yet intuitive metrics to quantify the local and global utilities of a social network. Intuitively, one's local utility measures the discrepency of her rating and her friends' rating. When the discrepency is small, friendship-based recommendation will have higher accuracy, which results in higher local utility (see Eq. (1)). Meanwhile, a network's global utility measures how fast a piece of recommendation travels. Here, average hitting time is used to measure the global utility (see Eq. (2)).

The set of metrics we propose here is fairly robust, and work even when the actual recommendation diffusion model is misspecified.

*2. Empirical studies of the Yelp dataset.* We next study, based on the proposed metrics, the tradeoff between local and global utilities in the Yelp dataset with 70k users writing 335k reviews on 15.5k businesses. Our discoveries are as follows.

• When a user has only one friend, it is unclear how much the user will benefit from his/her social ties. But when the users have more than one friends, we start to see improvements in the quality of the recommendations from friends.

• While the local utility is profound, the global efficiency is also quite good. It is quite comparable to a random graph. This is quite remarkable because random graphs are very efficient at propagating information. On the other hand, the tradeoff between global and local utilities are not optimized, so we need the third part of our studies, discussed below.

*3. An algorithm for social recommender network optimization.* We also design an algorithm for optimizing the social network that can simultaneously address individuals' need to establish links to high-quality friends and the service providers' need to improve network efficiency in information propagation. Recommender systems are an extensively studied area. We notice that there already exist numerous works that focuses on either improving user experience (see [5], [6] and references therein), or the network's efficiency (*e.g.,* [7], [8], [9], [10]). But to our knowledge, there exists no prior work that simultaneously addresses both objectives. We also believe that this is an interesting direction for future recommender systems research.

**Why study Yelp.** We have two reasons to focus on the Yelp dataset. First, to our knowledge, Yelp is the only large online serivce that makes both rating information and the social network information publicly accessible. Second, unlike other services such as Amazon or Netflix, Yelp does not directly use Facebook's API and its social network. Instead, it built its own in-house social network service. In this way, the users know they will interact with the friends they add only at the Yelp platform. Consequently, there will be much fewer connections established for reasons other than sharing information in Yelp.

## II. Our model and metrics

We shall start with a fairly natural but stylized model for the diffusion of recommendations/reviews in a social network. Based on this model, we may define both an individual's utility and network efficiency in an intuitive way. In the next section we will analyze this model and present a number of key results in computing the local and global utilities. Along the way, we will also discuss the scenarios where the model is misspecified, or when a portion of the data is missing.

Let us denote the social network as $G = \{V, E\}$, where $V = \{v_1, v_2, ..., v_n\}$. Let $B = \{b_1, b_2, ..., b_\ell\}$ be the set of businesses (*e.g.,* restaurants) of interest. We use the following discrete-time stochastic process to model how users visit the businesses and write reviews. Here, our process is parametrized by a "strength-of-tie" variable $W \in \mathbb{R}_+^{n\times n}$. Intuitively, the number $W_{i,j}$ represents how strong the social-tie between $v_i$ and $v_j$ is, *e.g.,* if $v_i$ and $v_j$ are close friends, then $W_{i,j}$ should be large. We also require $W_{i,j} = 0$ if $\{v_i, v_j\} \notin E$, *i.e.,* there should not be any tie between two users if they do not know each other. As the social network is formed in a decentralized manner, we shall imagine the value of $W_{i,j}$ to be chosen by $u_i$ and $u_j$ but nobody else.

Our discrete time process works as follows: at time $t = 0$, each business $b \in B$ will be visited by a user, namely $p_0(b) \in V$. After the user $p_0(b)$ visits $b$, he/she will form an opinion regarding $b$ (*i.e.,* a review), namely $R(p_0(b))$. Here, $R(\cdot)$ is a real number and it is large when the user has high opinion on $b$. In each of the subsequent timesteps $t \geq 1$ and for each $b$, one neighbor of $p_{t-1}(b)$ will visit the business, and the probability that $v$ will visit $b$ is proportional to $W_{v,p_{t-1}(b)}$. Let us call this user $p_t(b)$. After the user visits the business, he/she also generates a review $R(p_t(b))$. This assumption captures the fact that close friends are more likely to influence each other. Notice that in this stochastic model, we allow the user to revisit the same business multiple times, but for simplicity, we assume a user's review remains unchanged after a revisit.

We now are able to define both the global efficiency and local efficiency of the social network, based on this process.

**Local efficiency/individual's utility.** We leverage the following intuition to define local efficiency/individual's utility. When a user decides to visit a business because of her friend's influence, she will think her friend's review is useful if the review is consistent with her own opinion, and useless otherwise. Thus, we shall use the discrepencies between $R(p_{t-1}(b))$ and $R(p_t(b))$ to measure the social network's local utility.

We now formalize this intuition. Let $v$ be an arbitrary user, and $e_t(v)$ be the set of busiensses that $v$ visits at timestep $t$. Note that $e_t(v)$ could be an empty set if $v$ has not visited any businesses at time $t$. For an arbitrary $b \in e_t(v)$, let us write $v$'s review on $b$ as $R_b^t$ and $v$'s neighbor's review on $b$ at time $t-1$ as $R_b^{t-1}$. We shall define user $v$'s local cost as

$$\lim_{t\to\infty} \mathrm{E}\left[\frac{1}{|e_t(v)|}\sum_{b\in e_t(v)} |R_b^t - R_b^{t-1}| \;\middle|\; e_t(v) \neq \emptyset\right] \quad (1)$$

With slight abuse in notation, we also refer to (1) as $v$'s local utility.

We make a few remarks here. First, as we are interested in only the timesteps in which $v$ visits a business, we focus on the conditional expectation $\mathrm{E}[\cdot \mid e_t(v) \neq \emptyset]$. Second, we use $|R_b^t - R_b^{t-1}|$ to quantify the discrepency between $v$'s opinion and her friend's opinion. There could be other choices but we use the $\ell_1$-norm because it is more robust against outliers. Also, we need to average out the discrepencies in case $|e_t(v)| > 1$, as we believe the volume of a user's reviews should be orthogonal to the quality of $v$'s friend's review. Finally, we take $t \to \infty$ as we are more interested in the system's equilibrium behavior.

**Global utility.** Intutively, the global utility function should be able to measure how fast information propagates. Under our recommendation diffusion model, we use the average waiting time for each person to review a specific business. More specifically, let $v_i = p_0(b)$ be the first person that reviews business $b$. And let $T(v_j, b) = \min\{t : p_t(b) = v_j\}$. The average waiting time can be defined as the mean of $T(v_j, b)$, over all $v_j$, *i.e.,* $\frac{1}{n}\mathrm{E}[\sum_j T(v_j, b)]$. Furthermore, we also require the average waiting time to be uniformly good, regardless the starting point $p_0(b)$. Thus, our final objective is

$$\mathrm{E}_{p_0(b)\sim_U V}\left\{\frac{1}{n}\mathrm{E}\Big[\sum_j T(v_j, b)\Big]\right\}, \quad (2)$$

where $p_0(b) \sim_U V$ means that the first reviewer of $b$ is uniformly sampled from $V$.

## III. Analysis of utilities and model mis-specification

### A. Local utility.

**Efficient computation.** We now describe the procedure for estimating an individual's utility starting with the assumption that there is no missing data. As all users will rate all businesses as $t \to \infty$, we shall refer to $R_{i,k}$ as $v_i$'s rating on business $b_k$. Then we have the following proposition, which is proved in the Appendix.

*Proposition 3.1:* Consider the aforementioned diffusion model. For a specific user $v_i$, we have

$$\begin{aligned}&\lim_{t\to\infty} \mathrm{E}\left[\frac{1}{|e_t(v_i)|}\sum_{b\in e_t(v_i)} |R_b^t - R_b^{t-1}| \;\middle|\; e_t(v_i) \neq \emptyset\right]\\ =\ & \frac{1}{\ell}\sum_{k\le\ell}\sum_{j}\frac{W_{i,j}|R_{i,k}-R_{j,k}|}{\sum_j W_{i,j}}\end{aligned}$$

Then by Proposition 3.1, the local utility metric $D_i$ of user $i$ can be calculated as

$$D_i = \frac{1}{\ell}\sum_k \frac{\sum_{j=1}^n W_{i,j}|R_{j,k}-R_{i,k}|}{\sum_{j=1}^n W_{i,j}} \tag{3}$$

To summarize the local utility of the whole network, we take average over all users:

$$\overline{D} = \frac{1}{n}\sum_{i=1}^n D_i \tag{4}$$

**Missing data.** We consider an approximation to Eq. (3) to account for the missing data problem in real datasets, *i.e.,* there is a large portion of $j$ and $k$ such that $R_{j,k}$ is not available in the data set. When $R_{j,k}$ is missing, we shall use the population mean of $k$ as our estimate of $R_{j,k}$. This is justified by the fact that the population mean is the maximum-likelihood estimate of $R_{j,k}$, assuming the ratings of $k$ are i.i.d. Thus, we use the formula below to compute a user's local utility:

$$D_i = \frac{1}{|\mathcal{R}_i|}\sum_{k\in\mathcal{R}_i} \frac{\sum_{j=1}^n W_{i,j}|\tilde{R}_{j,k}-R_{i,k}|}{\sum_{j=1}^n W_{i,j}} \tag{5}$$

$$\tilde{R}_{j,k} = \begin{cases} R_{j,k} & R_{j,k} \text{ available} \\ r_k & \text{otherwise,}\end{cases} \tag{6}$$

where $\mathcal{R}_i = \{k \mid R_{i,k} > 0\}$, and $r_k$ is the population mean of $k$. While the above equation is motivated by our stylized model, it also has a fairly natural interpretation, and thus even when our model is mis-specified, Eq. (5) still offers a reasonable approach to estimate local utilities. More specifically, the above formula possesses two interesting properties:

*1. Having no friends does not harm you.* If none of your friends have ever rated a business, the social network should not harm you, *i.e.,* you should get the average reviews.

*2. Having no or few ratings from friends is informative.* If some of your friends have rated a business, then we expect the recommender system to take into account the following two factors: (a) the values of the friends' ratings, and (b) the proportion of friends that have rated the business. Consider two scenarios, where in the first one, only 1% of your friends have rated the business, while in the second one, more than 40% of your friends have rated it. Our formula gives a metric that assigns less weight on friends' opinions for the first case, and more weight on friends' opinions for the second case.

### B. Global utility.

**Computation.** The hitting time $H_{i,j}$ is the time taken for user $i$ to hear about a business that was originally discovered by user $j$. Averaging over all pairs of users, we define $\overline{H}$ as

$$\overline{H} = \frac{1}{n(n-1)}\sum_{i,j=1}^n H_{i,j} \tag{7}$$

which is in turn related to the eigenvalues of the unnormalized graph Laplacian $L = \mathrm{diag}(W\mathbf{1}) - W$. Let the sorted eigenvalues of $L$ be $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, then it can be proved [8] that $\overline{H}$ and $\{\lambda_i\}$ are related as

$$\overline{H} = \frac{2\sum_{i,j=1}^n W_{i,j}}{n(n-1)}R_{\text{tot}} = \frac{2\sum_{i,j=1}^n W_{i,j}}{n(n-1)}\sum_{i=2}^n \frac{1}{\lambda_i}, \tag{8}$$

where $R_{\text{tot}} = n\sum_{i=2}^n (1/\lambda_i)$ is known as the effective resistance of $W$. Note that a smaller $\overline{H}$ means better global utility.

**Robustness.** Let us start by considering plausible candidates for measuring global utility. Recall a network is "high-dimensional" as there can be as many as $\binom{n}{2}$ edges in a network with $n$ nodes (so we need $\binom{n}{2}$ independent variables to represent it). Thus, we shall look for a concise statistic that can well summarize its efficiency in information propagation.

This is an extensively studied area and numerous stochastic models have been proposed (see [9], [10], [11], [12]). We observe that the efficiencies in information propagation among all these models are invariably characterzied by one of these closely related statistics: (1) the second largest eigenvalue of the Laplacian or transition matrix [13], (2) the conductance of the network [9], (3) the mixing time, and (4) the hitting time, as used in our model. All these statistics are closely related, *e.g.,* if we know the average hitting time of a Markov chain, we can compute (tight) upper and lower bounds for the mixing time, conductance, and second largest eigenvalue. Thus, all these statistics are roughly equivalent, *i.e.,* when the social network is optimized for average hitting time, it is also asymptotically optimal in almost all other information diffusion models (see [13] for more discussion), and the average hitting time is a fairly robust statistic.

We also remark that the second largest eigenvalue or the mixing time are less appropriate statistics in our setting for the following reasons. (1) Even in our simple model, the mixing time or second largest eigenvalue alone is insufficient to bound the average hitting time. Instead, we also need knowledge on the stationary distribution (see Fact 3.1). And (2) we prefer a simpler model with less parameters (*i.e.,* not specifying the stationary distribution) by Occam's Razor principle. And there is also no natural way to interpret the stationary distirbution of a social network.

*Fact 3.1:* There exists a growing family of networks $\{M(\tau)\}_{\tau>0}$ such that (a) the mixing time of $M$ is 1, (b) the second largest eigenvalue of the associated transition matrix is 0, and (c) the average hitting time tends to $\infty$ as $\tau \to \infty$.

## IV. DATASET AND ASSOCIATED GRAPHS

The social recommender network we draw data from is the Yelp social network from Phoenix, Arizona.[1] The dataset consists of 335,022 reviews/ratings made by 70,817 users on 15,585 businesses. The dataset also includes the users' social network with 151,516 edges. Our analysis focuses on two graphs. The first graph is Yelp's social network constructed by the users in a decentralized but strategic manner. The second graph is built such that, roughly speaking, it represents how users interact with each other, either directly or indirectly in Yelp. For instance, if two users co-rated a large number of businesses, we consider them to have strong (indirect) interactions. We use the second graph as the baseline graph as it represents how users "organically" interact with each other, when no intervention is presented. The first graph is the focus of our studies. We would like to understand how the tradeoff between local and global utilities changes when the social network is formed by a strategic and decentralized way. Below, we describe the details of these two graphs.

**Social graph.** The Yelp social network is built in house (*i.e.,* it does not solely rely on Facebook or other 3rd party's API to manage the network) though users may import their friends from email or Facebook accounts. Following related work in OSN analysis [14], we focus on the largest connected component in the Yelp social graph for our analysis. While the largest connected component has only 28,977 users, the remaining users appear not interested in any social activity, as seen from the second largest component having only 4 users. Our analysis is meaningful only for the users who care about building the social network. To allow for a direct comparison with the Yelp co-rating graph (see below), we further remove 330 users that do not have sufficient reviewing activity and are isolated in the co-rating graph. Overall, we have a social graph of 28,647 users and 149,348 edges.

**Co-rating graph.** We create the baseline graph of users such that two users share an edge if and only if they have co-rated at least one business. This graph can be considered as the largest network that can be discovered by users reading reviews of others.[2] Since our metrics are based on reviews and ratings, we also expect the optimal network to be embedded in this graph with a suitable choice of edge weights and topology (*i.e.,* pruning useless edges). Considering only the 28,647 users, this graph has 8.13m edges, 54 times more than the social graph.

Figure 1 shows the basic properties of the data in terms of making friends and reviewing businesses, the two major forms of user activity. The degree distribution exhibits higher skewness than review activity, which indicates more effort is required in reviewing a business. Nevertheless, the two forms of user activity are positively correlated, especially towards the tail (see Figure 1(c)).

Given the two base graphs, we consider variations of them through altering their topology and edge weights.

**Perturbing network topology.** We also compare the social graph with a random graph baseline. There are two reasons for doing so. (1) A random graph represents the "average behavior" of all graphs sharing the same degree distribution. Thus, such a comparison allows us to understand how the social graph differs from an "average graph." (2) Random graphs are known to have small mixing time, small conductance, and small second largest eigenvalues [15], *i.e.,* they are very efficient in information propagation. We consider a network efficient if its global utility is comparable to random graphs.

Given the social graph, we shuffle its edges to obtain a perturbed graph. Algorithm 1, similar to that in [16], ensures the resultant graph is connected (by checking for connectedness every `window` steps, and if not, reverting to the previous snapshot $G'$) and the degree of each node is conserved. By increasing `num_iter`, we obtain a family of graphs with increasing distance (in terms of the number of differing edges) from the original social graph, and in the limit `num_iter` $\to \infty$, we obtain a random graph such that each node's degree is the same as its degree in the social graph.

**Algorithm 1** Degree and connectedness-preserving shuffling.

**Input:** $G = \{V, E\}$, `num_iter`, `window`
  **for** $i = 1$ to `num_iter` **do**
    $G' \leftarrow G$
    **repeat**
      $G \leftarrow G'$
      **for** $j = 1$ to `window` **do**
        Sample two edges $\{a, b\}$ and $\{c, d\}$ from $E$
         s.t. $a \neq c$, $a \neq d$, $b \neq c$, $b \neq d$, and $\{a, c\}, \{b, d\} \notin E$
        Remove $\{a, b\}$ and $\{c, d\}$ from $E$
        Add $\{a, c\}$ and $\{b, d\}$ to $E$
      **end for**
    **until** $G$ is connected
  **end for**

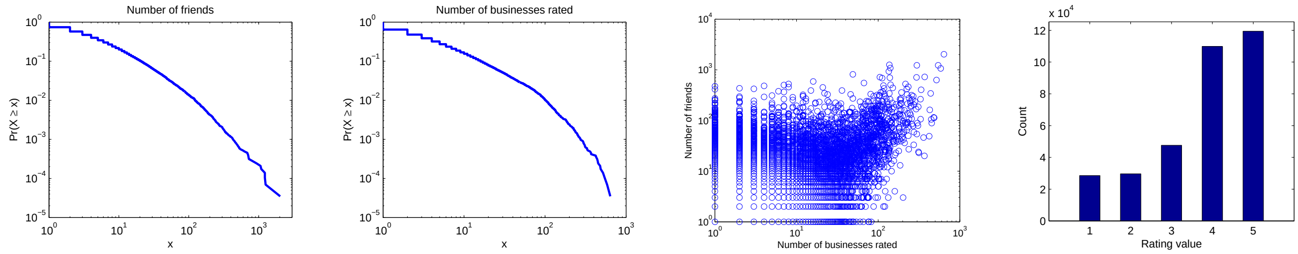
**Scaling edge weights.** In reality, we expect each user to prioritize over different friends' recommendations, so we consider several options of edge weighting based on ratings.

- Jaccard coefficient on rating sets: $W_{ij} = \frac{|\mathcal{R}_i \cap \mathcal{R}_j|}{|\mathcal{R}_i \cup \mathcal{R}_j|}$. This captures the likelihood of a pair of users interacting through reading and writing reviews.
- Pearson correlation coefficient on rating vectors:

$$W_{ij} = 1 + \frac{\sum_{k \in \mathcal{R}_{ij}} (R_{ik} - r_i)(R_{jk} - r_j)}{\sqrt{\sum_{k \in \mathcal{R}_{ij}} (R_{ik} - r_i)^2} \sqrt{\sum_{k \in \mathcal{R}_{ij}} (R_{jk} - r_j)^2}}$$

[1] `http : //www.yelp.com/dataset_challenge`

[2] While it is possible that two users become friends even if they have not co-rated anything, we consider this unlikely because a friendship relation is symmetric, *i.e.,* two users must have mutual interests and are likely to have rated sufficiently many businesses to get at least one co-rated.




(a) Degree distribution. (b) Review activity distribution. (c) Degree correlates with reviews. (d) Rating distribution.

Fig. 1: Basic data statistics.

where $\mathcal{R}_{ij} = \mathcal{R}_i \cap \mathcal{R}_j$, $r_i = \sum_{k \in \mathcal{R}_i} R_{ik}/|\mathcal{R}_i|$, and the 1 is added to make $W_{ij} \in [0, 2]$

- Cosine similarity on rating vectors:

$$W_{ij} = \frac{\sum_{k \in \mathcal{R}_{ij}} R_{ik} R_{jk}}{\sqrt{\sum_{k \in \mathcal{R}_{ij}} R_{ik}^2} \sqrt{\sum_{k \in \mathcal{R}_{ij}} R_{jk}^2}}$$

defined such that $W_{ij} \in [0, 1]$ ($W_{ij} \geq 0$ because $R_{ij} \geq 0$).

## V. Empirical Data Analysis

### A. Metrics

We compare the metrics of the graphs with different weight scaling strategies. Table I summarizes the results:

• The Yelp social graph has better global utility $\overline{H}$ than the other graphs except for the random graph baseline.[3] This is expected because random graphs have better efficiency properties as discussed in Section IV, but the difference is only moderate (only 12% degradation). In fact, we believe that most of the edges in the random graphs are not actually discoverable by users. Also, the slight loss in global utility is compensated by the improvement in $\overline{D}$.[4]

It is not surprising to see that the social graph has better global utility than the uniformly weighted co-rating graph, even though the latter has many more edges. We note that the $\overline{H}$ metric does not monotonically improve with the addition of edges. In fact, when the co-rating graph is used, the spread of a specific message could "get lost" in the network (due to having excessive amount of edges), reminiscent of findings from social navigation studies [17]. Perhaps more surprisingly, even when we use the other three similarity measures to weigh edges, $\overline{H}$ does not improve. This implies finding good edge weights is not straightforward.

• Another finding is that the social graph performs better than all other graphs under the $\overline{D}$ metric. This confirms that users are conscious of local utility when forming the social network. Comparing the three similarity measures, Pearson coefficient performs significantly better than the other two. To weigh confidence level into user similarity, *i.e.,* assign higher similarity to user pairs with more co-rated businesses, we also take the product of Jaccard and Pearson/Cosine coefficients as edge weights, but this strategy does not help in improving $\overline{D}$.

| Network | $\overline{H}$ | $\overline{D}$ |
|---|---|---|
| Social, uniform | $1.68 \times 10^5$ | 0.882 |
| Random, uniform | $1.47 \times 10^5$ | 0.890 |
| Co-rating, uniform | $7.28 \times 10^5$ | 0.994 |
| Co-rating, Jaccard | $1.03 \times 10^6$ | 0.993 |
| Co-rating, Pearson | $4.23 \times 10^6$ | 0.955 |
| Co-rating, Cosine | $7.30 \times 10^5$ | 0.994 |
| Co-rating, Jaccard ∘ Pearson | $6.15 \times 10^6$ | 0.961 |
| Co-rating, Jaccard ∘ Cosine | $1.03 \times 10^6$ | 0.992 |

TABLE I: Network metrics on connected networks. Results of "random" the average over 10 random graph instances.

| Network | $\overline{D}$ | Network | $\overline{D}$ |
|---|---|---|---|
| Yelp prior (no network) | 0.887 | Social, Jaccard | 0.875 |
| Social, Jaccard ∘ Pearson | 0.872 | Social, Pearson | 0.872 |
| Social, Jaccard ∘ Cosine | 0.875 | Social, Cosine | 0.876 |

TABLE II: Local utility of disconnected networks.

We can also build a "hybrid" graph, where the graph is induced by the social graph but the edge weights are calculated from the three similarity measures discussed above. Surprisingly, only about 29% of the edges in the social graph appear also in the co-rating graph, and taking the edge intersection of the two graphs results in a disconnected graph. This suggests that many users tend to import friends from other OSNs.

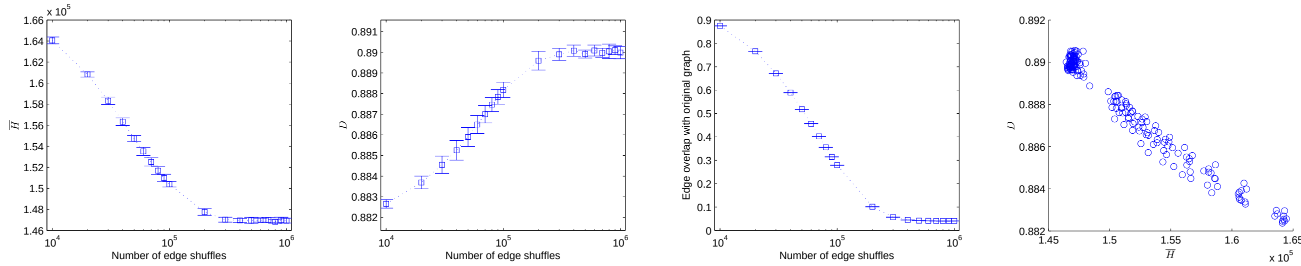
Even for a disconnected graph, we can still compute its $\overline{D}$ (note $\overline{H} = \infty$ in this case), and Table II shows the results for different graph-weight combinations. Our baseline of $\overline{D}$ is that of "Yelp average," which corresponds to users relying only on Yelp-provided average ratings and not consulting any friends. Somewhat surprisingly, "Yelp average" performs better than all $\overline{D}$'s reported in Table I, except for that of the social graph. This supports our claim that users form a social network to improve their local utility. Weighing edges of the social graph by Jaccard/Cosine/Pearson coefficients further improve $\overline{D}$. In Section V-C we consider a modification to weigh social graph edges that preserve graph connectness.

### B. Varying Graph Randomness

The purpose of this and the next section is to study the tradeoff between global and local utilities. Consider interpolating the social graph and its random graph counterpart, with randomness quantified by the number of edge shuffles. Figures 2(a) and 2(b) show that as randomness increases, global utility improves but local utility degrades. This tradeoff is exemplified by Figure 2(d), a scatterplot showing strong

[3] Generated by applying Algorithm 1 on the social graph such that 1m edge shuffles are performed by setting `num_iter` = `window` = 1000.

[4] Alternatively, we compare two graphs' local utilities by a Wilcoxon signed rank test on their $D_i$ vectors. The results are consistent with $\overline{D}$ comparison, *i.e.,* when one graph's $\overline{D}$ is better than that of the other, we also have the test rejecting the null hypothesis of two populations ($D_i$ vectors) being equal at 5% level. The only exception is discussed in Section V-D.




(a) More randomness improves $\overline{H}$. (b) More randomness degrades $\overline{D}$. (c) Similarity with original graph. (d) Tradeoff between $\overline{H}$ and $\overline{D}$.

Fig. 2: Results from perturbing social graph. For each number of shuffles, 10 graphs are generated with 1 s.d. reported.

negative correlation between the two metrics.

Empirical results [16] show that similar edge shuffling strategies produce a random graph with $O(m)$ shuffles, where $m$ is the number of edges in the original graph. Since in Figure 2 the results stabilize roughly at 200k shuffles, we are convinced that any graph generated with more than 200k shuffles are close to random. Figure 2(c) further supports our claim by showing the proportion of edges that overlap with the social graph to be small (about 4%).

### C. Varying Edge Weight Skewness

Here we consider a edge weight skewing strategy to study the tradeoff between $\overline{H}$ and $\overline{D}$. From Tables I and II we choose to build a graph by combining the social graph and the co-rating graph weighted by Pearson coefficient, because this combination appears to be able to generate the best $\overline{H}$ and $\overline{D}$. We apply a weight transformation parameterized by $\alpha, \beta > 0$: $W_{ij} = (W_{ij,\text{Pearson}} + \beta)^{\alpha}$ for $W_{ij,\text{social}} > 0$ and $W_{ij} = 0$ otherwise. Note that setting $\beta > 0$ ensures the graph is connected, and weight skewness decreases with decreasing $\alpha$ or increasing $\beta$. In the limits $\alpha = 0$ or $\beta \to \infty$, the graph is equivalent to the social graph with uniform weights.

Figure 3 shows a tradeoff of $\overline{H}$ and $\overline{D}$ under this weight skewing strategy: for $(\alpha, \beta)$ values that result in a good $\overline{H}$ (blue areas), the corresponding $\overline{D}$ is bad (red areas), and vice versa. This is exemplified by the scatterplot of Figure 3(c).

### D. Does Social Network Help in Collaborative Filtering?

We first investigate the distribution of $\{D_i\}$ associated with Yelp's social graph (Figure 4(a)). One can see that most users' $D_i$'s are between 0 and 1, but there is also a considerable portion of outliers that come from the population with few friends (Figure 4(b)). While local utility improves as a user build more connections, it does not prove that having more friends improves social recommendations, because it is possible that when a user is more active, she will both have more friends and have better $D_i$ (*i.e.,* being active is a confounder of $D_i$ and number of friends). Indeed, we notice that a user's $D_i$ decreases as she produces more reviews (see Figure 4(c)). Also, the number of reviews one produces is positively correlated with the number of friends she has.

To eliminate the effect of confounding, we carry out statistical tests, enabled by our local utility metric, on populations with similar activity levels. First, we run a signed rank test on the two $\{D_i\}$ vectors of the full population due to the social graph and the Yelp prior. Although the social graph has a lower $\overline{D}$, the test is inconclusive because it fails to reject at 5% level (p-value: 0.915) the null hypothesis that the two $\{D_i\}$ are equal. However, if we run the test on the $\{D_i\}$ vectors restricted to users with at least two friends, we successfully reject the null hypothesis (p-value: $6.07 \times 10^{-3}$). Thus, we can conclude that when a user makes more than one connection, she will see better recommendations from her friends.

## VI. Social Recommender Network Optimization

### A. Optimization Problem

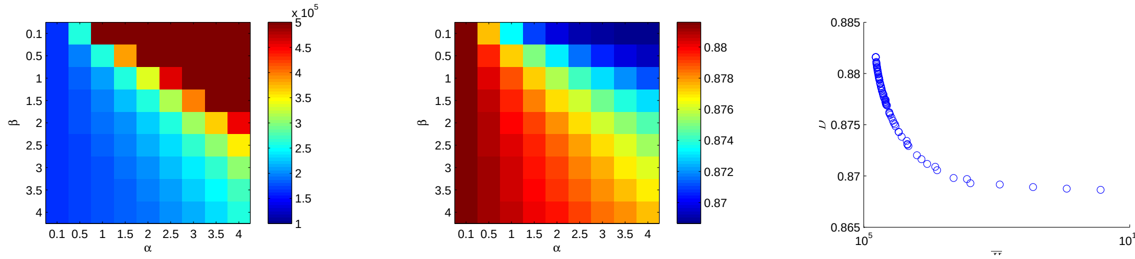
Even though the Yelp social network performs well in balancing global and local utilities, it is possible to do better through optimization. If an OSN service provider knows what the optimal network is, it can facilitate network formation with strategies like friend recommendation and news feed curation (*e.g.,* promote updates of more "useful" friends). Here our goal is to find optimal edge weights $W$, so that we recommend users $i$ and $j$ to become friends if currently $W_{ij} = 0$ but the optimal $W^*_{ij} > 0$, and news feed curation is equivalent to scaling the frequency of updates by $W^*$.

Consider the scenario where each user $i$ has a constraint on her local utility $D_i \le \delta_i$, where $\delta_i$ the worst tolerable utility observed from existing data, *e.g.,* measuring $D_i$ on the given Yelp social graph. In addition, we are given an allowable edge set $E$ with $|E| = m$. Then social recommender network optimization problem becomes maximizing network efficiency, subject to topology and local utility constraints:

$$\begin{array}{lll} \underset{W \in \mathbb{R}^{n\times n}}{\text{minimize}} & \overline{H}(W) & \\ \text{subject to} & D_i(W) \le \delta_i & i = 1, 2, \ldots, n \\ & W_{ij} = 0 & \text{for } \{i, j\} \notin E \\ & W = W^T, W \ge 0 & \end{array} \tag{9}$$

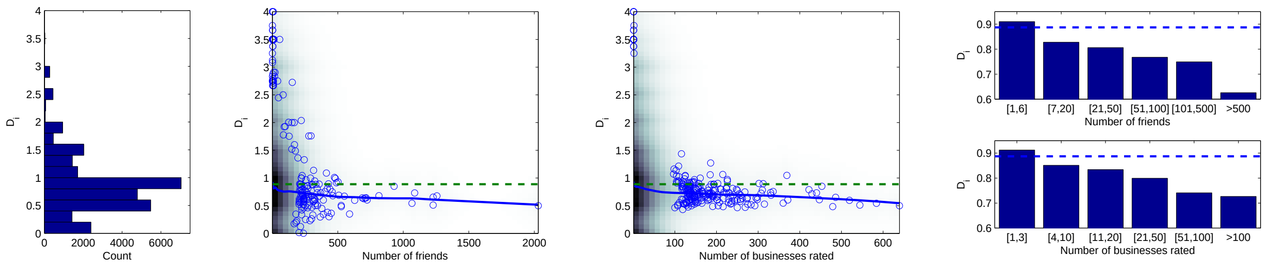
Since both $\overline{H}$ and $\{D_i\}$ are invariant to scaling, *i.e.,* $W$ and $\alpha W$ (for any $\alpha > 0$) result in the same $\overline{H}$ and $\{D_i\}$, problem (9) can be rewritten in terms of effective resistance (see Section III-B for a discussion):

$$\begin{array}{ll} \underset{W \in \mathbb{R}^{n\times n}}{\text{minimize}} & R_{\text{tot}}(W) \\ \text{subject to} & D_i(W) \le \delta_i \quad i = 1, 2, \ldots, n \\ & W_{ij} = 0 \quad \text{for } \{i, j\} \notin E \\ & W = W^T, W \ge 0, \sum_{i,j=1}^{n} W_{ij} = m \end{array} \tag{10}$$




(a) Less weight skew improves $\overline{H}$. (b) More weight skew improves $\overline{D}$. (c) Tradeoff between $\overline{H}$ and $\overline{D}$.

Fig. 3: Effect of skewing edge weights by controlling $(\alpha, \beta)$ parameters.




(a) Distribution of $D_i$. (b) $D_i$ decreases with degree. (c) $D_i$ decreases with review activity. (d) $D_i$ decreases with activity.

Fig. 4: Distribution of $D_i$ on Yelp social graph. For (b) and (c), 2D histograms are shown with 1% outliers overlaid. Dashed lines are $\overline{D}$ baseline due to Yelp prior, and solid lines are locally weighted linear regression fit of data.

where we add the edge weight sum constraint $\sum_{i,j} W_{ij} = m$; otherwise one can set $W$ arbitrarily large to make $R_{\text{tot}}$ small. Then we can show the following.

*Lemma 6.1:* Optimization problem (10) is convex.

*Proof:* We need to verify the convexity of the objective function and the constraints. $R_{\text{tot}}$ has been shown to be convex in [8], and the constraints $W = W^T$, $W \geq 0$ and $\sum_{i,j=1}^{n} W_{ij} = m$ are affine. Hence it remains to show $D_i \leq \delta_i$ is convex. By (5) and rearranging, $D_i \leq \delta_i$ is equivalent to

$$\sum_{j=1}^{n} W_{ij} \left( \sum_{k \in \mathcal{R}_i} |\tilde{R}_{j,k} - R_{i,k}| - \delta_i |\mathcal{R}_i| \right) \leq 0$$

which is another affine constraint. ■

## B. Algorithm

Our problem of finding optimal edge weights is solved on networks with at least 100k edges. Off-the-shelf tools based on interior point and SDP methods are prohibitively expensive on problems of this size. Thus we use a projected gradient method instead. In the following, it is more convenient to express the variables as a vector $w = \{w_l\}$, for all $l \in E$. Let $E_i = \{l \mid l = \{i,j\} \text{ any } j\}$, then we rewrite problem (10) as

$$\begin{aligned} &\underset{w \in \mathbb{R}^m}{\text{minimize}} \quad R_{\text{tot}}(w) \\ &\text{subject to} \quad w \geq 0, \ \mathbf{1}^T w = m \\ &\qquad \sum_{l=\{i,j\} \in E_i} w_l \left( \sum_{k \in \mathcal{R}_i} |\tilde{R}_{j,k} - R_{i,k}| - \delta_i |\mathcal{R}_i| \right) \leq 0 \end{aligned} \tag{11}$$

Deriving the optimization algorithm involves two steps.

**Deriving the gradient.** From [8] we have for $l = \{i,j\}$,

$$\begin{aligned} \frac{\partial R_{\text{tot}}}{\partial w_l} &= -n \|(L + \mathbf{1}\mathbf{1}^T/n)^{-1} a_l\|_2^2 \\ &= -n \|L^\dagger a_l + \mathbf{1}\mathbf{1}^T a_l / n\|_2^2 \\ &= -n \|L^\dagger (e_i - e_j) + (e_i - e_j)/n\|_2^2 \end{aligned} \tag{12}$$

where $L^\dagger$ is the pseudoinverse of $L$, $a_l$ is the vector with $(a_l)_i = 1$, $(a_l)_j = 0$, $(a_l)_k = 0$ for $k \neq i,j$, and $e_i$ is the vector with $(e_i)_i = 1$ and $(e_i)_k = 0$ for $k \neq i$. The second step is due to $(L + \mathbf{1}\mathbf{1}^T/n)^{-1} = L^\dagger + \mathbf{1}\mathbf{1}^T/n$ (Eq. (7) of [8]).

Naively computing the pseudoinverse $L^\dagger$ is prohibitively expensive with a time complexity of $O(n^3)$, but there exist efficient Laplacian solvers (which solves for $x$ in the system $Lx = b$), and we use the solver in [18] to compute $L^\dagger e_i$. The gradient vector $\nabla R_{\text{tot}}$ can be computed in $\tilde{O}(mn)$ time.

**Deriving projections.** Projecting $w$ to satisfy the constraints can be cast as a linearly-constrained quadratic program. Define $w_i \in \mathbb{R}^{|E_i|}$ as the vector of $\{w_l\}$ for $l \in E_i$, and $b_i \in \mathbb{R}^{|E_i|}$ such that $(b_i)_l = \sum_{k \in \mathcal{R}_i} |\tilde{R}_{j,k} - R_{i,k}| - \delta_i |\mathcal{R}_i|$ for $l = \{i,j\}$. The last constraint can be written as $b_i^T w_i \leq 0$. Then computing the projection is equivalent to solving

$$\begin{aligned} &\underset{x \in \mathbb{R}^m}{\text{minimize}} \ \|x - w\|_2^2 \\ &\text{subject to } x \geq 0, \ \mathbf{1}^T x = m, \ b_i^T x_i \leq 0, \ i = 1, \ldots, n \end{aligned} \tag{13}$$

where $x_i$ is defined with the same indexing scheme as $w_i$. We use Gurobi [19] to solve this quadratic program. Algorithm 2 summarizes the our solution approach to problem (11).

## C. Results

We implement our algorithm on a machine with two Intel E5540 processors and 16GB RAM. Then we input $E$ as the

**Algorithm 2** Projected gradient descent to solve (11).

**Input:** $G = \{V, E\}, R, \tilde{R}, \delta, t$
  Initialize: $w \leftarrow \mathbf{1}$
  **repeat**
    $w \leftarrow w - t \cdot \nabla R_{\text{tot}}$ with $\nabla R_{\text{tot}}$ as calculated by (12)
    $w \leftarrow x^*$, where $x^*$ is solution to problem (13)
  **until** $w$ converges

edges of the Yelp social graph to study the how the algorithm modifies the graph with uniform edge weights. While not studied in this paper, we note that friend recommendation can easily be incorporated by expanding the $E$ set. Compared to the graph with uniform edge weights, we see substantial improvement in the $\overline{H}$, *i.e.,* the optimal graph achieves $\overline{H} = 5.78 \times 10^4$, a 65% reduction in $\overline{H}$, and a slightly better $\overline{D} = 0.879$. The hitting time of this optimal graph is also substantially better than that of random graphs.

**Statistics of the optimized graph** We are interested in how the optimized graph "looks like". First, we check whether the degree distribution still follows power law. Figure 5(a) shows the log-log plot of the degree distribution. We notice tht the key difference between the optimized graph and a standard social network is that there are very few nodes with degree $\leq 5$. In general, it has been known that having too many small-degree nodes will increase the mixing time of a graph. Our optimization procedure recognizes this problem and avoids the presence of small-degree nodes as much as possible. Nodes with degree $\geq 5$ follow a power law distribution well.

Second, we would also like to understand the distribution of edge weights (see Figure 5(b)). We see a double phase transition effect, *i.e.,* approximately half of the weights concentrate at $10^{-7}$ (which is essentially 0) and 35% of the edges with a weight of above 1 serve as the backbone that connects 99.6% of the network. The maximum edge weight is 14.2. Thus, it appears that edges with weights below 1 are not needed in the optimal solution.

**Correlations between the original and the optimal graph.** Next, we study the correlation between the original and the optimal graph. Figure 5(c) shows the result for the correlation of degrees. We can see that while in general the degrees in both graphs are correlated, many nodes have substantial changes in weights. This shows global surgery is need to turn the original graph into optimal. Second, we also compute the Jaccard coefficients of the $k$ highest degree nodes for all $k$. We can see that for $k$ below $10^4$ (recall that there are a total number of 28k nodes), the Jaccard coefficients are consistently below 0.5. This implies that the overlap between "important" nodes is not substantial.

## VII. Related Work

Our global utility metric based on average hitting time is related to proposals of using effective resistance [20] and related measures [21], [22] to quantify network robustness. These works implicitly assume a network routing model for information dissemination, so redundant paths are beneficial to network robustness, but this may not be the case for our hitting time-based global utility metric.

Empirical analyses of the utility of OSNs have traditionally focused on global properties like mixing time [14] and information diffusion [23], but some recent works have also addressed local utility. [24] proposes a random walk-based metric to study the tradeoff between privacy and local utility in the context of privacy-preserving graph perturbation. [25] shows high precision and fast dissemination can simultaneously be achieved in some broadcast networks. [26] studies content curation in social media at a microscopic level. Our work takes an extra step in designing algorithms for network formation that jointly optimizes local and global utilities.

Network topology optimization with second largest eigenvalue modulus [7] or effective resistance [8] have found applications in, *e.g.,* computer networks [27] and power networks [28]. Our contribution is in applying the techniques on OSNs with constraints on a novel local utility metric. We also remark that the scale of our problem (>100k edges) is much larger than those seen in existing work (1k edges), and this poses unique computational challenges.

## VIII. Conclusions

Motivated by a stochastic model for diffusion of recommendations, we propose a set of metrics to measure the local and global utilities in a social recommender network. We estimate these metrics on a recently released dataset and make a number of interesting discoveries: (a) at the individual level, users are good at identifying connections that can help to improve the quality of recommendations they receive. So long as a user has more than one connection, the improvement is statistically significant, and (b) at the macro level, the average hitting time for the social graph is only moderately worse than randomly generated graphs, while the latter ones have been known to be efficient in disseminating information. We also examined how we can construct social graphs that simultaneously optimizes local and global utilities. While our optimization problem is convex, it is a large scale problem which prompts for a hybrid solution that blends in-house developed algorithmic techniques together with optimization technologies for sparse matrices and quadratic programs. In general, we believe a better understanding the tradeoff between user experience and network efficiency in information propagation is an important research direction in recommender systems research.

## References

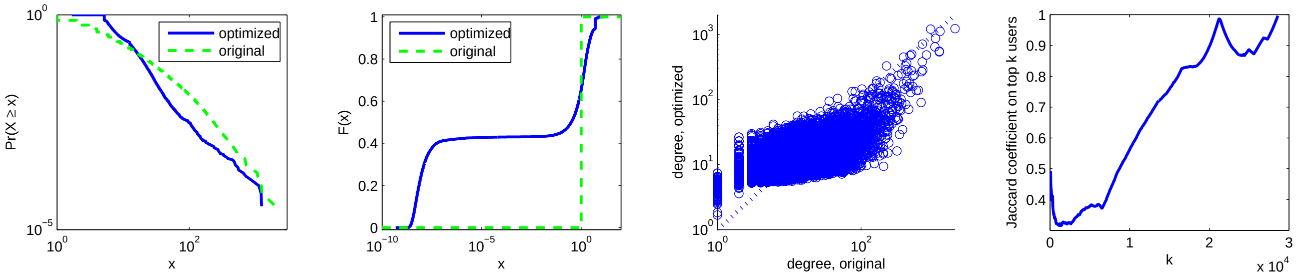

[1] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD*, 2003.

[2] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincn, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence maximization in social networks when negative opinions may emerge and propagate." in *SDM*, 2011.

[3] M. Granovetter, "The strength of weak ties: A network theory revisited," *Sociological Theory*, vol. 1, 1982.

[4] F. E. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 1, 2008.

[5] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.




(a) Degree distribution. (b) CDF of edge weights. (c) Optimization makes node degrees more equal. (d) Jaccard coeff. on top $k$ user sets of uniform and optimal social graphs.

Fig. 5: Results from optimizing edge weights of Yelp social graph.

[6] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009.

[7] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM Review*, vol. 46, no. 4, 2004.

[8] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Review*, vol. 50, no. 1, 2008.

[9] F. Chierichetti, S. Lattanzi, and A. Panconesi, "Almost tight bounds for rumour spreading with conductance," in *STOC*, 2010.

[10] M. Abdullah, C. Cooper, and M. Draief, "Viral processes by random walks on random regular graphs." in *APPROX-RANDOM*, 2011.

[11] H. Lam, Z. Liu, M. Mitzenmacher, X. Sun, and Y. Wang, "Information dissemination via random walks in d-dimensional space." in *SODA*, 2012.

[12] A. Clementi, R. Silvestri, and L. Trevisan, "Information spreading in dynamic graphs," in *PODC*, 2012.

[13] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[14] A. Mohaisen, A. Yun, and Y. Kim, "Measuring the mixing time of social graphs," in *IMC*, 2010.

[15] B. Bollobás, *Random graphs*. Academic Press, 1985.

[16] F. Viger and M. Latapy, "Fast generation of random connected graphs with prescribed degrees," in *COCOON*, 2005.

[17] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *STOC*, 2000.

[18] O. E. Livne and A. Brandt, "Lean algebraic multigrid (LAMG): Fast graph laplacian linear solver," *SIAM J. Sci. Comput.*, vol. 34, no. 4, 2012.

[19] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2014. [Online]. Available: http://www.gurobi.com

[20] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, and R. Kooij, "Effective graph resistance," *Linear Algebra and its Applications*, vol. 435, no. 10, 2011.

[21] A. Tizghadam and A. Leon-Garcia, "Autonomic traffic engineering for network robustness," *IEEE JSAC*, vol. 28, no. 1, 2010.

[22] G. Ranjan and Z.-L. Zhang, "Geometry of complex networks and topological centrality," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 17, 2013.

[23] D. Wang, H. Park, G. Xie, S. Moon, M.-A. Kaafar, and K. Salamatian, "A genealogy of information spreading on microblogs: A Galton-Watson-based explicative model," in *INFOCOM*, 2013.

[24] P. Mittal, C. Papamanthou, and D. Song, "Preserving link privacy in social network based systems," in *NDSS*, 2013.

[25] R. B. Zadeh, A. Goel, K. Munagala, and A. Sharma, "On the precision of social and information networks," in *COSN*, 2013.

[26] A. May, A. Chaintreau, N. Korula, and S. Lattanzi, "Filter & follow: How social media foster content curation," in *SIGMETRICS*, 2014.

[27] C. Gkantsidis, G. Goel, M. Mihail, and A. Saberi, "Towards topology aware networks," in *INFOCOM*, 2007.

[28] J. K. Johnson and M. Chertkov, "A majorization-minimization approach to design of power transmission networks," in *CDC*, 2010.