

On the Efficiency of Social Recommender Networks

Felix Ming Fai Wong, *Student Member, IEEE*, Zhenming Liu, Mung Chiang, *Fellow, IEEE*

Abstract—We study a fundamental question that arises in social recommender systems: whether it is possible to simultaneously maximize (a) an individual’s benefit from using a social network and (b) the efficiency of the network in disseminating information. To tackle this question, our study consists 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 macroscopic level. We also propose a set of metrics for quantifying both user experience and 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 and news feed curation algorithm that can simultaneously address individuals’ need to connect to high quality friends, and service providers’ need to maximize network efficiency in information propagation.

Index Terms—Social networks, recommender systems, stochastic modeling, effective resistance, convex optimization, measurement.

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 news article from CNN, or a game from PS Plus), users are more likely to make a purchase and/or re-visit the recommender system’s website, which can result in a substantial increase in site traffic and revenue.

A recent trend in the design of recommender systems is the introduction of “social” features. For example, Yelp personalizes restaurant rankings by taking into account the opinions of a user’s friends. Netflix shows a user what her friends have recently watched. Amazon allows users to share their purchases on Facebook and/or Twitter. These features have benefits that are apparent: it is usually more fun to interact with friends, and hopefully, friends will be able to influence each other to make purchases, or visit restaurants, which will increase a website’s revenue. In addition, social features can impact the entire recommender system in ways that are less visible, but perhaps even more important. Specifically, (a) at the individual level, friends are more trusted and tend to share

common interests. Thus, individuals are more likely to adopt their friends’ recommendations [1, 2]. (b) At the macroscopic level, recommendations are more likely to propagate through social ties. This word-of-mouth effect plays an important role in promoting less-known but otherwise excellent businesses or products [3, 4].

In other words, the social network in a recommender system plays at least two important roles: (a) it helps individuals improve their experience through interaction with their friends. Let us refer to the resultant benefit as the *local utility* of the social network. (b) It also helps the recommender system in disseminating information or recommendations more efficiently. Let us refer to the resultant benefit as the *global utility* of the social network. Here, an important question arises:

For the social network in a recommender system, are its local and global utilities 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 propagating 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 social networks are usually formed in a decentralized manner [5]: two people can establish a link so long as both parties agree to do so. They often do not have knowledge of the global properties of the network. They make a connection usually not for the sole purpose of sharing opinions and recommendations. It is also not always clear to an individual whether a potential new friend will have high-quality recommendations. Thus, it would be quite surprising to find this crude network formation mechanism to be able to optimize local and/or global utilities.

Our work and contributions. This paper presents an extensive study of the local-global utility tradeoff in a recommender system’s social network, or a *social recommender network*, focusing on a recently-released dataset from Yelp. Our contributions can be summarized as follows.

1. *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 recommender network. Intuitively, one’s local utility measures the discrepancy between her ratings and her friends’ ratings. When the discrepancy is small, friendship-based recommendation will be more reliable, which results in higher local utility (see Eq. (1)). Meanwhile, a network’s global utility measures how fast a recommendation travels. Here, average hitting time is used to measure global utility (see Eq. (2)).

The set of metrics we propose are fairly robust, and work well even when the actual recommendation diffusion model is

Felix M.F. Wong and Mung Chiang are with the Department of Electrical Engineering, Princeton University (emails: {mwthree, chiangm}@princeton.edu).

Zhenming Liu is with the Departments of Computer Science and Electrical Engineering, Princeton University (email: zhenming@cs.princeton.edu).

A preliminary version of this paper was presented at IEEE INFOCOM 2015.

misspecified.

2. *Empirical study of a Yelp dataset.* We next study, based on the proposed metrics, the tradeoff between local and global utilities with data from Yelp, a major rating and review website for local businesses. Our discoveries are as follows.

- When a user has only one friend, it is unclear how much the user benefits from her social ties, but when the user has more than one friend, we see improvements in the quality of recommendations from her friends.
- The global utility of Yelp’s social network is also quite good. It is comparable to that of a random graph, which is known to be efficient at propagating information. On the other hand, the tradeoff between global and local utilities is not optimized, so we need the third part of our study, as discussed below.

3. *Algorithm for social recommender network optimization.* We also design an algorithm for finding a social network topology that simultaneously addresses individuals’ need to establish links to high-quality friends, and service providers’ need to maximize network efficiency in information propagation. Recommender systems and social networks are extensively studied areas with numerous works in improving a recommender system’s user experience (see [6, 7] and the references therein) and maximizing network efficiency through topology design (e.g., [8–11]), but to the best of our knowledge, there exists no prior work that simultaneously addresses both objectives. We believe this is a promising direction for recommender systems research.

Why study Yelp. We have two reasons to focus on the Yelp dataset. First, Yelp is the only large online service, of which we are aware, to make both recent¹ rating and social network data publicly accessible.² Second, unlike other providers such as Amazon and Netflix, Yelp does not directly use Facebook’s API or social network, but instead builds its own in-house social networking service. As a result, its users know they will interact with their Yelp friends only on the Yelp platform, and there are 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 of the diffusion of recommendations and reviews in a social network. Based on this model, we define both an individual’s utility and a network’s efficiency in an intuitive way. In the next section we will analyze this model and present a number of key results in computing the proposed 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 are missing.

Let us denote the social network as $G = \{V, E\}$, where $V = \{v_1, v_2, \dots, v_n\}$. Let $B = \{b_1, b_2, \dots, b_\ell\}$ 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 $W = [W_{i,j}] \in$

$\mathbb{R}_+^{n \times n}$. Intuitively, the number $W_{i,j}$ represents the strength of the social tie between v_i and v_j , i.e., if v_i and v_j are close friends, then $W_{i,j}$ should be large, and vice versa. We also require $W_{i,j} = W_{j,i}$, and $W_{i,j} = 0$ 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 imagine the value of $W_{i,j}$ to be chosen by v_i and v_j , but not anyone else.

Our discrete-time process works as follows: at time $t = 0$, each business $b \in B$ is visited by a user, call it $p_0(b) \in V$. After user $p_0(b)$ visits b , she forms an opinion regarding b (i.e., writes a review), call it R_b^0 . Here, R_b^* is a real number and is large when the user has high opinion of b (in Yelp this can be taken as the rating score, which is an integer between 1 and 5). In each of the subsequent times $t \geq 1$ and for each b , one neighbor of $p_{t-1}(b)$ visits the business, and the probability that some v is this visiting neighbor, i.e., $p_t(b) = v$, is proportional to $W_{p_{t-1}(b),v}$. After the user visits the business, she also generates a review R_b^t . 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 to remain unchanged after a revisit.

Based on this process, we now are able to define both the local and global utilities of the social network.

Local utility. We make use of the following intuition to define an individual’s local utility. When a user decides to visit a business because of her friend’s influence, she should consider her friend’s review useful only if it is consistent with her own opinion. Thus, we use the discrepancy between R_b^{t-1} and R_b^t 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 businesses that v visits at time t . Note that $e_t(v)$ could be an empty set, in the case of v not visiting any businesses at time t . For arbitrary $b \in e_t(v)$, denote 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 define user v ’s local cost as

$$\lim_{t \rightarrow \infty} \mathbb{E} \left[\frac{1}{|e_t(v)|} \sum_{b \in e_t(v)} |R_b^t - R_b^{t-1}| \mid e_t(v) \neq \emptyset \right]. \quad (1)$$

With slight abuse in word usage, we also refer to (1) as v ’s local utility, the lower the better.

We make a few remarks here. First, as we are interested in only the times at which v visits a business, we focus on the conditional expectation $\mathbb{E}[\cdot \mid e_t(v) \neq \emptyset]$. Second, we use $|R_b^t - R_b^{t-1}|$ to quantify the discrepancy between v ’s opinion and her friend’s opinion. There are other choices but we use the ℓ_1 -norm because it is more robust against outliers. Also, we need to average out the discrepancies in the case of $|e_t(v)| > 1$ for normalization purposes. Finally, we take $t \rightarrow \infty$ as we are more interested in the system’s equilibrium behavior.

Global utility. Intuitively, the global utility function should be able to measure how fast information propagates. Under our recommendation diffusion model, we take it as the average waiting time for each user to review a specific business. More specifically, let $v_i = p_0(b)$ be the first person that reviews business b , and let $T(v_j, b) = \min\{t : p_t(b) = v_j\}$.

¹Including data from 2014.

²Without the need of scraping and crawling.

The average waiting time can be defined as the mean of $T(v_j, b)$, over all v_j , *i.e.*, $\frac{1}{n}\mathbb{E}[\sum_j T(v_j, b)]$. Furthermore, we also require the average waiting time to be uniformly good, regardless the starting point $p_0(b)$. Thus, our final objective is

$$\mathbb{E}_{p_0(b) \sim U^V} \left\{ \frac{1}{n} \mathbb{E} \left[\sum_j T(v_j, b) \right] \right\}, \quad (2)$$

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 MISSPECIFICATION

A. Local utility

Efficient computation. We now describe the procedure for estimating an individual's utility, starting with the assumption that there are no missing data. As all users will rate all businesses as $t \rightarrow \infty$, we 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

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

Recall ℓ is the number of businesses in the social recommender network.

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

$$D_i = \frac{1}{\ell} \sum_k \frac{\sum_j W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}}. \quad (3)$$

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

$$\bar{D} = \frac{1}{n} \sum_{i=1}^n D_i. \quad (4)$$

Missing data. We consider an approximation to Eq. (3) to account for the missing data problem in real datasets, *i.e.*, there are many (j, k) pairs such that $R_{j,k}$ is not available in the dataset. When $R_{j,k}$ is missing, we 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:

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

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

where $\mathcal{R}_i = \{k \mid R_{i,k} \text{ is available}\}$, 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 misspecified, 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 you need a recommendation on a specific business but none of your friends have ever rated it, the social network should not harm you, *i.e.*, the recommender system platform should return the average rating of that business.

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 \bar{H} as

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

which is related to the eigenvalues of the unnormalized graph Laplacian: $L = \text{diag}(W\mathbf{1}) - W$. Let the sorted eigenvalues of L be $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. It can be proved [9] that \bar{H} and $\{\lambda_i\}$ are related as

$$\bar{H} = \frac{2 \sum_{i < j} W_{i,j}}{n(n-1)} R_{\text{tot}} \quad (8)$$

$$R_{\text{tot}} = n \sum_{i=2}^n \frac{1}{\lambda_i}, \quad (9)$$

where R_{tot} is known as the effective resistance of W . Note that smaller \bar{H} means better global utility.

Robustness. Let us start by considering the 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 [10–13] for examples). We observe that the efficiency in information propagation from each of the models is invariably characterized by one of these closely related statistics: (a) the second largest eigenvalue of the Laplacian or transition matrix [14], (b) the conductance of the network [10], (c) mixing time, and (d) hitting time, which is used for our model. All of 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 of its mixing time, conductance and second largest eigenvalue. Thus, they are all roughly equivalent, *i.e.*, when a social network is

optimized for average hitting time, it is also asymptotically optimal in almost all other information diffusion models (see [14] for more discussions), and average hitting time is a fairly robust statistic.

We also remark that the second largest eigenvalue and mixing time are less appropriate statistics in our setting. Even in our simple model, mixing time or the second largest eigenvalue alone is insufficient to bound average hitting time. Knowledge of the stationary distribution is also needed (see Fact 3.1), but by Occam’s razor principle, we prefer a simpler model with fewer parameters (*i.e.*, not specifying the stationary distribution), and there is also no natural way to interpret the stationary distribution 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 ∞ as $\tau \rightarrow \infty$.

C. Generality of local and global utilities

We also remark that both the local and global utilities derived from our model resemble metrics that are routinely used to assess the efficacy of recommender systems and information networks. Specifically, Eq. (3) is equivalent to weighted ℓ_1 regression or the mean absolute error, which is often used to measure the quality of a recommender system [7], and spectral properties (including the effective resistance used here) measures a network’s efficiency in propagating information [10–14]. This means: (a) our stochastic model establishes an explicit (and unexpected) connection between commonly used metrics in recommender systems and information networks, and (b) the definitions of local and global utilities are agnostic about the specific information propagation model one uses, so understanding their tradeoff is important despite a number of possibly unrealistic assumptions made in our specific model.

IV. DATASET AND ASSOCIATED GRAPHS

The social recommender network we study is the Yelp social network from Phoenix, Arizona.³ The dataset contains 335,022 reviews made by 70,817 users on 15,585 businesses, and the users’ social network with 151,516 edges.

Our analysis focuses on two graphs. The first is the provided social graph (network), which is constructed by users in a decentralized but strategic manner. It is the focus of our empirical study for understanding the tradeoff between local and global utilities. The second graph is constructed from users’ ratings. It represents how users “organically” interact with each other, either directly or indirectly in Yelp, when no intervention is presented. For instance, if two users have co-rated a large number of businesses, we consider them to have strong interaction. This graph serves as a baseline for comparison with the social graph. 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 third party API to

manage the network), although users may import their friends from email or Facebook accounts. Following related work in online social network (OSN) analysis [15], we focus on the largest connected component of the social graph. 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 four users. We note that our analysis is meaningful only for the users who care about their social network. To allow for a direct comparison with the co-rating graph (see details below), we further remove 330 users that do not have sufficient reviewing activity and are isolated in the co-rating graph. The result is a connected social graph with 28,647 users and 149,348 edges, covering 94.7% of the users in the dataset who have at least one friend.

Co-rating graph. The co-rating graph is constructed such that two users share an edge if and only if they have co-rated at least one business. This graph can be interpreted as the largest network that can be discovered by users reading the reviews of other users. Since our metrics are based on reviews and ratings, we also expect a “good” network (*i.e.*, a network that has good local and global utilities) to be embedded in this graph with a suitable choice of edge weights and topology (*i.e.*, pruning useless edges). Focusing on only the 28,647 users seen in the social graph, this graph has 8.13 million edges, 54 times more edges than the social graph.

Figure 1 shows the basic properties of the Yelp data in terms of making friends and reviewing businesses, the two major forms of user activity. The degree distribution exhibits higher skew than review activity distribution, which is consistent with the intuition that reviewing a business (writing and typing in text) requires more effort than adding a friend (clicking a “friend request” button). Nevertheless, the two forms of user activity are positively correlated, especially towards the tail (see Figure 1(c)).

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

Perturbing network topology. We also compare the social graph with a random graph baseline. There are three reasons for doing so. (a) 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.” (b) Random graphs are known to have small mixing time, small conductance, and small second largest eigenvalues [16], *i.e.*, they are efficient in information propagation. We consider a network efficient if its global utility is comparable to that of random graphs. (c) In terms of local utility, random graphs correspond to the case where users rely on random peers, rather than friends, for opinions. If the Yelp social network has good local utility, *i.e.*, there is positive correlation between friends’ ratings, the social graph should perform better than the random graph baselines.

Starting from the social graph, we shuffle its edges to obtain a edge-perturbed graph. Algorithm 1, similar to that presented in [17], ensures the resultant graph is connected (by checking for connectedness after every window steps, and if not, reverting to the previous snapshot G') and the degree of each

³http://www.yelp.com/dataset_challenge. We use the data released for the third round of the Yelp Dataset Challenge.

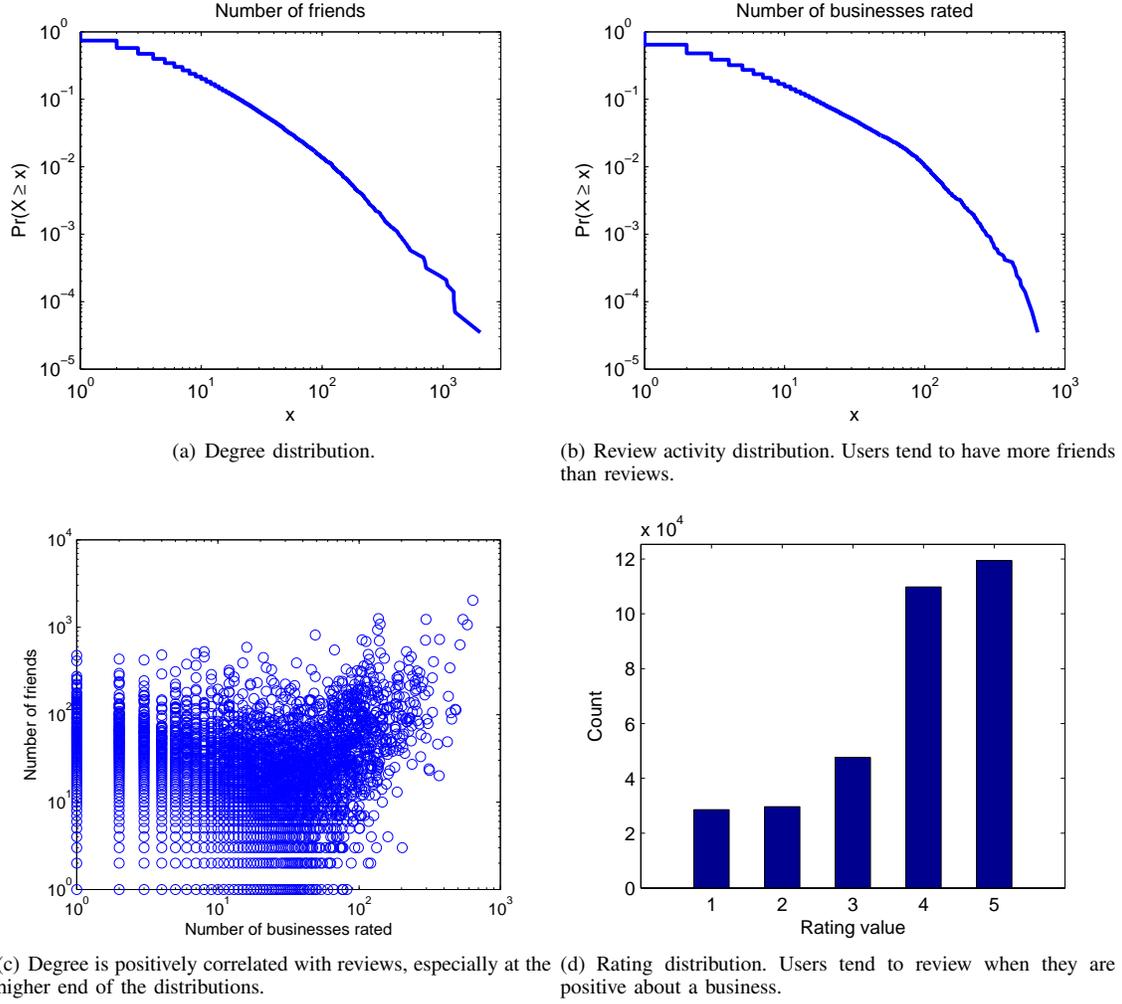


Fig. 1: Basic data statistics.

node is preserved (since each shuffle is degree-preserving). By running Algorithm 1 with increasing `num_iter`, we obtain a family of graphs with increasing distance (measured by the number of differing edges) from the original social graph, and in the limit $\text{num_iter} \rightarrow \infty$, 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: $G = \{V, E\}, \text{num_iter}, \text{window}$

```

for  $i = 1$  to  $\text{num\_iter}$  do
   $G' \leftarrow G$ 
  repeat
     $G \leftarrow G'$ 
    for  $j = 1$  to  $\text{window}$  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 the following *nonnegative*⁴ edge weighting strategies based on rating similarity.

- Jaccard coefficient on rating sets:

$$W_{i,j} = \frac{|\mathcal{R}_i \cap \mathcal{R}_j|}{|\mathcal{R}_i \cup \mathcal{R}_j|}.$$

This captures the likelihood of a pair of users interacting through reading and writing reviews, but is agnostic to the values of the ratings.

- Pearson correlation coefficient on rating vectors:

$$W_{i,j} = 1 + \rho_{i,j}$$

$$\rho_{i,j} = \frac{\sum_{k \in \mathcal{R}_{i,j}} (R_{i,k} - r_i)(R_{j,k} - r_j)}{\sqrt{\sum_{k \in \mathcal{R}_{i,j}} (R_{i,k} - r_i)^2} \sqrt{\sum_{k \in \mathcal{R}_{i,j}} (R_{j,k} - r_j)^2}},$$

⁴Many user similarity measures in the recommender systems literature can take negative values. While leveraging negative correlation, *i.e.*, taking the opposite opinion of a dissimilar peer, can help improve rating prediction, it would be unrealistic to expect a person to make friends with those with whom they strongly disagree.

where $\mathcal{R}_{i,j} = \mathcal{R}_i \cap \mathcal{R}_j$, $r_i = \sum_{k \in \mathcal{R}_i} R_{i,k} / |\mathcal{R}_i|$, and the 1 is added to make $W_{i,j} \in [0, 2]$.⁵

- Cosine similarity on rating vectors:

$$W_{i,j} = \frac{\sum_{k \in \mathcal{R}_{i,j}} R_{i,k} R_{j,k}}{\sqrt{\sum_{k \in \mathcal{R}_{i,j}} R_{i,k}^2} \sqrt{\sum_{k \in \mathcal{R}_{i,j}} R_{j,k}^2}},$$

defined such that $W_{i,j} \in [0, 1]$. Note that $W_{i,j} \geq 0$ because rating values are positive.

V. EMPIRICAL DATA ANALYSIS

A. Comparing Graphs

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

Comparing global utility. The Yelp social graph has better global utility \bar{H} than all other graphs, except for the random graph baseline.⁶ This is expected because random graphs have better efficiency properties (as discussed in Section IV), but the difference is only moderate (only 12% degradation), and we believe most of the edges in random graphs are not discoverable by users. Also, the slight loss in global utility is compensated by an improvement in local utility \bar{D} .⁷

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 \bar{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 too many edges), reminiscent of findings from social navigation studies [18]. Perhaps more surprisingly, even when we use the other three rating similarity measures to weigh edges, \bar{H} does not improve. This implies finding good edge weights is not straightforward.

Comparing local utility. Again, the social graph performs better than all other graphs under the \bar{D} metric. This confirms that users are conscious of local utility when forming the social network. Comparing the three rating 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 \bar{D} .

We also build a “hybrid” graph, where the graph is induced by the social graph but the edge weights are derived from the three rating similarity measures. Surprisingly, only about

⁵Alternatively, we can set $W_{i,j} = \max\{0, \rho_{i,j}\}$, *i.e.*, clipping the Pearson correlation coefficient above 0, to obtain nonnegative edge weights. The resultant weighted graph has essentially the same local utility $\bar{D} = 0.954$ (compare with Table I) but is disconnected (meaning $\bar{H} = \infty$).

⁶Generated by applying Algorithm 1 on the social graph with `num_iter = window = 1000`, so as to perform 1 million edge shuffles.

⁷A statistically principled way to compare two graphs’ local utilities is to apply a Wilcoxon signed rank test on their D_i vectors. We find the results to be consistent with our simple \bar{D} comparison, *i.e.*, when one graph’s \bar{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.

TABLE II: Local utility of disconnected networks. Here we also compare the networks with the social graph. Combining the social graph with non-uniform edge weighting results in better local utility.

Network	\bar{D}	Difference in \bar{D}	t-statistic
Yelp average (no network)	0.887	-0.006 ± 0.001	-8.48
Social, Jaccard	0.875	0.007 ± 0.001	10.3
Social, Pearson	0.872	0.009 ± 0.001	7.39
Social, Cosine	0.876	0.006 ± 0.001	7.30
Social, Jaccard \circ Pearson	0.872	0.010 ± 0.001	10.2
Social, Jaccard \circ Cosine	0.875	0.007 ± 0.001	6.68

29% of the edges in the social graph also appear 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 \bar{D} (note $\bar{H} = \infty$ in this case), and Table II shows the results for different graph-weight combinations. Our baseline \bar{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 achieves better \bar{D} than all \bar{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. Weighting edges of the social graph by Jaccard/Cosine/Pearson coefficients further improve \bar{D} . In Section V-C we consider a modification to weight social graph edges that preserve graph connectedness.

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 made evident in Figure 2(d), a scatterplot showing strong negative correlation between the two metrics.

Empirical results [17] 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 at around 200,000 shuffles, we are convinced that any graph generated with more than 200,000 shuffles are close to random. Figure 2(c) further supports our claim by showing the proportion of edges that overlap with those of the social graph to be small (about 4%).

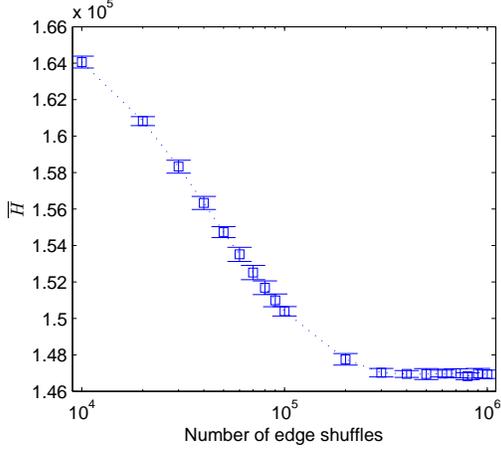
C. Varying Weight Skew

Here we consider an edge weighting strategy to study the tradeoff between \bar{H} and \bar{D} . From Tables I and II, it appears that weighting the social graph with Pearson coefficient will result in the best possible \bar{H} and \bar{D} , so we apply the weight transformation parametrized by $\alpha, \beta > 0$:

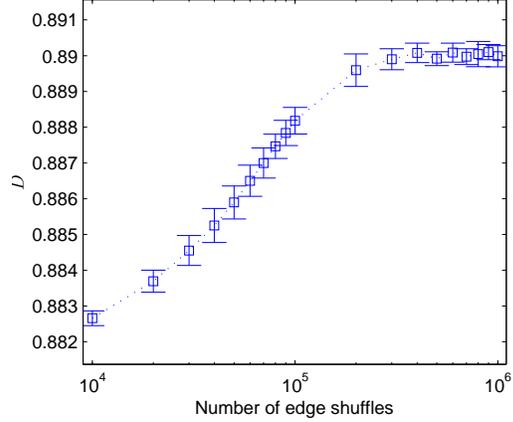
$$W_{i,j} = \begin{cases} (W_{i,j}^{\text{Pearson}} + \beta)^\alpha & W_{i,j}^{\text{social}} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

TABLE I: Network metrics on connected networks. Results of “random” are due to 10 random graph instances summarized by the mean and standard error of \bar{H} and \bar{D} . We also report the change in each user’s D_i when switching to the social graph (the smaller the better) with the mean, standard error and t-statistic. The social graph has better local utility than all other graphs, and its global utility is only slightly worse than that of the random graph baseline.

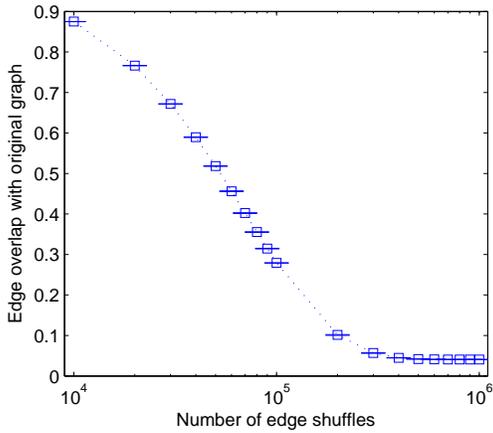
Network	\bar{H}	\bar{D}	Difference in $\{D_i\}$	t-statistic
Social, uniform	1.68×10^5	0.882	—	—
Random, uniform (10 trials)	$1.47 \times 10^5 \pm 72.7$	$0.890 \pm 9.31 \times 10^{-5}$	-0.008 ± 0.001	-12.6
Co-rating, uniform	7.28×10^5	0.994	-0.112 ± 0.002	-71.0
Co-rating, Jaccard	1.03×10^6	0.993	-0.111 ± 0.002	-54.3
Co-rating, Pearson	4.23×10^6	0.955	-0.073 ± 0.002	-43.6
Co-rating, Cosine	7.30×10^5	0.994	-0.112 ± 0.002	-70.8
Co-rating, Jaccard \circ Pearson	6.15×10^6	0.961	-0.079 ± 0.002	-37.9
Co-rating, Jaccard \circ Cosine	1.03×10^6	0.992	-0.111 ± 0.002	-54.2



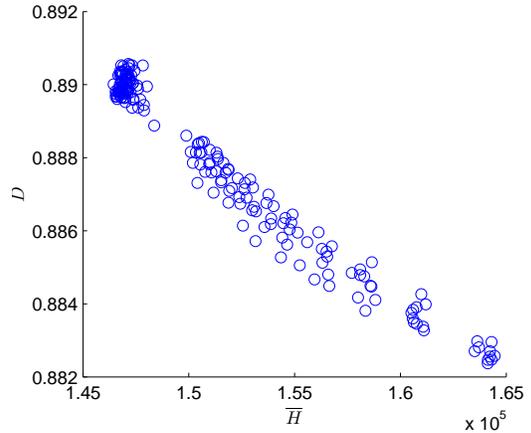
(a) Increasing the number of edge shuffles improves global utility as graphs become closer to random.



(b) Increasing the number of edge shuffles degrades local utility as users’ original friend preferences are forgotten.



(c) Similarity with original graph, *i.e.*, number of edges that remain after shuffling, decreases with more shuffling.



(d) Negative correlation between local and global utilities. Each datapoint is a generated graph instance.

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

Note that $W_{i,j}^{\text{Pearson}} = 0$ means users v_i and v_j have not co-rated anything, and setting $\beta > 0$ ensures the graph is connected. Weight skew, *i.e.*, the difference between the maximum and minimum nonzero edge weights, decreases with decreasing α or increasing β . In the limits $\alpha = 0$ or $\beta \rightarrow \infty$, the graph is equivalent to the social graph with uniform weights, *i.e.*, with zero weight skew.

Figures 3(a) and 3(b) show the tradeoff between \bar{H} and \bar{D} by varying weight skew through (α, β) : for (α, β) pairs that result in good \bar{H} (blue areas), the corresponding \bar{D} is bad (red areas), and vice versa. Figure 3(c) further illustrates the tradeoff with a scatterplot.

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 have their D_i between 0 and 1, but there is also a considerable portion of outliers coming from the population with few friends (Figure 4(b)). While local utility improves as a user builds 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.

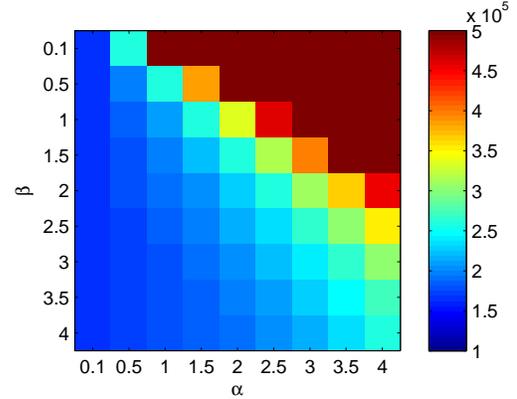
To eliminate the effect of confounding, we carry out statistical tests, enabled by our local utility metric, on populations with similar activity levels. First, we run a signed rank test on the two $\{D_i\}$ vectors of the full population due to the social graph and Yelp average. Although the social graph has lower \bar{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: 6.07×10^{-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

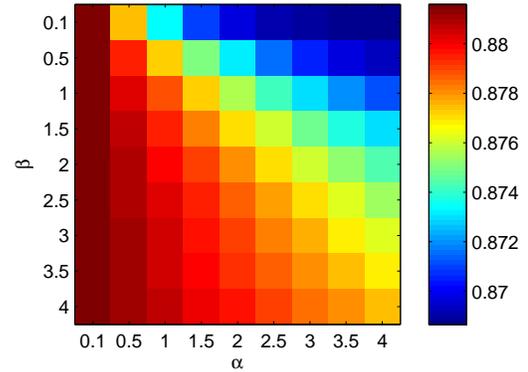
A. Optimization Problem

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

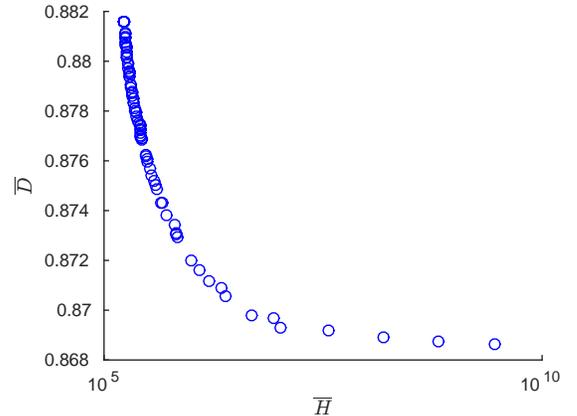
Consider the scenario where each user v_i has a constraint on her local utility $D_i \leq \delta_i$, where δ_i is the worst tolerable utility as observed from existing data, *e.g.*, D_i measured on the given social graph. In addition, we are given an allowable edge set E with $|E| = m$. Then the social recommender network



(a) Less weight skew, *i.e.*, decreasing α or increasing β , improves global utility (blue area).



(b) More weight skew, *i.e.*, increasing α or decreasing β , improves local utility (blue area).



(c) Negative correlation between local and global utilities. Each datapoint is a weighted graph corresponding to an (α, β) pair.

Fig. 3: Effect of skewing edge weights by controlling (α, β) parameters.

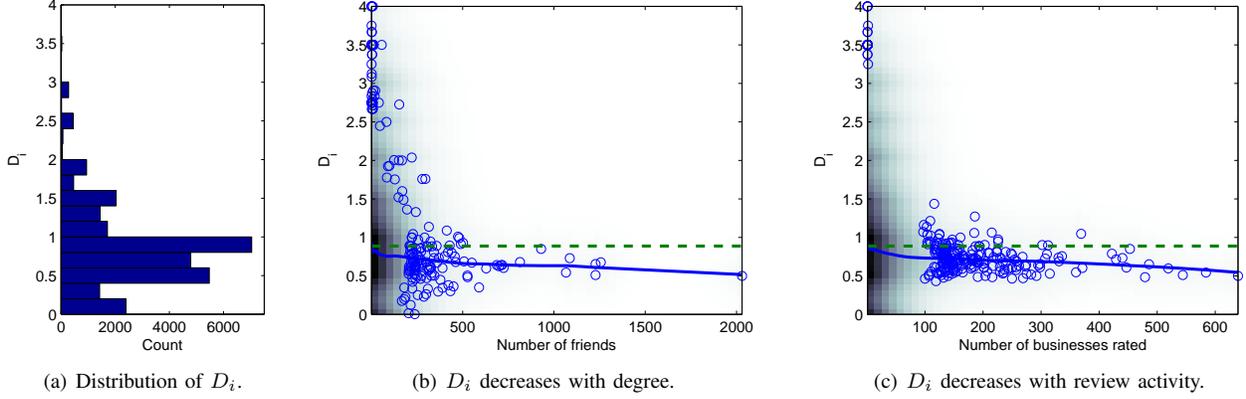


Fig. 4: Distribution of D_i on Yelp social graph. For (b) and (c), 2D histograms are shown with 1% outliers overlaid. Dashed lines are the \bar{D} baseline due to Yelp average, and solid lines are locally weighted linear regression fit of data.

optimization problem becomes maximizing network efficiency, subject to topology and local utility constraints:

$$\begin{aligned}
 & \underset{W \in \mathbb{R}^{n \times n}}{\text{minimize}} && \bar{H}(W) \\
 & \text{subject to} && D_i(W) \leq \delta_i \quad i = 1, 2, \dots, n \\
 & && W_{i,j} = 0 \quad \text{for } \{i, j\} \notin E \\
 & && W = W^T, W \geq 0.
 \end{aligned} \tag{10}$$

Since both \bar{H} and $\{D_i\}$ are invariant to scaling, *i.e.*, W and αW (for any $\alpha > 0$) result in the same \bar{H} and $\{D_i\}$, problem (10) can be rewritten in terms of effective resistance (see Section III-B for a discussion):

$$\begin{aligned}
 & \underset{W \in \mathbb{R}^{n \times n}}{\text{minimize}} && R_{\text{tot}}(W) \\
 & \text{subject to} && D_i(W) \leq \delta_i \quad i = 1, 2, \dots, n \\
 & && W_{i,j} = 0 \quad \text{for } \{i, j\} \notin E \\
 & && W = W^T, W \geq 0 \\
 & && \sum_{i < j} W_{i,j} = m,
 \end{aligned} \tag{11}$$

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

Lemma 6.1: Optimization problem (11) is convex.

Proof: We need to verify the convexity of the objective function and the constraints. R_{tot} has been shown to be convex in [9], and the constraints $W = W^T$, $W \geq 0$ and $\sum_{i < j} W_{i,j} = m$ 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

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

which is another affine constraint. \blacksquare

B. Algorithm

Our problem of finding optimal edge weights is solved on networks with at least 100,000 edges. Off-the-shelf tools based on interior point and semidefinite programming methods are

prohibitively expensive on problems of this size. Thus we opt for a projected gradient method instead. In the following, it is more convenient to express the variables as the vector $w = \{w_l\}$, for all $l \in E$. Let $E_i = \{l \mid l = \{i, j\} \text{ any } j\}$, then we rewrite problem (11) as

$$\begin{aligned}
 & \underset{w \in \mathbb{R}^m}{\text{minimize}} && R_{\text{tot}}(w) \\
 & \text{subject to} && w \geq 0, \mathbf{1}^T w = m
 \end{aligned}$$

$$\sum_{l=\{i,j\} \in E_i} w_l \left(\sum_{k \in \mathcal{R}_i} |\tilde{R}_{j,k} - R_{i,k}| - \delta_i |\mathcal{R}_i| \right) \leq 0. \tag{12}$$

Deriving the optimization algorithm involves two steps.

Deriving the gradient. From [9] we have for $l = \{i, j\}$,

$$\begin{aligned}
 \frac{\partial R_{\text{tot}}}{\partial w_l} &= -n \left\| \left(L + \frac{1}{n} \mathbf{1}\mathbf{1}^T \right)^{-1} a_l \right\|_2^2 \\
 &= -n \left\| L^\dagger a_l + \frac{1}{n} \mathbf{1}\mathbf{1}^T a_l \right\|_2^2 \\
 &= -n \left\| L^\dagger (e_i - e_j) + \frac{1}{n} (e_i - e_j) \right\|_2^2,
 \end{aligned} \tag{13}$$

where L^\dagger is the pseudoinverse of L , a_l is the vector with $(a_l)_i = 1$, $(a_l)_j = -1$, $(a_l)_k = 0$ 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 [9]). We use the LAMG solver [19] to compute the vectors $L^\dagger e_i$.

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 $\{w_l\}$ 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

$$\begin{aligned}
 & \underset{x \in \mathbb{R}^m}{\text{minimize}} && \|x - w\|_2^2 \\
 & \text{subject to} && x \geq 0, \mathbf{1}^T x = m \\
 & && b_i^T x_i \leq 0 \quad i = 1, \dots, n,
 \end{aligned} \tag{14}$$

where x_i is defined with the same indexing scheme as w_i . We use Gurobi [20] to solve this quadratic program. Algorithm 2 summarizes our solution approach to problem (12).

Computational cost. The cost of a gradient step is dominated by computing the vector $L^\dagger(e_i - e_j)$. An efficient Laplacian solver [19] can compute $L^\dagger x$ in $\tilde{O}(m)$ for any $x \in \mathbb{R}^n$, so we can naively compute ∇R_{tot} in $\tilde{O}(m^2)$ time: for every $(i, j) = l \in E$, directly compute $L^\dagger(e_i - e_j)$ to obtain $(\nabla R_{\text{tot}})_l$. However, a speedup is possible by noting there are only n such e_i 's. We compute $L^\dagger(e_i - e_j)$ as $L^\dagger e_i - L^\dagger e_j$, with $L^\dagger e_i$ precomputed for $i = 1, \dots, n$. The precomputation takes $\tilde{O}(mn)$ time, and computing each $(\nabla R_{\text{tot}})_l$ takes $O(n)$ time since it involves summing n -dimensional vectors. Thus computing ∇R_{tot} takes $\tilde{O}(mn) + O(mn) = \tilde{O}(mn)$ time.

On the other hand, we cannot estimate the time complexity of the projection step because Gurobi [20] and other commercial solvers use proprietary heuristics. General bounds on quadratic programs based on interior point methods appear to be too conservative, as in practice we find the projection step to take negligible time (less than 1 second, compared to more than 10 minutes for a gradient step).

Finally, our optimized graph is computed with about 130 iterations, or roughly 24 hours using the hardware described below.

Algorithm 2 Projected gradient descent to solve (12).

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 ∇R_{tot} as calculated by (13)
 $w \leftarrow x^*$, where x^* is solution to problem (14)
until w converges

C. Results

We implement our algorithm on a machine with two Intel E5540 processors and 16GB RAM. Then we input E as the 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 allowable edge set E . Compared to the graph with uniform edge weights, we see substantial improvement in \overline{H} , *i.e.*, the optimal graph achieves $\overline{H} = 5.78 \times 10^4$, a 65% reduction in \overline{H} , and slightly better $\overline{D} = 0.879$. The hitting time of this optimized graph is also substantially better than that of random graphs (with uniform edge weights).

Statistics of optimized graph. We are interested in what the optimized graph looks like. First, we check whether the node degrees still follow a power law distribution. Figure 5(a) shows the log-log plot of the degree distribution. We notice that the key difference between the optimized graph and a standard social network is that there are very few nodes with degree ≤ 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 ≥ 5 have their degrees 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 transi-

tion 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.

Correlation between original and optimized graph. Next, we study the correlation between the original and the optimized graph. Figure 5(c) shows the degree correlation. While the node degrees of both graphs are correlated in general, many nodes have substantial changes in weights. This shows global surgery is needed to optimize the original graph. 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 there are 28,647 nodes), the Jaccard coefficients are consistently below 0.5. This implies the overlap between the ‘‘important’’ nodes of the two graphs is not substantial.

VII. RELATED WORK

Our global utility metric based on average hitting time is related to proposals of using effective resistance [21] and related measures [22, 23] 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 [15] and information diffusion [24], but some recent works have also addressed local utility. Mittal et al. [25] proposed a random walk-based metric to study the tradeoff between privacy and local utility in the context of privacy-preserving graph perturbation. Zadeh et al. [26] showed high precision and fast dissemination can simultaneously be achieved in some broadcast networks. May et al. [27] studied content curation in social media at a microscopic level. Ma [28] studied the correlation between social relationships and user interest similarity. 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 [8] or effective resistance [9] have found applications in, *e.g.*, computer networks [29] and power networks [30]. 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 ($>100,000$ edges) is much larger than those seen in existing work ($\sim 1,000$ 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 of a social recommender network. We compute 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

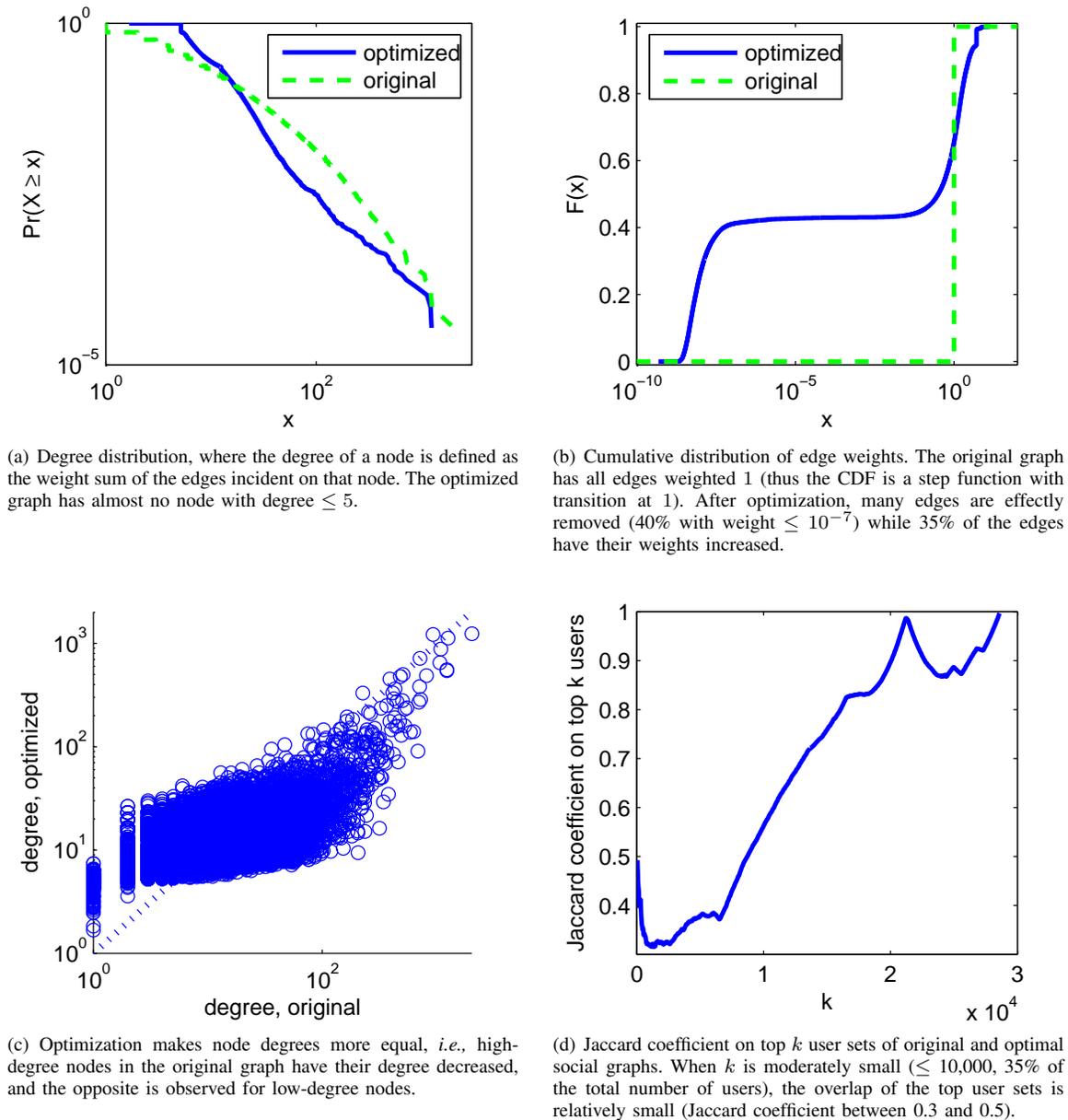


Fig. 5: Results from optimizing edge weights of Yelp social graph.

statistically significant, and (b) at the macroscopic 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 examine how we can construct social networks that have local and global utilities jointly optimized. While our optimization problem is convex, it is a large-scale problem that prompts for a hybrid solution blending in-house developed algorithmic techniques 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.

IX. ACKNOWLEDGMENTS

This work was in part supported by ARO grants W911NF-11-1-0036 and W911NF-14-1-0190.

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. Rincon, 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] F. M. F. Wong and P. Marbach, “‘Who are your friends?’—A simple mechanism that achieves perfect network formation,” in *INFOCOM*, 2011.

- [6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, 2009.
- [7] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009.
- [8] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM Review*, vol. 46, no. 4, 2004.
- [9] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Review*, vol. 50, no. 1, 2008.
- [10] F. Chierichetti, S. Lattanzi, and A. Panconesi, "Almost tight bounds for rumour spreading with conductance," in *STOC*, 2010.
- [11] M. Abdullah, C. Cooper, and M. Draief, "Viral processes by random walks on random regular graphs," in *APPROX-RANDOM*, 2011.
- [12] H. Lam, Z. Liu, M. Mitzenmacher, X. Sun, and Y. Wang, "Information dissemination via random walks in d-dimensional space," in *SODA*, 2012.
- [13] A. Clementi, R. Silvestri, and L. Trevisan, "Information spreading in dynamic graphs," in *PODC*, 2012.
- [14] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.
- [15] A. Mohaisen, A. Yun, and Y. Kim, "Measuring the mixing time of social graphs," in *IMC*, 2010.
- [16] B. Bollobás, *Random graphs*. Academic Press, 1998.
- [17] F. Viger and M. Latapy, "Fast generation of random connected graphs with prescribed degrees," in *COCOON*, 2005.
- [18] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *STOC*, 2000.
- [19] O. E. Livne and A. Brandt, "Lean algebraic multigrid (LAMG): Fast graph laplacian linear solver," *SIAM Journal on Scientific Computing*, vol. 34, no. 4, 2012.
- [20] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2014. [Online]. Available: <http://www.gurobi.com>
- [21] 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.
- [22] A. Tizghadam and A. Leon-Garcia, "Autonomic traffic engineering for network robustness," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 1, 2010.
- [23] 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.
- [24] 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.
- [25] P. Mittal, C. Papamanthou, and D. Song, "Preserving link privacy in social network based systems," in *NDSS*, 2013.
- [26] R. B. Zadeh, A. Goel, K. Munagala, and A. Sharma, "On the precision of social and information networks," in *COSN*, 2013.
- [27] A. May, A. Chaintreau, N. Korula, and S. Lattanzi, "Filter & follow: How social media foster content curation," in *SIGMETRICS*, 2014.
- [28] H. Ma, "On measuring social friend interest similarities in recommender systems," in *SIGIR*, 2014.
- [29] C. Gkantsidis, G. Goel, M. Mihail, and A. Saberi, "Towards topology aware networks," in *INFOCOM*, 2007.
- [30] J. K. Johnson and M. Chertkov, "A majorization-minimization approach to design of power transmission networks," in *CDC*, 2010.

APPENDIX

Proof of Proposition 3.1 First, notice that for any b , the path $\{p_t(b)\}_{t \geq 0}$ is a random walk over the undirected but weighted graph induced by W . Specifically, the transition probabilities are defined as

$$\Pr[p_t(b) = v_j \mid p_{t-1}(b) = v_i] = \frac{W_{i,j}}{\sum_j W_{i,j}}.$$

As we are interested in the conditional expectation when $t \rightarrow \infty$, we may safely assume that $p_0(b)$ is sampled from the stationary distribution π of the Markov chain. Furthermore, as one can see the Markov chain is time-reversible, we have

$$\Pr[p_t(b) = v_j \wedge p_{t-1}(b) = v_i] = \Pr[p_t(b) = v_j \wedge p_{t+1}(b) = v_i].$$

We need more notation. Let $X_{k,t}(v_i)$ be the indicator random variable that sets to 1 if and only if $p_t(b_k) = v_i$ (i.e.,

v_i visits the business b_k at time t). When v_i is clear from the context, we write $X_{k,t}$ directly. Also, let $C_t = \sum_k X_{k,t}$. We have

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{|e_t(v_i)|} \sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid e_t(v_i) \neq \emptyset \right] \\ &= \sum_{d=1}^{\ell} \mathbb{E} \left[\frac{1}{d} \sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = d, e_t(v_i) \neq \emptyset \right] \\ & \quad \cdot \Pr[C_t = d \mid e_t(v_i) \neq \emptyset]. \end{aligned}$$

Notice that $C_t \geq 1$ implies $e_t(v_i) \neq \emptyset$ and thus $(C_t = d, e_t(v_i) \neq \emptyset)$ is the same as $C_t = d$.

For exposition purposes, let us consider the summands in the above equation for $d = 1$. Then we will derive the equations for general $d \geq 1$. For $d = 1$, we are interested in computing $\Pr[C_t = 1 \mid e_t(v_i) \neq \emptyset]$ and

$$\begin{aligned} & \mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = 1, e_t(v_i) \neq \emptyset \right] \\ &= \mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = 1 \right]. \end{aligned}$$

First, notice that

$$\begin{aligned} \Pr[C_t = 1 \mid e_t(v_i) \neq \emptyset] &= \frac{\Pr[C_t = 1]}{\Pr[e_t(v_i) \neq \emptyset]} \\ &= \frac{\binom{\ell}{1} \pi_i (1 - \pi_i)^{\ell-1}}{1 - (1 - \pi_i)^\ell}, \end{aligned}$$

where π_i is the probability mass at v_i at the stationary distribution vector π .

Let us define $E_{k,t}$ as the event that v_i makes exactly one review at time t and the review is on business b_k , i.e., the event $e_t(v_i) = \{b_k\}$. One can see that $E_{k,t}$ are mutually exclusive on k , and the event $C_t = 1$ is equivalent to $\bigvee_{k \leq \ell} E_{k,t}$. Thus,

$$\begin{aligned} & \mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = 1 \right] \\ &= \sum_{k \leq \ell} \mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = 1, E_{k,t} = 1 \right] \\ & \quad \cdot \Pr[E_{k,t} = 1 \mid C_t = 1] \\ &= \sum_{k \leq \ell} \mathbb{E} [|R_{b_k}^t - R_{b_k}^{t-1}| \mid E_{k,t} = 1] \cdot \Pr[E_{k,t} \mid C_t = 1]. \end{aligned}$$

Now we use the time-reversibility property to compute:

$$\begin{aligned} & \mathbb{E} [|R_{b_k}^t - R_{b_k}^{t-1}| \mid E_{k,t} = 1] \\ &= \sum_j \Pr[p_{t-1}(b_k) = v_j \mid p_t(b_k) = v_i] \cdot |R_{j,k} - R_{i,k}| \\ &= \sum_j \frac{\Pr[p_{t-1}(b_k) = v_j \wedge p_t(b_k) = v_i]}{\Pr[p_t(b_k) = v_i]} \cdot |R_{j,k} - R_{i,k}| \\ &= \sum_j \frac{\Pr[p_{t+1}(b_k) = v_j \wedge p_t(b_k) = v_i]}{\Pr[p_t(b_k) = v_i]} \cdot |R_{j,k} - R_{i,k}| \\ & \quad \text{(Markov chain is time-reversible)} \\ &= \sum_j \Pr[p_{t+1}(b_k) = v_j \mid p_t(b_k) = v_i] \cdot |R_{j,k} - R_{i,k}| \end{aligned}$$

$$= \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}}.$$

Also,

$$\begin{aligned} \Pr[E_{k,t} | C_t = 1] &= \frac{\Pr[E_{k,t}]}{\Pr[C_t = 1]} \\ &= \frac{\pi_i(1 - \pi_i)^{\ell-1}}{\binom{\ell}{1} \pi_i(1 - \pi_i)^{\ell-1}} \\ &= \frac{1}{\binom{\ell}{1}}. \end{aligned}$$

Thus, we conclude that

$$\begin{aligned} &\mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = 1 \right] \\ &\quad \cdot \Pr[C_t = 1 \mid e_t(v_i) \neq \emptyset] \\ &= \frac{\pi_i(1 - \pi_i)^{\ell-1}}{1 - (1 - \pi_i)^\ell} \sum_{k \leq \ell} \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}}. \end{aligned}$$

Extending the analysis to fixed $d \geq 1$, let us first define $E_{\{k_1, \dots, k_d\}, t}$ as the event $e_t(v_i) = \{b_{k_1}, \dots, b_{k_d}\}$. Note that all $E_{\{k_1, \dots, k_d\}, t}$ are mutually exclusive, where $\{k_1, \dots, k_d\}$ is a size- d combination from $\{1, \dots, \ell\}$, and the event $C_t = d$ is equivalent to $\bigvee_{\{k_1, \dots, k_d\}} E_{\{k_1, \dots, k_d\}, t}$.

Then,

$$\begin{aligned} &\mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = d \right] \\ &= \sum_{\{k_1, \dots, k_d\}} \mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid E_{\{k_1, \dots, k_d\}, t} \right] \\ &\quad \cdot \Pr[E_{\{k_1, \dots, k_d\}, t} \mid C_t = d] \\ &= \sum_{\{k_1, \dots, k_d\}} \mathbb{E} \left[\sum_{k \in \{k_1, \dots, k_d\}} |R_{b_k}^t - R_{b_k}^{t-1}| \mid E_{\{k_1, \dots, k_d\}, t} \right] \\ &\quad \cdot \Pr[E_{\{k_1, \dots, k_d\}, t} \mid C_t = d] \\ &= \sum_{\{k_1, \dots, k_d\}} \sum_{k \in \{k_1, \dots, k_d\}} \mathbb{E} [|R_{b_k}^t - R_{b_k}^{t-1}| \mid E_{\{k_1, \dots, k_d\}, t}] \\ &\quad \cdot \Pr[E_{\{k_1, \dots, k_d\}, t} \mid C_t = d], \end{aligned}$$

where

$$\begin{aligned} \Pr[E_{\{k_1, \dots, k_d\}, t} \mid C_t = d] &= \frac{\Pr[E_{\{k_1, \dots, k_d\}, t}]}{\Pr[C_t = d]} \\ &= \frac{\pi_i^d (1 - \pi_i)^{\ell-d}}{\binom{\ell}{d} \pi_i^d (1 - \pi_i)^{\ell-d}} \\ &= \frac{1}{\binom{\ell}{d}} \end{aligned}$$

and

$$\begin{aligned} &\mathbb{E} [|R_{b_k}^t - R_{b_k}^{t-1}| \mid E_{\{k_1, \dots, k_d\}, t}] \\ &= \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}}. \end{aligned}$$

Thus, for general $d \geq 1$,

$$\begin{aligned} &\mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = d \right] \\ &= \sum_{\{k_1, \dots, k_d\}} \sum_{k \in \{k_1, \dots, k_d\}} \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}} \frac{1}{\binom{\ell}{d}}. \end{aligned}$$

We note that the double summation can be rewritten as

$$\sum_{\{k_1, \dots, k_d\}} \sum_{k \in \{k_1, \dots, k_d\}} \equiv \sum_{k=1}^{\ell} \sum_{\substack{\{k_1, \dots, k_d\} \text{ s.t.} \\ k \in \{k_1, \dots, k_d\}}}.$$

and since the summands are independent of k_1, \dots, k_d given fixed k , the inner summation is simply the number of combinations $\{k_1, \dots, k_d\}$ with k inside, which is just $\binom{\ell-1}{d-1}$. Thus we have

$$\begin{aligned} &\mathbb{E} \left[\sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = d \right] \\ &= \sum_{k=1}^{\ell} \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}} \frac{\binom{\ell-1}{d-1}}{\binom{\ell}{d}}. \end{aligned}$$

and

$$\begin{aligned} &\mathbb{E} \left[\frac{1}{d} \sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid C_t = d \right] \\ &\quad \cdot \Pr[C_t = d \mid e_t(v_i) \neq \emptyset] \\ &= \frac{1}{d} \sum_k \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}} \frac{\binom{\ell-1}{d-1}}{\binom{\ell}{d}} \cdot \frac{\binom{\ell}{d} \pi_i^d (1 - \pi_i)^{\ell-d}}{1 - (1 - \pi_i)^\ell} \\ &= \frac{\binom{\ell-1}{d-1} \pi_i^d (1 - \pi_i)^{\ell-d}}{d \cdot 1 - (1 - \pi_i)^\ell} \sum_k \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}}. \end{aligned}$$

Thus, we have

$$\begin{aligned} &\mathbb{E} \left[\frac{1}{|e_t(v_i)|} \sum_{b \in e_t(v_i)} |R_b^t - R_b^{t-1}| \mid e_t(v_i) \neq \emptyset \right] \\ &= \left(\sum_{d=1}^{\ell} \binom{\ell-1}{d-1} \frac{\ell}{d} \cdot \frac{\pi_i^d (1 - \pi_i)^{\ell-d}}{1 - (1 - \pi_i)^\ell} \right) \\ &\quad \cdot \left(\frac{1}{\ell} \sum_k \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}} \right) \\ &= \frac{\sum_{d=1}^{\ell} \binom{\ell}{d} \pi_i^d (1 - \pi_i)^{\ell-d}}{1 - (1 - \pi_i)^\ell} \\ &\quad \cdot \left(\frac{1}{\ell} \sum_k \sum_j \frac{W_{i,j} |R_{j,k} - R_{i,k}|}{\sum_j W_{i,j}} \right), \end{aligned}$$

but since

$$\sum_{d=0}^{\ell} \binom{\ell}{d} \pi_i^d (1 - \pi_i)^{\ell-d} = (\pi_i + (1 - \pi_i))^\ell = 1,$$

we have

$$\begin{aligned} \sum_{d=1}^{\ell} \binom{\ell}{d} \pi_i^d (1 - \pi_i)^{\ell-d} &= 1 - \binom{\ell}{0} \pi_i^0 (1 - \pi_i)^{\ell-0} \\ &= 1 - (1 - \pi_i)^\ell \end{aligned}$$

or equivalently,

$$\frac{\sum_{d=1}^{\ell} \binom{\ell}{d} \pi_i^d (1 - \pi_i)^{\ell-d}}{1 - (1 - \pi_i)^{\ell}} = 1,$$

which we substitute into the above product to obtain our result.

PLACE
PHOTO
HERE

Felix Ming Fai Wong received the BEng, MSc and PhD degrees in computer engineering, computer science and electrical engineering from the Chinese University of Hong Kong, the University of Toronto and Princeton University respectively. He is currently a software engineer in search and data mining at Yelp Inc. Dr. Wong's research interests are in the application of optimization techniques to computer networks, online social networks and big data analytics. He was a winner of the Yelp Dataset Challenge, and a recipient of the Best Paper Award

Runner-up at IEEE INFOCOM 2015.

PLACE
PHOTO
HERE

Zhenming Liu is a postdoctoral research associate at Princeton University, working with Jennifer Rexford, Mung Chiang, and Vincent Poor. He received his PhD in theory of computation at Harvard in 2012. His doctoral research focuses on the intersections among applied probability, combinatorial optimization, and machine learning. More recently, he works with networking and data mining researchers to understand how these theoretical tools can help in building scalable systems for analyzing massive datasets. He has received several awards for his

research, including the Best Student Paper Award at ECML/PKDD 2010.

PLACE
PHOTO
HERE

Mung Chiang Mung Chiang (S'00, M'03, SM'08, F'12) is the Arthur LeGrand Doty Professor of Electrical Engineering at Princeton University and the recipient of the 2013 Alan T. Waterman Award. He created the Princeton EDGE Lab in 2009 to bridge the theory-practice divide in networking by spanning from proofs to prototypes, resulting in a few technology transfers to industry, several startup companies and the 2012 IEEE Kiyo Tomiyasu Award. He serves as the inaugural Chairman of Princeton Entrepreneurship Council and the Director of Keller

Center for Innovation in Engineering Education at Princeton. His Massive Open Online Courses on networking reached over 250,000 students since 2012 and the textbook received the 2013 Terman Award from American Society of Engineering Education. He was named a Guggenheim Fellow in 2014.