Organizational Overlap on Social Networks and its Applications

Cho-Jui Hsieh
University of Texas at Austin
cjhsieh@cs.utexas.edu

Mitul Tiwari
LinkedIn
mtiwari@linkedin.com

Deepak Agarwal
LinkedIn
dagarwal@linkedin.com

Xinyi (Lisa) Huang
University of Waterloo
x37huang@uwaterloo.ca

Sam Shah
LinkedIn
samshah@linkedin.com

ABSTRACT

Online social networks have become important for networking, communication, sharing, and discovery. A considerable challenge these networks face is the fact that an online social network is partially observed because two individuals might know each other, but may not have established a connection on the site. Therefore, link prediction and recommendations are important tasks for any online social network. In this paper, we address the problem of computing edge affinity between two users on a social network, based on the users belonging to organizations such as companies, schools, and online groups. We present experimental insights from social network data on organizational overlap, a novel mathematical model to compute the probability of connection between two people based on organizational overlap, and experimental validation of this model based on real social network data. We also present novel ways in which the organization overlap model can be applied to link prediction and community detection, which in itself could be useful for recommending entities to follow and generating personalized news feed.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Experimentation

Keywords

Social Networks, Organizational Overlap, Link Prediction, Community Detection

1. INTRODUCTION

The ability to model the organizational structure—that is, understand the affinity between links within an organization in the social graph—is an important problem for online social networks. For example, a substantial challenge is that any online social network is only partially observed: two individuals may know each other, but may not have established a connection on the site. Thus, most online social network sites have a feature that recommends these possible connections. LinkedIn, the largest online professional social network, exposes its link prediction system through its "People You May Know" feature.

Detecting communities within an organization is another important challenge. On most online social networks, a user can follow an entity to receive updates on it within a personalized news feed. For example, members can follow a company on LinkedIn and receive company updates. To recommend entities for a member to follow, the entities the member's community are already following are good candidates. Simply using the entire organization yields inferior results as most organizations are diverse with several orthogonal groups (for example, sales, marketing, engineering) and subgroups (for example, front-end, database, machine learning). As another example, this news feed is generated by online activity and its volume can quickly overwhelm a user. A key feature in ranking this feed is to promote an update if the member is in the same community as the originator of the update.

In this paper, we present a model of edge affinity between two users that uses the time of joining and departing an organization. Here, the intuition is simple: the affinity between two members working in a company together for 10 years is greater than members who've worked at the company for only a few months. In this work, we present a mathematical model of organizational time overlap and experimental validation of this model based on real social network data from LinkedIn. We also show empirically that our model works for diverse organizations such as companies, schools, and online groups.

This work aids the link prediction scenario as previous approaches [1, 9, 10, 18, 29, 31] have mainly exploited existing edges in the social graph, but not organizational time overlap. Using this model to predict existing edges on LinkedIn and two other public networks show that our method is 42% better than Common Neighbor and Adamic-Adar based link prediction in terms of precision at top5.

The organizational overlap model also works well for detecting communities within an organization. It is usually hard to evaluate the quality of communities because of a lack of ground truth. We therefore use an indirect method to evaluate the quality of communities: intuitively, the speed of information propagation should be faster within a community, so we measure the quality of the detected communities by the speed of information propagation within it. Using the LinkedIn social network, we evaluate the detected communities by the propagation speed of company follows and sharing activity. Results show that communities detected by our method are up to 66% better than communities detected by only

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

links in terms of the propagation speed of shared articles, and 15% better in terms of the propagation speed of company follows.

The **main contributions** of this paper include:

- Experimental insights from social network data on organizational overlap
- A novel mathematical model to compute the probability of connection between two people based on organizational overlap, including experimental validation of the model
- New ways the organizational overlap model can be applied to link prediction and community detection.

The rest of the paper is organized as follows. First, Section 2 describes related work. Section 3 presents experimental insights in modeling probability of connection between two people based on organizational overlap, a mathematical model formulation, and experimental evaluation of the model. Section 4 and Section 5 discuss applications of our organization overlap model for community detection and link prediction. We conclude in Section 6 and discuss future work.

2. RELATED WORK

In this paper, we model the probability of connection between people using organizational time overlap information, and apply our model to community detection and link prediction. Although there are many studies on modeling organizational overlap of people [11, 31], and some of the properties of community structure in social networks have been discussed previously in Leskovec et al. [13], but none of them have incorporated the overlap time information in their models. In this section, we discuss the related work for community detection and link prediction.

Community detection is an active research topic in social network analysis [2, 7, 8, 14, 20, 21]. Traditional community detection algorithms find a partition of graph to optimize a measurement of the quality based on the given network, for example, modularity-based methods aim to find a partition of the graph that maximizes the modularity [5]; or spectral clustering methods, such as normalized cut, minimize the number of between-cluster edges [6, 26]. One drawback of these community detection techniques is that they only use mainly edge information. Recently, the use of explicit feature information for community detection has been studied. In Pathak et al. [23], the Community-Author-Recipient-Topic (CART) model has been proposed to combine the topic information with graph structure for predicting communities. Also, the topic-link LDA model [17] and relational topic model [4, 24] were also proposed to incorporate the content information. However, none of these existing works consider using organizational time overlap for community detection, which we will show in this paper can outperform the use of graph only algorithms.

Predicting an unknown link in a given social network is another important application, especially for online social networks such as Facebook, LinkedIn, and Twitter. Similar to the case of community detection, traditional link prediction algorithms predict unknown edges based only on given edges in the social networks [15]. Examples of such techniques are common neighbor (CN), Katz [10] and Adamic-Adar (AA) [1], which are efficient to compute and are widely used. Recently, graphical model-based methods have also been applied to obtain more accurate predictors [19, 29]. However, these methods are only based on link structure. If we are given context information, supervised learning frameworks [9, 16] have been proposed to train a linear model based on the given features. For example, Backstrom and Leskovec [3] used a weighted random walk strategy for predicting unknown edges. Soundarajan and Hopcroft [28] proposed to use community overlap (without time information) to improve common neighbor. In this paper, we also consider a cold-start problem of link prediction [12], where we want to predict connections of a new user on a social network and users who are not connected to any other users. In this case, the connection information is unknown, and we show that using organizational time overlap is a very promising way to recommend connections in a cold-start setting.

3. MODELING ORGANIZATIONAL TIME OVERLAP

To start, we formally define organizational time overlap as the following. Assume two people A and B belonged to the same organization O, A belonged to the organization during time interval $t_A = [s_A, e_A]$ and B belonged to the same organization during time interval $t_B = [s_B, e_B]$. We can compute the organizational time overlap between A and B in organization O as

$$T(A, B, O) = \max(0, \min(e_A, e_B) - \max(s_A, s_B)). \qquad (1)$$

Note that a person can belong to an organization for more than one time interval. That is, consider A belonged to the organization in time intervals $t_A^{(1)}, t_A^{(2)}, \cdots, t_A^{(n)}$ and B belonged to the same organization during time intervals $t_B^{(1)}, \cdots, t_B^{(m)}$. We can then compute $T(A, B, O)$ by summing over all nm pairs of intervals. However, usually people belong to an organization (for example, work in a company) only once, so m and n are typically 1.

In this section, given $T(A, B, O)$ for all organization O, we aim to compute the probability that A and B are connected. We will use $P(A, B)$ to denote this probability. We first assume that A and B have only one organization in common, so

$$P(A, B) = f(T(A, B, O), O). \qquad (2)$$

Equation (2) means the probability that A and B are connected depends on two factors: the organizational time overlap $T(A, B, O)$ and the property of the organization O. We first investigate this function using data from LinkedIn, and then derive a mathematical model of f based on our observations.

3.1 Exploratory Data Analysis

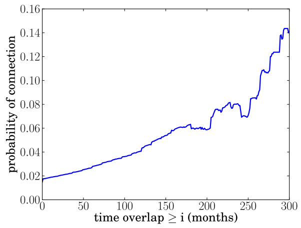
We use the organizational time overlap data from LinkedIn to investigate the properties of $P(A, B)$, the probability of connection between A and B. LinkedIn is the largest professional social network, where more than 200 million members maintain their profile and connections. A member's profile normally includes employment history, education information, and online group affiliations. From LinkedIn data we can obtain several different types of organization, for example, "company", "group" and "school". Here we use "company" and "group" as two types of organization. Also, we consider each member on LinkedIn as a node in a graph, and the "connections" on LinkedIn as edges between nodes.

As expressed in Equation (2), $P(A, B)$ depends on two factors: the time overlap and properties of the organization. We first investigate $P(A, B)$ for a fixed organization O and changing the time overlap $T(A, B, O)$. Assuming n users had worked in company O, we compute the time overlap for all $\binom{n}{2}$ pairs of users, and compute the following empirical probability

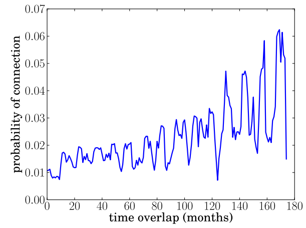
$$\hat{p}(t) = \frac{\text{\# connections with time overlap } t}{\text{total number of pairs with time overlap } t}. \qquad (3)$$

Because the denominator of (3) can be very small for some values of t, which makes $\hat{p}$ unstable, we also plot

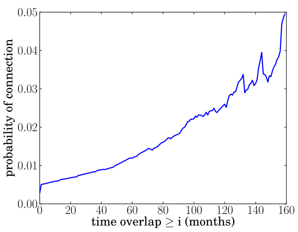
$$\tilde{p}(t) = \frac{\text{\# connections with time overlap} \geq t}{\text{total number of pairs with time overlap} \geq t}. \qquad (4)$$



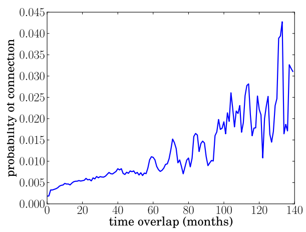
(a) Company A (cumulated)



(b) Company A



(c) Company B (cumulated)



(d) Company B

Figure 1: Connection density with respect to time overlap for two sampled companies A and B. Observe that the probability of connection increases with the time overlap.

Figure 1 shows $\tilde{p}$ and $\hat{p}$ of two randomly-sampled companies. We observe that the probability of connection increases with the time overlap t. This observation leads to the following insight:

Insight 1 *Connection density increases with organizational time overlap.*

Also, as shown in Equation (2) the probability of making a connection is related to the properties of the organization. Therefore, we also investigate the relationship of $P(A, B)$ with the organization size. From our data analysis, it appears that other forms of organization such as schools and groups on LinkedIn also follow a similar distribution. To investigate the relationship between probability of connection and the company size, we sample companies with different sizes $(25, 50, 75, 100, 200)$ and calculate the probability of connection (as defined in Equation (3)) $\hat{p}$ with various time overlap, as shown in Figure 2. We can observe that generally the probability of a connection decreases as organization size increases. This observation makes sense because in a smaller organization people usually have a better chance to know each other by happenstance, which is not the case in a larger organization. Therefore, we have the following second insight:

Insight 2 *Connection density decreases with the size of the organization.*

3.2 Mathematical Formulation

In the previous section, we observed that organizational time overlap is an important factor for computing the probability of a connection. In this section, we derive a mathematical formulation of $P(A, B)$, the probability of a connection between A and B. For convenience, we use $P(t)$ to denote $f(t, O)$ in Equation (2), that is, $P(t)$ is the probability of an edge between two nodes with time overlap t.

Our first insight is that the problem can be transformed to an "overlapping community" problem. In an overlapping community setting, each user can belong to more than one community. To compute $P(t)$, we can decompose t into two smaller time periods t_1, t_2 such that $t = t_1 + t_2$. With this decomposition, a natural way to model the probability $P(t_1 + t_2)$ is to think as if there are two organizations: O_1 corresponds to time interval $[0, t_1]$ and O_2 corresponds to the time interval $[t_1, t_1 + t_2]$. Because the probability of making a connection in O_1 and O_2 are $P(t_1)$ and $P(t_2)$ respectively, assume $P(t_1)$ and $P(t_2)$ are known. The problem of computing $P(t)$ is now transformed to how to compute the probability that an edge between A and B exists when both A, B belong to O_1 and O_2. A large amount of research has been conducted for modeling probability given an overlapping community structure in social networks [22, 30], and recently a simple Community-Affiliation Graph Model (AGM) [31] was proposed. As proposed in AGM, assume that A and B both belong to $O_1, \dots, O_n$, and that the probability of making a connection in $O_1, \dots, O_n$ are $p_1, \dots, p_n$, then

$$1 - \prod_{i=1}^{n}(1 - p_i)$$

is the probability that A is connected to B. In our case, $n = 2$, $p_1 = P(t_1)$, $p_2 = P(t_2)$, so we can make the following assumption:

Assumption 1 *For any two non-negative time intervals t_1, t_2,*

$$P(t_1 + t_2) = 1 - (1 - P(t_1))(1 - P(t_2)) \tag{5}$$

Also, because $P(t)$ is a probability, we can simply assume the following:

Assumption 2 *For any $t > 0$, $1 > P(t) > 0$, and $P(t) = 0$ if and only if $t = 0$.*

We can then easily generalize Assumption 1 to decompose a time interval t into m smaller time intervals, resulting in the following lemma:

Lemma 1 *For any m nonnegative integers $t_1, \dots, t_m$,*

$$P(\sum_{i=1}^{m} t_i) = 1 - \prod_{i=1}^{m}(1 - P(t_i)) \tag{6}$$

PROOF. Proof by induction, when $m = 2$, (6) is equivalent to Assumption 1. Assume (6) holds for $m = a$, for $m = a + 1$ we have

$$\begin{aligned} P(\sum_{i=1}^{a+1} t_i) &= P(t_{a+1} + \sum_{i=1}^{a} t_i) \\ &= 1 - (1 - P(t_{a+1}))(1 - P(\sum_{i=1}^{a} t_i)) \quad \text{(by Assumption 1)} \\ &= 1 - \prod_{i=1}^{a+1}(1 - P(t_i)) \quad \text{(by induction hypothesis)} \end{aligned}$$

□

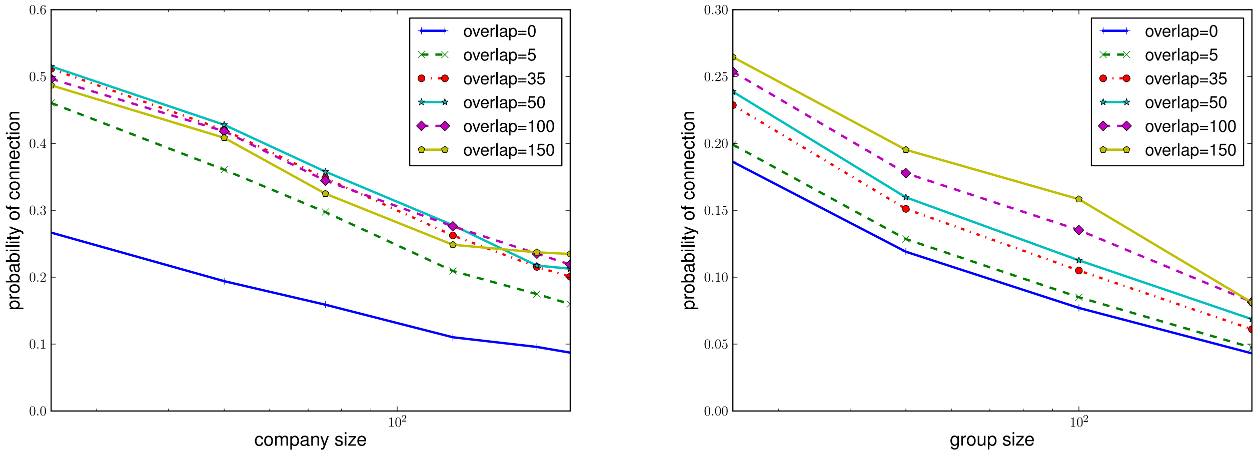
Lemma 2 *$P(t)$ is continuous on $t \in (0, \infty)$.*

PROOF. First, from Lemma 1 and Assumption 2 we can easily show $\lim_{\delta t \to 0^+} P(\delta t) = 0$. Using this fact, from Assumption 1 we have

$$P(t_1 + t_2) = P(t_1) + P(t_2) - P(t_1)P(t_2), \forall t_1, t_2 \geq 0.$$

Let $t_1 = t$ and $t_2 = \delta t$. We have $\lim_{\delta t \to 0^+} P(t + \delta t) = P(t)$. Also, let $t_1 = t - \delta t$ and $t_2 = \delta t$ we have

$$P(t - \delta t) = P(t) - P(\delta t)/(1 - P(\delta t)).$$



(a) company size (b) group size

Figure 2: Connection density with respect to organization size. We observe that the probability of a connection decreases as organization size increases.

Thus, $\lim_{\delta t \to 0^-} P(t + \delta t) = P(t)$. $\square$

Based on the above assumptions and lemmas, we can then prove $P(t) = 1 - e^{-\lambda t}$ for some positive constant λ by the following arguments.

First, we use $Q(t) \equiv 1 - P(t)$ to denote the probability that two users with time overlap t do not know each other. From Lemma 1, we have

$$Q(\sum_{i=1}^{m} t_i) = \prod_{i=1}^{m} Q(t_i), \tag{7}$$

therefore

$$Q(nt) = Q(t)^n \text{ and } Q(\frac{t}{n}) = Q(t)^{\frac{1}{n}}$$

for any integer n. Furthermore,

$$Q(\frac{n_1}{n_2}) = Q(\frac{1}{n_2})^{n_1} = Q(1)^{\frac{n_1}{n_2}}.$$

Therefore, $Q(t) = Q(1)^t$ for any rational number t. Because $P(\cdot)$ is continuous, $Q(\cdot)$ will also be continuous. Therefore, $Q(t) = Q(1)^t$ for any real number $t \in [0, \infty)$. Based on Assumption 2, $1 > Q(1) > 0$. We can rewrite $Q(1) = e^{-\lambda}$ where $\lambda > 0$. We then have the following main theorem for modeling organizational time overlap:

Theorem 1 *Let $P(t)$ denotes the probability that two users with organizational overlap of time t know each other, then*

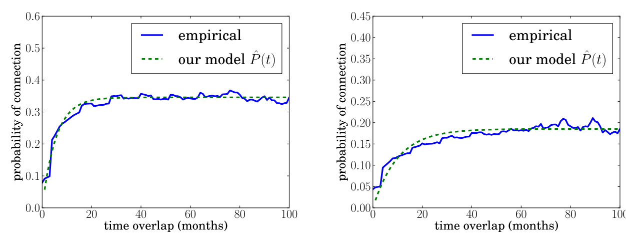
$$P(t) = 1 - e^{-\lambda t}, \textit{ for some } \lambda > 0. \tag{8}$$

3.3 Experimental Validation

Next, we verify the model we derived in Theorem 1 using real-world datasets from LinkedIn, and then show how to select the parameter λ in Equation (8).

3.3.1 Probability of connection versus time overlap

To verify that our model fit the real social network, we conducted the following experiments. Using the companies with the same size (50 or 200), we generated all the within-company pairs and computed the empirical distribution $\hat{p}(t)$ for $P(t)$ as in Equation (3).



(a) Companies with size 50 (b) Companies with size 200

Figure 3: Connection density with respect to time overlap comparing our model with respect to empirical data.

The results are shown in Figure 3. Although Theorem 1 indicates that $P(t)$ can be 1 when $t \to \infty$, in a large company it is not realistically possible for people to know all their colleagues. Empirically, we observe there is an upper bound of the probability of connection within a company. Thus, the probability can be modified to

$$\hat{P}(t) = \mu(1 - \exp(-\lambda t)),$$

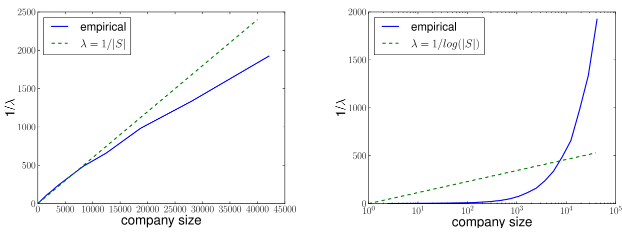
where $0 \le \mu \le 1$ is a constant controlling the upper bound of the probability. We also plot $\hat{P}(t)$ in Figure 3, with μ selected to be the maximum empirical probability when $t \to \infty$, and λ selected to be the best parameter to fit the curve. We can see that the empirical distributions are close to the theoretical guess $\hat{P}(t)$, which supports that our model derived in Theorem 1 is a good fit to the empirical observation.

3.3.2 The Relationship Between λ and Company Size

The parameter λ described in Equation (8) is a company-dependent parameter, which may depend on many company properties. An obvious observation is that employees in a small company are more likely to know each other, while in a large company it is less likely. Therefore, we conducted the following experiment to investigate the relationship between λ and company size.

For simplicity, we assume

$$\frac{|E|}{|S|^2} = 1 - e^{-\lambda \bar{t}}, \tag{9}$$



(a) (b)

Figure 4: (a) Comparison of the empirical statistics of λ to the hypothesis that $\lambda = 1/|S|$, where $|S|$ is the company size. This figure shows that the rate of increase of $1/\lambda$ is slower than that of $|S|$. (b) Comparison of the empirical statistics of λ to the hypothesis that $\lambda = 1/\log(|S|)$. This figure shows that empirically the rate of increase of $1/\lambda$ is higher than that of $\log(|S|)$.

where $|E|$ is number of edges within a company, $|S|$ is number of nodes in the company, and $\bar{t}$ is the average organizational time overlap in that company. Then we have

$$\frac{1}{\lambda} = -\frac{\bar{t}}{\log(1 - |E|/|S|^2)}. \tag{10}$$

We do not know the value of λ. However, the righthand side of (10) can be computed from real social networks. To see how the company size affects the probability of making a connection, which depends on p, we can investigate how the righthand side value of (10) changes with company size. For each company size $|S|$, we extract all the companies with size $|S|$ from LinkedIn data, and use the connection data to compute the righthand side of (10) empirically.

A simple assumption is that $\lambda \propto \frac{1}{|S|}$ where $|S|$ is company size. We compare the righthand side of (10) and this hypothesis in Figure 4(a). However, we find the decay rate of p with respect to company size $|S|$ is slower than linear.

We then consider the second hypothesis: $\lambda \propto \frac{1}{\log(|S|)}$. This is a function slower than the linear decay. Therefore, we plot $1/\lambda$ versus log of company size in Figure 4(b). If $\lambda \propto \frac{1}{\log(|S|)}$, the empirical line in Figure 4(b) is a straight line; however, it is growing faster than a straight line, which indicates that $\frac{1}{\lambda}$ grows faster than the log function and slower than the linear function.

Therefore, our third hypothesis is that

$$\lambda = \beta|S|^{-\alpha} \quad \text{for some } 1 > \alpha > 0 \text{ and } \beta > 0, \tag{11}$$

which is between $1/|S|$ and $1/\log(|S|)$. To verify this hypothesis, we first assume Equation (11). We define

$$y(|S|) := \frac{1}{\lambda} = \frac{1}{\beta S^{-\alpha}} = \frac{-\bar{t}}{\log(1 - |E|/|S|^2)}.$$

To compute α, we compute the ratio between $y(|S|)$ and $y(10)$[1] as

$$\frac{y(|S|)}{y(10)} = \frac{|S|^\alpha}{10^\alpha} = (\frac{|S|}{10})^\alpha,$$

so $\alpha = \log_{|S|/10}(\frac{y(|S|)}{y(10)})$. Because the righthand side can be computed empirically, we plot this empirical α versus company size in Figure 5. Figure 5 shows that the α value is stable between 0.8 and 0.85 for company sizes larger than 1000.

3.3.3 Better Estimation of λ

[1]Any other positive constant can be used here.

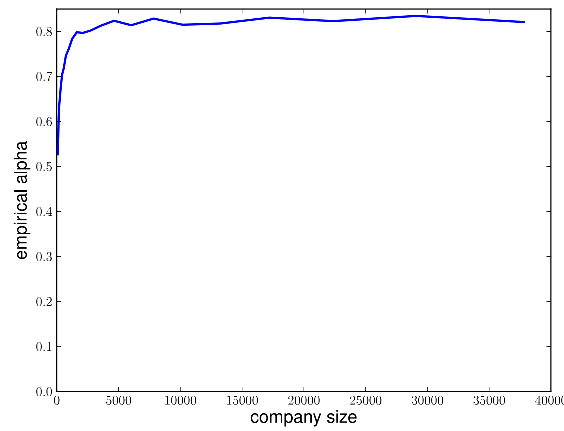


Figure 5: Comparison of the empirical statistics of λ to the hypothesis that $\lambda = 1/|S|^\alpha$, where $|S|$ is company size. This figure shows that α is stable between 0.8 and 0.85 for company sizes larger than 1000.

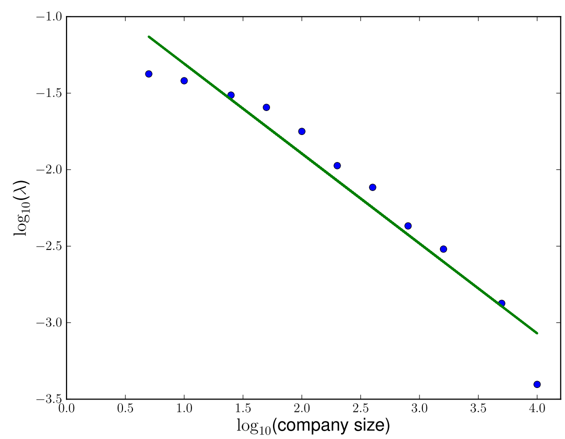


Figure 6: MLE estimates of λ as function of company size. Figure displays a log-log relationship.

We estimated λ heuristically as a solution to equation (9), which can roughly be thought of as a method-of-moments estimator assuming every edge in an organization has an overlap period close to $\bar{t}$. This might not be true and we do see heterogeneity in time overlaps in our data. To adjust for such heterogeneity, we estimate λ through a maximum likelihood (MLE) approach. To investigate the relationship between λ and company size $|S|$ more rigorously, we obtain a separate MLE estimate for companies of different sizes and investigate the log-linear relationship $\lambda = \beta|S|^{-\alpha}$.

Each pair i in our graph has a pair of observations (t_i, X_i), where t_i is the time overlap and X_i is binary indicator of presence/absence of an edge. According to our model, X_is are independently distributed Bernoulli random variables with mean $(1 - e^{-\lambda t_i})$, hence the MLE of λ is readily obtained by maximizing

$$\sum_i (X_i log(1 - e^{-\lambda t_i}) - (1 - X_i)\lambda t_i)) \tag{12}$$

Figure 6 shows the estimated values of λ as a function of different company size. Again, we clearly see a monotonically decreasing trend in values of λ as function of size, and the log-linear rela-

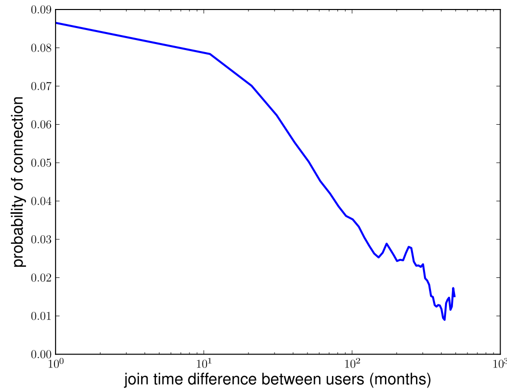


Figure 7: Probability of making a connection with respect to the difference in the join time between two users. The difference in the join time seems to be an important factor in computing the probability of a connection.

tionship posited earlier holds. In fact, a least squares fit of MLEs of $\log(\lambda)$ on $\log|S|$ provides an estimate $\alpha = 0.59$. But as evident from the figure, the value of α is higher if we ignore small companies, consistent with our earlier empirical analysis. For example, if we ignore companies with size smaller than 20, the α will be 0.7.

3.3.4 The Effect of the Difference in Join Time

Another observation is that the difference in the join time is important. Intuitively, an employee makes many connections when the employee first joins a company, while the probability of making connections tends to decay as the employee works longer in the company. Therefore, with the same overlapping time, the probability of a connection between two employees also depends on the difference in the join time. In Figure 7, from LinkedIn dataset, we plot the probability of connection versus join time difference with the same overlapping time (1 year). The results indicate that this observation is important for modeling the probability, and we will use this as a signal in the link prediction (see Section 5).

4. COMMUNITY DETECTION

Community detection within an organization is an important problem for online social networks like LinkedIn. On most social network, one can follow an entity to receive updates on it within a personalized news feed. For example, members can follow a company on LinkedIn to receive updates from the company. To recommend entities to a member to follow, the entities member's community are already following are good candidates. Therefore, detecting communities on a social network is an important task for recommendation applications, such as companies to follow or articles to read.

Given the link information in a graph, traditional community detection methods utilize the graph structure to find communities [5–7, 20, 21, 26]. However, in real applications, link information is often too sparse and may not be fully observed in a social network. Thus, traditional community detection methods give poor performance in such cold start setting. In this section, we will demonstrate that we can detect good quality communities using the organizational time overlap signal in such cold start setting.

Evaluating community detection algorithms is a hard problem [14, 21] even if labeled gold-standard ground-truth communities are given [32]. In our applications, the ground truths are not given. Therefore, we consider the detected communities as the input of two applications, vitality of company follow and virality of updates, and define the quality of communities based on the virality of these two events, which has real applications in practice. In this section, we first describe how to use time overlap information for community detection, and then use two applications, virality of company follow and virality of updates, to evaluate the detected communities and demonstrate that our algorithm is very useful for a social network.

4.1 Within-organization Community Detection by Organizational Time Overlap

In many social networks, such as LinkedIn, the profile of a user includes employment, education, groups, and other organization affiliations. In such datasets, a natural way would be grouping users by organizations (can be companies, groups, or schools). However, most organizations are diverse with several orthogonal groups (for example, sales, marketing, and engineering) and subgroups (for example, front-end, database, machine learning). Therefore, we are interested in finding communities within an organization, which is important when we want to get hierarchical community structures for recommendation applications. For each specified organization, we are given users who belonged to the organization at some point of time, and we want to find communities among these users.

To start, we construct a graph using the organizational time overlap signal described in the previous section. We use the first order Taylor expansion on $e^{-\lambda t}$ on Equation (8) (since $\lambda t \ll 1$ usually), so

$$P(t) \approx \lambda t, \tag{13}$$

where λ is a company-dependent parameter, and so is a constant under this setting. Based on this observation, we construct an organizational time overlap graph $G_T(V, E)$ by setting the node set V to be all the users in the organization and an edge (A, B) in E if the organizational time overlap between A and B is more than 0, with weight equal to the organizational time overlap of the two end nodes.

We can then run any graph-clustering algorithms on this organizational time overlap graph. In our experiments, we use Graclus [6], which is a multilevel graph-clustering approach minimizing the following normalized cut value:

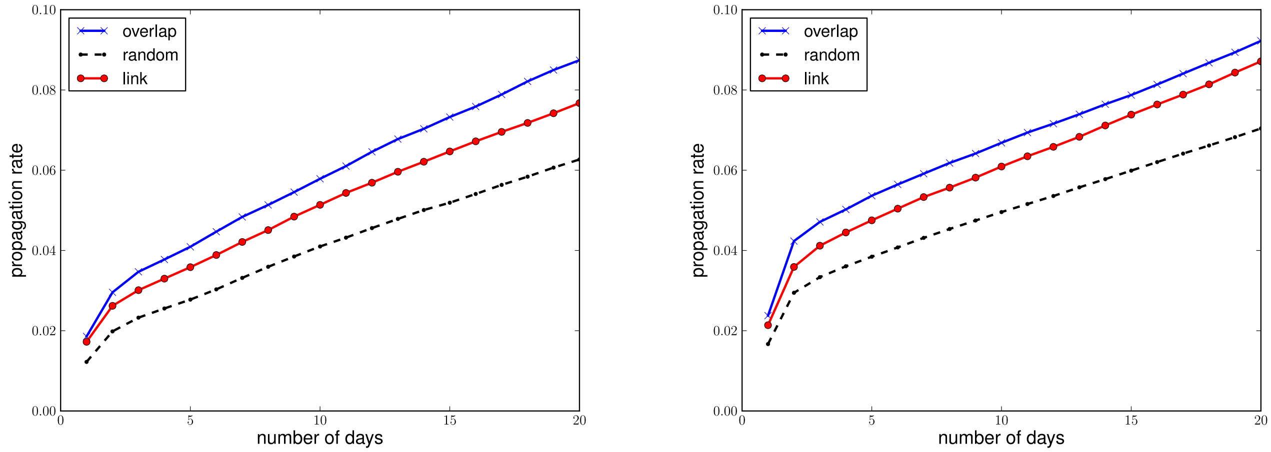
$$\text{NCut}(G, \{\mathcal{V}_c\}_{c=1}^k) = \sum_{c=1}^{k} \frac{\sum_{i\in\mathcal{V}_c, j\notin\mathcal{V}_c} G_{ij}}{d(\mathcal{V}_c)}$$

$$\text{where } d(\mathcal{V}_c) = \sum_{i\in\mathcal{V}_c}\sum_{j=1}^{p} G_{ij}.$$

Notice that sometimes G_T may be dense and the clustering on G_T may be expensive. In those cases, we want to further approximate G_T using less memory. We first observe that the time overlap defined in (1) can be approximated by

$$T(A, B, O) = \sum_{s=1}^{k} I(A, O, s) I(B, O, s), \tag{14}$$

where we divide the time into k buckets, and each $I(A, O, s)$ is the indicator function of whether A was in O at the s'th time bucket. Based on this observation, we can approximate G_T by $U^T U$ where U is a $n \times k$ matrix and $U_{is} = I(i, O, s)$. The memory consumption of this $U^T U$ approximation is $O(nk)$, which may be much less than storing G_T. For example, if each month is a time bucket and the average time for each user to stay in company O is 5 years, than storing U costs $60n$ memory while storing G_T may use



(a) Average degree: 4-6

(b) Average degree: 12-14

Figure 8: The propagation rate of company follow with respect to the number of days

$O(n^2)$ memory. Moreover, with this memory-efficient representation, eigenvalue solvers such as Lanczos method can be applied efficiently to compute leading eigenvectors since each matrix-vector product can be easily computed, thus spectral clustering can be conducted efficiently.

In the following section, we conduct two sets of experiments to show that clustering using organizational time overlap can be useful. Because the size of organization is not usually too large in our datasets, we directly use G_T with Graclus in all the experiments.

4.2 Virality of Company Follow

On LinkedIn, a company can create a page with a profile, and post updates about the company. Any LinkedIn member can follow a company to receive updates. When a member chooses to follow a company, an update is published that the member is following the company, and that update is shared with the member's connections (if the member permits). LinkedIn has a recommendation system that recommends companies that a member can follow. In this work, we explore how clustering members belonging to a company can help in recommending companies to follow. To evaluate the quality, the detected clusters, we measured the spread of company following inside the clusters.

4.2.1 Evaluation and Discussion

To compare the communities detected by edges and by organizational time overlap, we sample companies with varying average degrees. More specifically, we consider companies with average degrees of $0-2$, $4-6$, $8-10$, $12-14$, and $16-18$. We randomly sample 100 companies for each interval.

For each of the selected companies, we run the following three community detection algorithms to detect 10 non-overlapping communities and compare the results:

- Community detection by organizational time overlap, as described in Section 4.1. Because the company size is generally not too large, we directly form G_T and run Graclus to detect the communities.

- Community detection by links. We construct the adjacency matrix for users belonging to each company, and run Graclus to detect 10 communities.

- Random. As a baseline method, we randomly partition the nodes into 10 communities with community size the same as the communities detected by time overlap.

We define a measure of the spread of company following event inside detected communities as follows. For each user i and company C, we are given $T(i,C)$ which denotes the time that i follows C ($T(i,C)=\infty$ if i never followed C). Given a partition of nodes, $\{\mathcal{V}_c\}_{c=1}^k$, we compute the first company follow time for each community $\mathcal{V}_c$ by

$$F(\mathcal{V}_c, C) := \min_{i\in\mathcal{V}_c} T(i,C).$$

Assume $A(\mathcal{V}_c)$ denotes the set of companies C, which is followed at least once in $\mathcal{V}_c$, then we can measure the spread of company following inside each detected community by

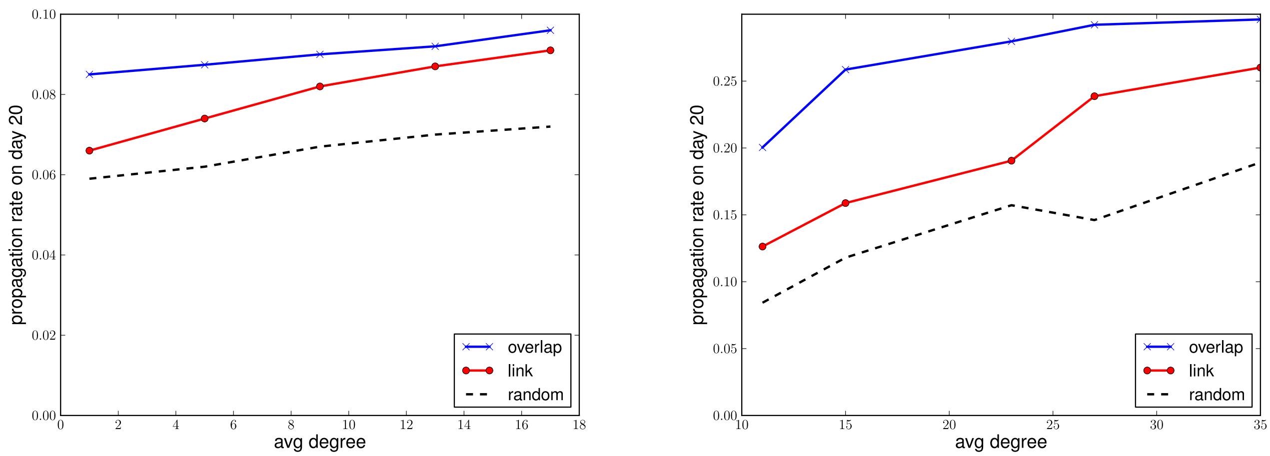
$$S(\mathcal{V}_c, d) = \frac{\sum_{C\in A(\mathcal{V}_c)} |\{i : i\in\mathcal{V}_c \text{ and } T(i,C) < d\}|}{|A(\mathcal{V}_c)|},$$

which is the average number of companies followed within d days of the first following event. We then normalized the previous measurement by the number of users in each community and computed the following *propagation rate*:

$$M(d) = \sum_{c=1}^{k} \frac{S(\mathcal{V}_c, d)}{|\mathcal{V}_c|}. \tag{15}$$

A partition $\{\mathcal{V}_c\}_{c=1}^k$ with a higher $M(d)$ value means the company follow event spreads faster within the community. For each of the three clustering algorithms described, we compute the $M(d)$ value and show the results for various link densities of companies. The results are shown in Figure 8.

We found that the rate of spread of company following inside detected communities is much faster than the rate of spread of company following inside communities formed by randomly grouping members. Interestingly, the communities detected by time overlap are consistently better than that by link. In addition, we can observe that the difference between them is large when average degree is between $4-6$, and becomes smaller when the average degree increases to $12-14$. This observation is not surprising because the link information is not enough to find good communities for companies with low average degree, while it is affective



(a) Company Follow propagation rate with respect to degree

(b) Articles propagation rate with respect to degree

Figure 9: The propagation rate of company follow and articles with respect to degree. As average degree increases, the propagation rate increases in communities detected by link information.

for companies with high average degree. We further investigate this phenomenon in Figure 9(a). As the average degree increases, the propagation rate of communities detected by link and overlap becomes closer. The average degree of companies usually follows the power-law distribution, which means there are many companies with very few connections, and only few companies with dense connections. Therefore, clustering by organizational time overlap is very useful in practice.

4.3 Virality of Updates

Besides the company following event, we can also use the virality of updates on LinkedIn to evaluate the detected communities.

On LinkedIn, users can view an article (in their feed of updates from connections) that are shared by a connected user, and reshare that article to their connections. We use the speed of an article's propagation within a cluster of users to evaluate the quality of clusters. We show that the propagation speed of articles among the communities detected by using organizational overlap information is much faster than those using link information, and in certain cases, the difference is up to 100%. Therefore, for a given user, articles shared by other users in the same community are good candidates to recommend on the user feed of updates.

4.3.1 *Evaluation*

To generate randomly-sampled companies, we follow the same methodology mentioned in the previous section. That is, we sample companies with varying average degrees. Similar to the experiments on company follow events, we sample companies with a wider range of average degree. For each interval, we randomly sample 100 companies.

We test the quality of communities detected by organizational overlapping time, by link information, and by random generation.

To measure an article's propagation speed, we define the measurement similar to Equation (15). The only change here is that $T(i, C)$ is now defined as the time that user i read article C on LinkedIn. We again plot the propagation rate $M(t)$ for the shared articles in Figure 10. For each of the three clustering algorithms described previously, we compute the $M(t)$ value and show the results for the various link densities of companies. We again see that communities detected by time overlap have faster propagation speeds than communities detected by link information or random partitions. Also, similar to the company follow experiments, as shown in Figure 9(b), when average degree increases, the propagation speed improves in communities detected by using link information.

5. LINK PREDICTION

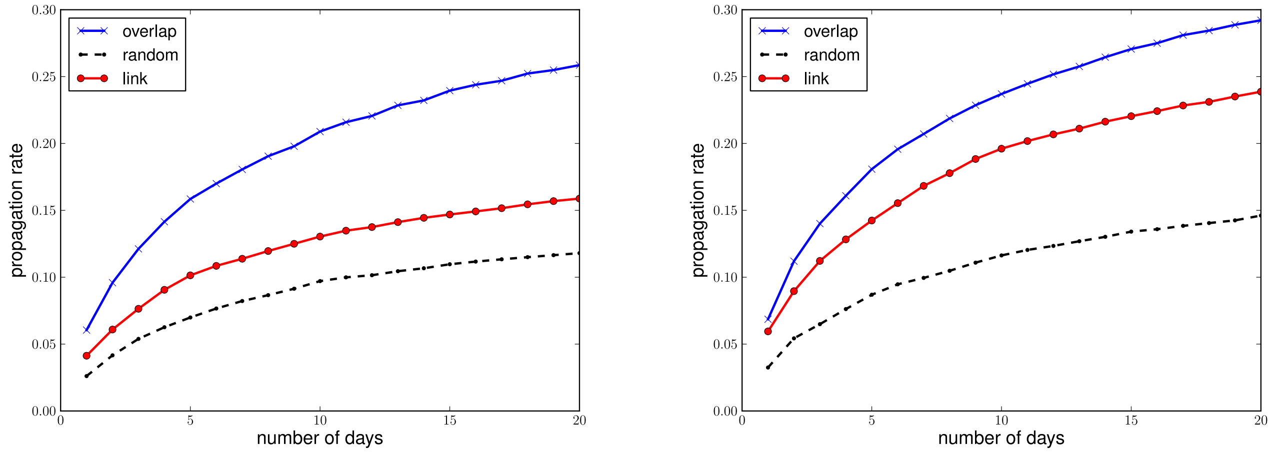
To apply the organizational time overlap model for link prediction, we consider link prediction under two settings: warm start problems and cold start problems.

In a *warm start link prediction problem*, we are given the current connection between users at some specific time, and the task is to predict future connections between users. In other words, we are given a graph $G(t_1) = (V, E(t_1))$ at time t_1 where edges correspond to connections at time t_1, and the task is to predict edges in $G(t_2) = (V, E(t_2))$, for some time $t_2 > t_1$. As discussed in Section 2, many link prediction algorithms (for example, as in Liben-Nowell and Kleinberg [15]) have been proposed to predict future edges based on current edges, and these algorithms are widely used in practice. However, we will show that a supervised learning method using only time overlap and company size information outperforms traditional link-based predictors.

The *cold start link prediction problem* is another setting that we use to predict future links for a given node with no link information to this node. This problem is harder because less information exists, but it is important in practice. For example, when a new user joins a social network, such as LinkedIn or Facebook, it is very important to provide good recommendations for connections to engage the user. Therefore, the cold start problem is getting more recent attention in both link prediction and recommender systems [12, 25]. We will tackle this problem by using the organizational time overlap signal with other information. We will show that our model is very effective in the cold start problem setting.

As shown in Equation (13), when λ and t are small, $P(t)$ can be approximated by a linear model; therefore, we use the following linear model to approximate $P(t)$:

$$P(t) \approx \sum_{i=1}^{d} w_i x_i,$$



(a) Average degree: 4-6

(b) Average degree: 12-14

Figure 10: The propagation rate of articles with respect to the number of days

where $\boldsymbol{x} = [x_1, \cdots, x_d]$ denotes features, including overlap time t and company properties, and $\boldsymbol{w} = [w_1, \cdots, w_d]$ denotes model parameters. We consider the following features on a pair of users:

- Time overlap: The organizational time overlap of two users.
- Company size: As we discussed in Section 4, the parameter λ highly depends on $|S|^{-0.8}$, where $|S|$ is the company size.
- Company propensity: How likely employees in a company are friends. We compute this feature by

$$2(\text{number of connections in company})/(\text{company size})^2.$$

- Company average age: We compute the average age of each company.
- Company cluster coefficient: We compute the cluster coefficient for each company by

$$\frac{\text{number of closed triplets}}{\text{number of connected triples of vertices}}.$$

  The cluster coefficient measures how closely users are connected in a company.
- Node propensity: We use average degree of each node as the node propensity, which denotes how likely the node is connected to another node.
- Join time difference: As discussed in Section 3.3.4, the join time difference is also important to the probability of connection. Thus, we also use this as a feature in our model.

For each problem, we consider two models:

- *2-features model*: using only time overlap and company size as features.
- *all-features model*: using all the features described in the model.

For warm start problems, we compare the link-based methods Common Neighbor (CN) and Adamic-Adar (AA) with our *2-features model*. We do not compare CN and AA with models utilizing more features because other features include the link information, and we want to show our model outperforms link-based methods without *utilizing any link information*. For both methods, we use the training data (connections before the specific time $\bar{t}$) to compute the measurement or learn model parameters, and compare the top-k accuracy (as defined in below) on the testing data (connections after $\bar{t}$).

For cold start problems, because there is no link information for a given node, we compare *2-features model*, *all-features model*, and *random by company*. For *all-features model*, we will only use the features available from the datasets. The model *random by company* indicates that we predict links between users in the same company by a uniformly random guess. The training/testing sets are generated differently for each dataset (see Section 5.1).

Each of the link prediction methods will output scores on all the node pairs. To evaluate the performance for each node i in the social network, we rank all other nodes j based on the score of pair (i, j), and select the top-k nodes without an edge connected to i in the training data as the top-k friend recommendations to user i. We then compute the average accuracy on these top-k recommendations based on the testing data. This measurement, which is directly related to the quality of top-k friend recommendations in online social networks, has been used in many recent papers [3, 27, 29]. We use "top-k accuracy" to denote this measurement throughout our experiments.

5.1 Experimental Setup

We consider three datasets: LinkedIn, Enron Email, and WikiTalk. The description of the datasets is as follows:

- **LinkedIn dataset**: For the cold start problem, we randomly sampled 20 companies, and formed a social network consisting of the past and current employees. We selected the date $2011/10/30$ to separate the training and testing data. We use all the connections before this time as training data (for link-based algorithm), and connections after it as testing data. (Note that we collected the data in Aug, 2012, so all the nodes and edges after this time are not present in this dataset.)

  For the warm start problem, we randomly sampled 2 separate networks, one for training and one for testing. Both training and testing datasets contained (past or current) employees from 5000 randomly-sampled companies.

- **Enron Email dataset**: A public dataset[2] containing the mailboxes of 150 Enron employees. We used the emails in the

[2]Downloaded from `http://www.cs.cmu.edu/~enron/`

inbox of these 150 Enron employees and generated a graph where edges corresponds to an email exchange of two employees. Because some of the inboxes are empty, this produced a social graph with 137 nodes and 1611 edges.

For the warm start problem, we consider all the links before September 2001 as training data, and all the links after it as testing data. The training data contains 529 edges and the testing data contains 1082 edges. For the cold start problem, we use a subset of users (20 users) as training data, and the rest of the social network as testing data.

Although this dataset does not have organizational time overlap information, we use the following heuristic to compute an *approximated overlap time* from this data: for each user, we can find his/her first and last email in the mailbox. Because it is the company Enron's mailbox, we use the first and last email in their mailbox as the joining time and leaving time for the user, and then compute the organizational time overlap by Equation (1).

- **Wiki talk dataset**: A public dataset[3] containing a collection of the edit history of Wikipedia talk pages. Because Wikipedia talk pages are discussions among Wikipedians, two users editing the same Wikipedia talk page can be considered linked. Based on the edit history, we generate a social graph with $1,038,443$ nodes and $488,283,928$ edges.

  Similar to the Enron Email dataset, we generate the *approximated overlap time* from Wiki talk dataset by identifying the first and last editing time for each user.

  For the cold start problem, we divide the edges into training/testing sets by the time before/after Jan 1, 2007 (the original data contains logs between $2003/01/06$ to $2008/02/06$). The resulting dataset has $199,058,354$ edges for training and $289,225,574$ edges for testing. For the code start problem, we use a subset with 1000 users for training, and the rest for testing.

5.2 Experimental Results

Table 1 shows the results on our three datasets with warm start. We can see that using only 2 features (time overlap and company size) without any link information can achieve competitive or even better top-k prediction compare to the two widely-used link prediction methods Common Neighbor (CN) and Adamic-Adar (AA). This result indicates that the organizational time overlap information described in this paper is useful for predicting future links, for example, 2-*features model* performs 42% better than CN and AA in terms of precision at top5 on LinkedIn dataset.

We further show the performance of our proposed method in the cold start setting. For the LinkedIn dataset, we extract all the features from LinkedIn data. For the other two public datasets, all users are in the same organization, thus the 2 *feature model* only has one feature – time overlap. Also, company propensity, company average age, and company cluster coefficients do not matter for Enron Email and Wiki Talk, so we only use the rest of the features for the *all feature model* on these two public datasets. The comparisons are shown in Figure 2. We can observe from Table 2 that the 2-*feature model*, which uses only organizational time overlap and company size, is dramatically improved over *random by company* (based on whether two users belong to the same company), for example, in terms of precision at top5, 2-*feature model* is more than 10 times better than *random by company* on LinkedIn dataset. Moreover, using more features can further improve the accuracy over 2 features. The results show that organizational time overlap is very important for link prediction tasks.

[3]Downloaded from the SNAP dataset collection `http://snap.stanford.edu/data/wiki-meta.html`.

Table 1: The experimental results for the warm-start problem

LinkedIn dataset					
	top5	top10	top20	top50	top100
2 features	**0.1207**	**0.1128**	**0.0893**	**0.0540**	**0.0313**
CN	0.0847	0.0912	0.0792	0.0528	0.0310
AA	0.0837	0.0905	0.0770	0.0524	0.0311
Enron Email dataset					
	top5	top10	top20	top30	top50
2 features	**0.3883**	0.2605	0.1580	0.1267	0.0880
CN	0.03118	0.2722	0.2220	0.1776	**0.1082**
AA	0.03215	**0.2833**	**0.2214**	**0.1812**	0.0958
Wiki Talk dataset					
	top5	top10	top20	top50	top100
2 features	**0.0819**	**0.0712**	**0.0495**	**0.0391**	**0.0241**
CN	0.0625	0.0551	0.0414	0.0361	0.0217
AA	0.0728	0.0654	0.0458	0.0365	0.0214

Table 2: The experimental results for the cold-start problem

LinkedIn dataset					
	top5	top10	top20	top50	top100
2 features	0.282	0.249	0.210	0.156	**0.122**
all features	**0.396**	**0.324**	**0.246**	**0.161**	0.112
random by company	0.026	0.025	0.022	0.018	0.012
Enron Email dataset					
	top5	top10	top20	top30	top50
2 features	0.55	0.35	0.21	0.17	**0.17**
all features	**0.61**	**0.38**	**0.22**	**0.18**	**0.17**
random by company	0.17	0.17	0.17	0.17	**0.17**
Wiki Talk dataset					
	top5	top10	top20	top50	top100
2 features	0.0912	0.0875	0.0755	0.0542	0.0433
all features	**0.1153**	**0.1041**	**0.0804**	**0.0626**	**0.0479**
random by company	0.0009	0.0009	0.0009	0.0009	0.0009

6. CONCLUSION

In this paper we proposed a novel model to compute probability of connection between two people based on organizational time overlap, and novel ways this could be applied to link prediction and community detection on a social network. We showed that this model is especially useful in cold start settings, which is very important for any online social network, and not much attention has been given to these settings in existing research. In the future we aim to utilize more node properties and combine that with graph properties for link prediction and community detection.

To obtain a better fit, we also considered models with success probability $\mu(1 - e^{-\lambda t_i^\gamma})$, where $0 < \mu \leq 1$ and $\gamma > 0$. The power function on overlap time moderates the edge probabilities, especially for extreme overlap values. As before, μ is a nugget to adjust for the fact that in larger companies a connection may not be made even after large time overlap. We fit two models using a MLE approach — (a) no-nugget model that fits (λ, γ) assuming $\mu = 1$, and (b) full model that fits all three parameters. The no-nugget model provided a fit that was similar to the full model. However, the no-nugget model seems promising and we are further exploring it.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] R. Andersen and K. Lang. Communities from seed sets. In *Proceedings of the 15th International Conference on World Wide Web*, 2006.

[3] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011.

[4] J. Chang and D. Blei. Relational topic models for document networks. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009.

[5] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(6), 2004.

[6] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Trans. on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.

[7] S. Fortunato. Community detection in graphs. *ArXiv:0906.0612*, 2009.

[8] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences of the United State of America*, volume 99, pages 7821–7826, 2002.

[9] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *Proceedings of SDM workshop on Link Analysis, Counterterrorism and Security*, 2006.

[10] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[11] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80, 2009.

[12] V. Leroy, B. B. Cambazoglu, and F. Bonchi. Cold start link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2010.

[13] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web*, 2008.

[14] J. Leskovec, K. J. Lang, and M. W. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

[15] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, 2003.

[16] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2010.

[17] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: joint models of topic and author community. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[18] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *Proceedings of the European Conference on Machine Learning*, 2011.

[19] K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. *Advances in Neural Information Processing Systems*, 22:1276–1284, 2010.

[20] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of USA*, 103(23): 8577–8582, 2006.

[21] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 2004.

[22] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[23] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson. Social topics models for community extraction. In *Proceedings of the 2nd SNA-KDD Workshop*, 2008.

[24] M. Sachan, D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st International Conference on World Wide Web*, 2012.

[25] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th International ACM SIGIR Conference*, 2012.

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[27] D. Shin, S. Si, and I. S. Dhillon. Multi-scale link prediction. In *Proceedings of the 21st ACM Conference on Information and Knowledge Management*, 2012.

[28] S. Soundarajan and J. Hopcroft. Using community information to improve the precision of link prediction methods. In *Proceedings of the 21st International Conference on World Wide Web*, 2012.

[29] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *Proceedings of the 7th IEEE International Conference on Data Mining*, 2007.

[30] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *CoRR*, abs/1110.5813, 2011.

[31] J. Yang and J. Leskovec. Structure and overlaps of communities in networks. *ArXiv:1205.6228*, 2012.

[32] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, 2012.