Understanding Task-Driven Information Flow in Collaborative Networks

Gengxin Miao[†] Shu Tao[‡] Winnie Cheng[‡] Randy Moulic[‡] Louise E. Moser[†] David Lo[♮] Xifeng Yan[♯]
[†]ECE Dept., University of California, Santa Barbara
[‡]IBM T.J. Watson Research Center
[♮]School of Information Systems, Singapore Management University
[♯]CS Dept., University of California, Santa Barbara
[†]{miao,moser}@ece.ucsb.edu, [‡]{shutao, wcheng, rmoulic}@us.ibm.com, [♮]davidlo@smu.edu.sg, [♯]xyan@cs.ucsb.edu

ABSTRACT

Collaborative networks are a special type of social network formed by members who collectively achieve specific goals, such as fixing software bugs and resolving customers' problems. In such networks, information flow among members is driven by the tasks assigned to the network, and by the expertise of its members to complete those tasks. In this work, we analyze real-life collaborative networks to understand their common characteristics and how information is routed in these networks. Our study shows that collaborative networks exhibit significantly different properties compared with other complex networks. Collaborative networks have truncated power-law node degree distributions and other organizational constraints. Furthermore, the number of steps along which information is routed follows a truncated power-law distribution. Based on these observations, we developed a network model that can generate synthetic collaborative networks subject to certain structure constraints. Moreover, we developed a routing model that emulates task-driven information routing conducted by human beings in a collaborative network. Together, these two models can be used to study the efficiency of information routing for different types of collaborative networks – a problem that is important in practice yet difficult to solve without the method proposed in this paper.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: Model and Principle—*Human information processing; Human*

General Terms

Algorithms, Human Factors

Keywords

Information Flow, Collaborative Networks, Social Routing

1. INTRODUCTION

Social networks as a means of communication have attracted much attention from both industry and academia. The studies so far have focused predominantly on public social networks, such as Facebook, Twitter, *etc.*, which support social interactions and information exchange among users. In this paper, we study another type of social network, *collaborative networks*, that are formed by members who collaborate with each other to achieve specific goals. Such collaborative networks often exist on the Web, such as open source software development sites, *e.g.*, Eclipse [2] and Mozilla [3] supported by Bugzilla [1], and in the private sector such as customer service centers [18].

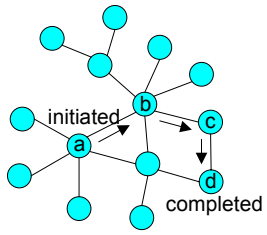

Figure 1: Task-driven information flow.

Information flow in collaborative networks is drastically different from that in public social networks [31]. In public social networks, information generated at a source spreads through the network with its members' forwarding activities [8, 13, 22, 25, 33]. The forwarding activities fade away as the information loses its value. In collaborative networks, information flow is driven by certain tasks. As illustrated in Figure 1, a task is initiated by or assigned to a source, and then routed through the network by its members until it reaches the person(s) who can handle it. The purpose of routing is to find the right person(s) for the task, not to influence others. The routing conducted by a member is based on (1) understanding of the expertise required to complete the task, and (2) awareness of other members' expertise. For example, in fixing software bugs, the bug report is the information routed in a developer network. If a developer cannot fix the bug, he/she will attempt to forward the bug report to another developer who he/she thinks is capable of fixing it. Table 1 shows one of the bug activity records extracted from the Eclipse development Web site.

The structure of collaborative networks usually evolves to facilitate the execution of tasks. It is desirable to determine whether the efficiency of the process can be improved. Ef-

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.

Table 1: Eclipse bug activity record.

Bug description: NullPointerException referencing non-existing plugins.		
Who	**When**	**Description**
dean	2001-11-01 07:17:38 EST	Added component Core. Reassigned.
rodrigo	2001-11-20 18:53:40 EST	Added component UI. Reassigned.
dejan	2002-01-09 20:46:27 EST	Converted the unresolved plugin to a link. Fixed.

`https://bugs.eclipse.org/bugs/show_activity.cgi?id=325`

ficiency can be measured by the number of steps it takes to navigate a task through a network to reach its resolver. For instance, a service provider might want to optimize the staffing structure of a call center, based on the expertise of its agents and the interactions between different agents. Such optimization might shorten the response time; however, it presents a challenge — one has to come up with recommendations without altering the network, an experiment that is not affordable in practice.

To address this challenge, we provide in this paper an understanding of how collaborative networks are structured, and how their structures affect the efficiency of task execution. More importantly, we present a simulation-based approach that allows various hypotheses to be tested with low cost. In general, a collaborative network can be characterized in terms of two aspects: (1) structure of the network, and (2) information routing driven by the tasks. Correspondingly, we develop the following models in this study.

- *Network Model:* A model that captures the key characteristics of a collaborative network and that can be used to simulate networks, given specific structural constraints.
- *Routing Model:* A model that simulates human behavior in routing task-related information in a collaborative network.

Models used to generate social networks have been studied extensively with substantial improvement in recent years, *e.g.*, [5, 9, 26, 29, 32]. In our problem setting, the network model must work consistently with the routing algorithm so that the routing length satisfies the distribution observed in real networks. This two-body modeling requirement is new and not easy to satisfy.

To develop these two models, we investigate three real-world collaborative networks collected from different sources. The first two were extracted from the Eclipse and Netbeans software development communities. The third one comes from an IT service management system, in which service agents collaborate to solve problems reported by customers. For all three networks, we analyze their structure, as well as information flows, using the routing history (*i.e.*, bug reports or problem tickets). We observe that collaborative networks exhibit not only the scale-free property in the node degree distribution, but also other organizational constraints. Furthermore, information routing in collaborative networks is different from routing tasks in conventional complex networks, such as IP packet routing in computer networks and itinerary planning in airline networks. The number of routing steps for each task follows a heavy-tailed distribution, indicating that a considerable number of tasks travel along long routes before reaching the resolvers. The three collaborative networks, collected independently from different sources, exhibit astonishingly similar characteristics, which validates the need to study them together. These observations contribute toward understanding the complicated behavior of human collaboration in these networks.

Based on our observations from real-world data, we develop a graph model to generate networks similar to real collaborative networks, and a stochastic routing algorithm to simulate the human dynamics of collaboration. The models are independently validated using real-world data and simulation-based studies. We demonstrate that the proposed models can be used to answer real-world questions, such as: *"How can one alter a collaborative network to achieve higher efficiency?"* To the best of our knowledge, our work is the first attempt to understand human dynamics in collaborative networks and to evaluate analytically the efficiency of real collaborative networks.

2. OBSERVATIONS

First, we illustrate the key characteristics of real-world collaborative networks and the information routing behavior in these networks. Our study is based on three datasets collected from two different domains: software development (public) and IT service center (private).

The Eclipse and Netbeans networks are extracted from the MSR 2011 Challenge [4], where each node represents a program developer. Both datasets contain a history of bug reports, user online interactions, and final resolutions. The Eclipse network has approximately $7,800$ developers who worked together on $272,000$ bugs. The Netbeans network contains around $156,000$ bug reports that involved $7,400$ developers. The third network, labeled "Enterprise network," is obtained from an IT service department, where each node represents a service agent. It contains around $2,000,000$ problem tickets submitted by customers. Similar to bug resolution in a programmer network, a ticket is transferred in a service agent network for resolution. The service network has around $19,000$ service agents. If one member in a collaboration network routes a bug report or a service ticket to another member in the network, we construct a directed edge. Thus, the three collaborative networks are represented by directed graphs.

Although developer networks and service agent networks seem to be quite different, we were amazed by the similarity exhibited in their network structures and dynamic routing structures, indicating that commonality exists in human collaboration behaviors.

2.1 Degree

Figure 2 shows the incoming and outgoing degree distributions of the three collaborative networks. Different from common observations in other complex networks like the Internet, the Web, and social networks, which exhibit the scale-free property, these collaborative networks have truncated power-law node degree distribution.

We tested the power-law hypothesis on the degree distributions of the collaborative networks using a principled statistical framework proposed by Clauset *et al.* [7]. The power-law model was not accurate enough to characterize the node degree distribution in collaborative networks after applying the p test [7]. However, we observed that the node degree of these networks follow a truncated power-law distribution (Eq. (1)) when the node degree k lies within a finite range. We applied a maximum likelihood approach, similar

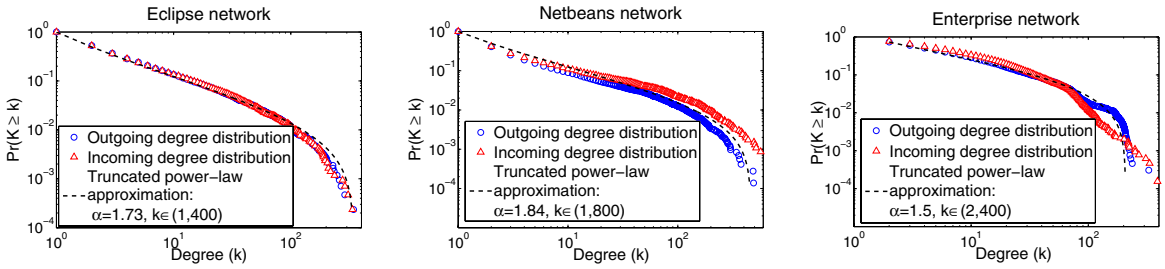

Figure 2: Degree distributions of collaborative networks.

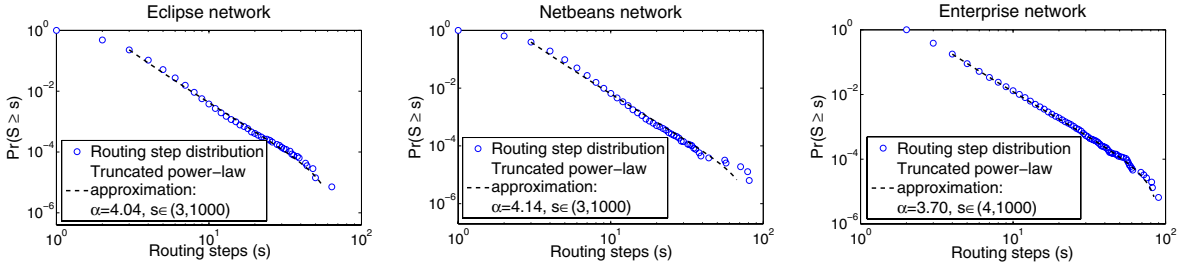

Figure 3: Routing steps distributions of problem solving in collaborative networks.

to [7], to fit the truncated power-law distribution. Inspired by [7], we further evaluated the goodness of fit using the p test based on the Kolmogorov-Smirnov statistic [20]. The truncated power-law model is a plausible fit to the node degrees because the statistical tests generate a value of p that is large enough ($p > 0.1$).

$$P(k) \propto k^{-\alpha} \text{ where } k \in (k_{min}, k_{max}) \quad (1)$$

The distributions in Figure 2 further differ from other complex networks in two aspects: (1) The power-law scaling parameter of the distribution falls in the range $\alpha \in (1, 2)$, in contrast to the commonly reported range $\alpha \in (2, 4)$, and (2) Both the incoming degree and the outgoing degree follow roughly the same power-law distribution.

The smaller value of the power-law scaling parameter indicates that, in a collaborative network, the probability $P(k)$ decreases more slowly as k increases. This distinctive property leads to the consequent effect that the node degrees are bounded. The distribution $P(k) \propto k^{-\alpha}$, where $\alpha \in (1, 2)$, does not have a converged mean $E(k) = \sum_{k=1}^{\infty} kP(k)$. However, in reality, the degrees of the nodes do have a mean value. This mismatch implies that the degree distribution is bounded: $P(k) \propto k^{-\alpha}$, where $k \in [k_{min}, k_{max}]$. The reason for this distinctive property is that human interactions in a collaborative network have more realistic constraints than those in an ordinary social network or the Web or other complex networks. In a collaborative problem solving environment, it takes a significant amount of time for a person to establish close interactions with other persons.

2.2 Routing Steps

The number of routing steps to complete a task is a critical measure of efficiency in collaborative networks. Figure 3 depicts the routing steps distribution for the three collaborative networks that we studied. The routing steps follow a truncated power-law distribution with a very similar scaling parameter $\alpha \in (3.5, 4.5)$ in all three collaborative networks. Unlike [29], which discovered that short paths exist between any pair of members in a collaborative network and that individual members are very adept at finding those short paths, the heavy-tailed distribution for routing steps indicates that a considerable proportion of tasks travel along long sequences before reaching a resolver. We conjecture that the heavy tails in these distributions are largely due to the varying complexities of the tasks assigned to the network. Namely, when a task is fairly complex and the expertise required to complete the task is concealed in the task description, the members in a collaborative network have to try different directions before the task is routed to the correct destination.

2.3 Clustering Coefficient

The clustering coefficient measures how closely the neighbors of a node are connected, by calculating the number of connected triplets in a network that are closed triplets. In an undirected graph, the local clustering coefficient of node i is defined as follows:

$$c_i = 2t_i/(k_i(k_i - 1)), \quad (2)$$

where k_i is the degree of node i, and t_i is the number of edges between i's neighbors. The global clustering coefficient is the average of the local clustering coefficients over all nodes in the network. To calculate the clustering coefficients in collaborative networks, we ignore the directions of edges. The clustering coefficients of the three networks studied are shown in Table 2. Note that the members in the enterprise network interact more closely in local teams than those in the public developer networks. This observation is not surprising, because enterprise networks typically have more rigid hierarchical structures.

Table 2: Clustering coefficients.

Eclipse network	Netbeans network	Enterprise network
0.19	0.21	0.35

3. NETWORK MODEL

As it is expensive, if not impossible, to alter real-world collaborative networks for hypothesis testing, *e.g.*, changing their structures for better performance, it is important

to develop a network model for which various hypotheses can be examined with low cost. The network model must take into account the structural constraints discussed in Section 2, *i.e.*, degree distribution and clustering coefficient. The network model must work consistently with the routing algorithm so that the routing steps satisfy the power-law distribution. This coupled modeling requirement is new and not easy to satisfy, especially when there is no way to generate simulated bugs or problem tickets. In this section, we present a network model for collaborative networks. In Section 4, we discuss the corresponding routing model.

In the network model, first we determine the location of each node in the network, which corresponds to a member's expertise. Next, we add edges between pairs of nodes, representing the interactions among members. Then, we tune the network model to capture the interactions among nodes with similar expertise, using the clustering coefficient.

3.1 Node Generation

To model a collaborative network with N nodes, first we randomly assign coordinates (x_i, y_i), where $x_i, y_i \in [0, L]$, to each node $i \in \{1, 2, ..., N\}$ in a two-dimensional rectangular area, simulating the *expertise space*.

The coordinates of a node represent the specific expertise of a network member. Thus, two members with similar expertise tend to be close to each other. Different collaborative networks can have different expertise distributions. To make the model general, we take a simplified representation of the expertise space and the node distribution. We assume that the nodes are uniformly distributed in the rectangular expertise space. That is, different expertise areas have the same representation in the generated nodes. However, this simplified representation in the general model can be substituted with specific network configurations of real collaborative networks. The routing algorithm that we introduce in Section 4 applies to these specific network configurations, as demonstrated by a direct embedding of real-world collaborative networks in two-dimensional space in Section 5.2.

Because the expertise space is limited to a rectangular area, nodes located at the center of the area are likely to have more neighbors than those located close to the boundary. To model the relationship between different expertise areas, we apply a periodic boundary condition that replicates the expertise area around the areas of interest, as shown in Figure 4. The distance $d_{i,j}$ between any pair of nodes i and j is defined as the minimum Euclidean distance between copies of i and j. In this way, each node is given a roughly equal-sized neighborhood.

3.2 Edge Generation

In a collaborative network, an edge from member i to member j exists when member i can transfer a task to member j. The establishment of an edge requires member j to expose his/her expertise sufficiently to the others, and member i to be aware of member j's exposed expertise. Only with these conditions will member i transfer a task to member j, when i believes j has the appropriate expertise to complete the task. Based on this intuition, we define two metrics for each node that guide edge generation in our network model: an expertise awareness coefficient and an expertise exposure coefficient.

For each node i in the network, its *expertise awareness coefficient* a_i and its *expertise exposure coefficient* e_i are

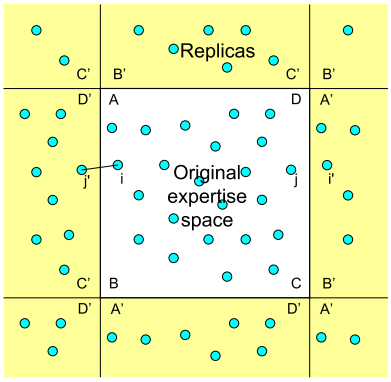

Figure 4: Periodic boundary condition in an expertise space.

random variables that follow probability distributions $a_i \sim P(a)$ and $e_i \sim P(e)$, respectively. An edge from node i to node j exists if and only if their awareness and exposure coefficients are large enough to cover the distance between i and j, *i.e.*, $a_i \times e_j > d_{i,j}$.

To simulate a network with certain incoming and outgoing node degree distributions, we need to tune the probabilities $P(a)$ and $P(e)$. Given that the incoming and outgoing degree distributions are identical in all collaborative networks studied in Section 2, we assume that the awareness and exposure coefficients have the same distribution. Therefore, if we know the form of one distribution, we can solve for the other symmetrically.

First, we assume that the distribution of the exposure coefficient is $P(e) = \beta \times e^{-\gamma}$, where $e \in [e_{min}, e_{max}]$. For any node i, when the awareness coefficient is chosen to be a_i, we calculate the probability that $edge_{i,j}$ exists, given the distance between node i and node j, as follows:

$$P(edge_{i,j}) = \begin{cases} 1 & d_{i,j} \le a_i \times e_{min} \\ P(e_j > d_{i,j}/a_i) & e_{min} < d_{i,j}/a_i \le e_{max} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Note that, when the nodes are uniformly distributed over the rectangular area, the node density ρ is a constant. Therefore, given the awareness coefficient a_i, we can estimate the outgoing degree $\widehat{k^i_{out}}$ of node i as

$$\begin{aligned} \widehat{k^i_{out}} &= \int_{d_0=0}^{\inf} \rho \times 2\pi d_0 P(edge_{i,j}) d(d_0) \\ &= \rho \times \pi (a_i e_{min})^2 \\ &+ \int_{e_0=e_{min}}^{e_{max}} \rho \times 2\pi a_i^2 e_0 P(e_j > e_0) d(e_0) \end{aligned} \tag{4}$$

Thus, $\widehat{k^i_{out}}$ can be expressed as ba_i^2, where b is a constant. To guarantee that the outgoing degrees of the nodes follow the desired power-law distribution $P(k_{out}) = c \times (k_{out})^{-\alpha}$, where $k_{out} \in [k_{min}, k_{max}]$, the awareness coefficient must

have the following probability distribution:

$$\begin{aligned} P(a) &= \lim_{\Delta a \to 0} \frac{P(a \le a_i \le a + \Delta a)}{\Delta a} \\ &= \lim_{\Delta a \to 0} \frac{P(ba^2 \le k_{out} \le b(a+\Delta a)^2)}{\Delta a} \\ &= \lim_{\Delta a \to 0} \frac{cb^{-\alpha+1}((a+\Delta a)^{-2\alpha+2} - a^{-2\alpha+2})}{(-\alpha+1)\Delta a} \\ &= 2cb^{-\alpha+1}a^{-2\alpha+1} \end{aligned} \quad (5)$$

That is, the awareness coefficient also follows a power-law distribution with coefficient $-2\alpha+1$. According to the symmetric assumption between the exposure and awareness coefficients, we conclude that the exposure coefficient follows the same power-law distribution with coefficient $-2\alpha+1$.

The range of the two coefficients should be set such that the degrees are restricted to the desired range. In Eq. (5), a node with minimum awareness coefficient a_{min} is expected to have the minimum outgoing degree k_{min}; a node with the maximum awareness coefficient a_{max} is expected to have the maximum outgoing degree k_{max}. Thus,

$$\begin{aligned} a_{min} = e_{min} &= \sqrt{\frac{k_{min}}{\rho \times \pi \langle e^2 \rangle}} \\ a_{max} = e_{max} &= \sqrt{\frac{k_{max}}{\rho \times 2\pi \langle e^2 \rangle}} \end{aligned} \quad (6)$$

where $\langle e^2 \rangle$ is the expected value of the squared exposure coefficient.

Given the power-law coefficient and the range of the awareness and exposure coefficients, their distributions are properly normalized. Using the normalized distributions, we generate edges in the network model with the probability given in Eq. (3), so that the incoming and outgoing degrees of the nodes follow the desired power-law distribution.

3.3 Modeling Expertise Domains

In a real collaborative network, the clustering coefficient indicates how closely its members work together in expertise domains. A higher clustering coefficient means that there are more collaborations between members within local expertise domains. To model collaborative networks with different expertise domains, the network model needs to form local teams of people that have specific expertise for certain tasks and that represent expertise domains. Intuitively, members with expertise in similar domains tend to interact more with each other when working on these tasks. Consequently, the network should have more links between nodes inside the same expertise domain, and fewer links between nodes in different or unrelated expertise domains. Even though it is less likely for members from unrelated expertise domains to interact with each other, such connections still exist in real collaborative networks and a member who reaches beyond his/her own expertise domain is usually one with high connectivity.

To model this behavior, first we associate nodes in the network with different domains. Then, for any two different domains, as illustrated in Figure 5, we break inter-domain links and replace them with intra-domain links, using an *edge swapping* process inspired by [28]. At each step of the edge swapping process, we choose a pair of inter-domain edges, pointing in opposite directions, and assign a swapping probability according to the degrees of the nodes to which they connect. If the connected nodes have high incoming or outgoing degrees, we swap the edges with low probabilities; otherwise, we swap the edges with high probabilities. Specifically, we consider two inter-domain edges $u_1 \to v_2$ and $u_2 \to v_1$, with users u_1 and v_1 from one domain, and users u_2 and v_2 from the other domain. We assign the edge swapping probability $p = 1 - \max(k_{out}^{u_1}, k_{in}^{v_2}, k_{out}^{u_2}, k_{in}^{v_1})/k_{max}$, where k_{max} is the maximum outgoing/incoming degree among all the nodes in the network. With probability p, we break the edges $u_1 \to v_2$ and $u_2 \to v_1$, and connect the edges $u_1 \to v_1$ and $u_2 \to v_2$. We repeat the edge swapping process until a certain fraction of the inter-domain edges have been swapped to intra-domain edges. The edge swapping process prefers to break inter-domain connections from nodes with low degrees and to maintain the edges for well-connected nodes. Thus, we avoid isolated subgraphs during the edge swapping process, and the resulting network matches real collaborative networks.

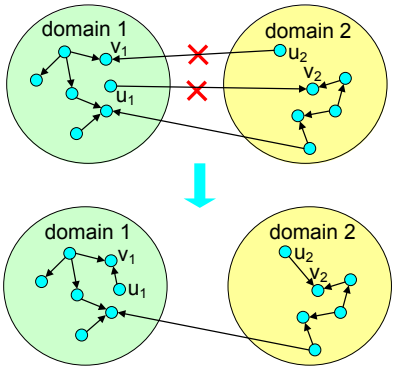

Figure 5: Inter-domains edge swapping.

With these adjustments, the node degree distribution will still fit the desired power-law distribution achieved in Section 3.2. The more edge swapping one performs, the higher the local connectivity the network has within each domain. The resulting networks have higher clustering coefficients.

For a network with a fixed number of nodes, as we increase the number of domains, the average size of a domain decreases. Consequently, the edge density inside each domain increases and the clustering coefficient increases. After forming local domains, the generated network has the desired incoming/outgoing degree distribution, and approximates the clustering coefficients of real collaborative networks.

4. ROUTING MODEL

The task-driven routing model must capture the behavior of humans in routing tasks to appropriate experts. Although the small-world phenomena [14, 29] is observed in collaborative networks, *i.e.*, a relatively short path typically exists between any pair of nodes in the three studied networks, there is no guarantee that the members in a collaborative network are able to route tasks through these short paths. In fact, our analysis in Section 2 has shown that the number of routing steps for a task typically follows a truncated power-law or heavy-tailed distribution. Consequently, a considerable number of tasks are routed along a long sequence of steps before they reach the resolvers. A commonly used

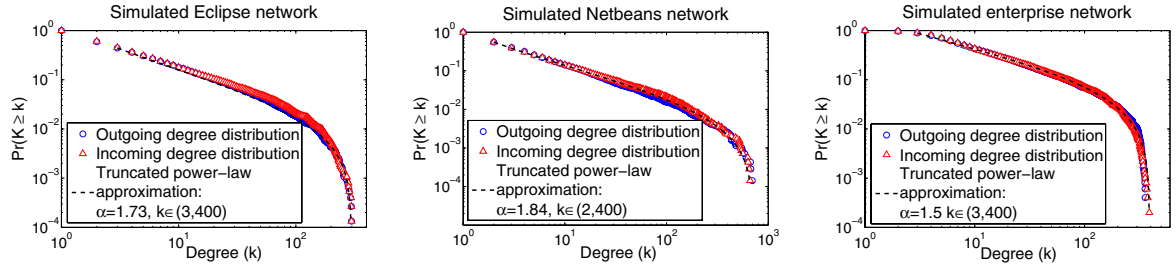

Figure 6: Degree distributions of simulated networks.

model in the Internet [6] and in social networks [14] is greedy routing. The greedy routing algorithm assumes that there exists a distance between any pair of nodes. In each routing thread, a node has access to the distance from itself and its neighbors to the destination node. If there exists one or more neighbors closer to the destination than the current node, it routes the task (packet) to the neighbor node closest to the destination. Otherwise, the node does not have a better routing choice than itself. In this case, the task (packet) fails to reach the destination.

Unfortunately, the greedy model is not adequate for simulating human task routing behavior. First of all, the greedy algorithm is deterministic, and often fails to navigate a task if the current task holder does not have a better choice. In the three networks we investigated, the greedy algorithm fails to route $\sim 14\%$ of the tasks. In contrast, most of these tasks have been successfully routed by humans. Secondly, the routing steps generated by the greedy algorithm follow an exponential distribution. As the number of routing steps increases, the probability drops much more quickly than the power-law distribution. In real decision-making scenarios, a human tends to make different routing decisions when the situations (*e.g.*, availability of neighbors, priority of tasks, *etc.*) are changing, even given similar tasks. Therefore, a more delicate model is needed to incorporate the stochastic process of task routing, which is essential for modeling human behavior.

In a collaborative network, people make their task routing decisions based on many factors, including the availability of neighbors, priority of tasks, *etc.* A member often makes a decision based on the local information available, rather than the global information that can be used to optimize the end-to-end routing efficiency. Thus, the same task can be transferred by a member along various non-optimal paths in different situations. Therefore, information routing in collaborative networks is a stochastic process, rather than a deterministic process.

We construct a Stochastic Greedy Routing (SGR) model based on the following intuition. When a member in a collaborative network cannot finish a task, he/she tends to transfer the task to a neighbor who has expertise closer to that of the resolver, similar to a greedy approach. The member also evaluates the connectivity of his/her neighbors, and tends to select a neighbor who has more outgoing connections, assuming that a better-connected neighbor is more likely to route the task along a shorter path to the resolver.

The SGR model assumes that each node relies on only local information to route tasks to one of its neighbors, following a stochastic process. Considering a task that is initially assigned to node u and has a resolver v, the SGR model guides each node to navigate the task through the network, from the initiator u to the resolver v. At each step, when a non-resolver node holds a task, it evaluates the candidate set $\mathcal{C}$, consisting of all its neighbors who have not yet been visited, and transfers the task to one of them. In some rare cases, the candidate set becomes empty and all the neighbors are marked as unvisited. As mentioned above, the task should be transferred to a node with closer expertise to that of the resolver and with a higher outgoing degree. Therefore, for each candidate i, we define the following utility function:

$$F(i) = d(i,v)^{-1} \times k_{out}^{i} \tag{7}$$

Note that this utility function is inversely proportional to $d(i,v)$, the geometric distance between a candidate and the resolver in our network model, which represents the similarity in their expertise. The holder of a task transfers the task to one of the candidates $i \in \mathcal{C}$ with a probability proportional to i's utility, *i.e.*, $P(i) = F(i)/\sum_{j\in\mathcal{C}} F(j)$. This process is repeated until the task reaches the resolver. The SGR model does not rely on the nature of the tasks to perform routing; thus, it avoids the issue of generating synthetic tasks. Instead, it needs only a pair of initiators and resolvers to simulate a task, which significantly simplifies the model.

The SGR model assumes that each node can evaluate the geometric distance between its neighbors and the resolver, without knowing the topology of the entire network. This assumption is very close to real-life situations. In our network model, geometric distances between nodes represent similarity in the expertise of the node. Although the current holder of a task does not know the shortest path to the resolver, he/she has knowledge of what expertise is required to complete the task, as well as the expertise of the neighbors. Hence, he/she can make a judgement as to which one of the neighbors is a better fit toward completing the task.

5. EVALUATION

In this section, we evaluate the network model and the routing model presented earlier. First, we evaluate the network model by comparing the key characteristics of the synthetic networks generated from this model and those of real collaborative networks. Then, we evaluate the effectiveness of the routing model by applying it to synthetic networks, as well as a direct two-dimensional representation of real collaborative networks. Finally, we demonstrate how to combine the two models to optimize the structure of collaborative networks in a case study.

5.1 Evaluating the Network Model

To evaluate the network model, first we use it to generate synthetic networks that have similar incoming and outgoing degree distributions as observed in real collaborative networks. For example, the Eclipse network has a power-law

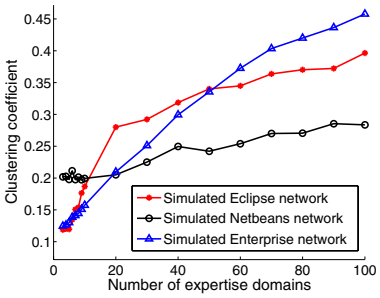

Figure 7: Tuning the clustering coefficient.

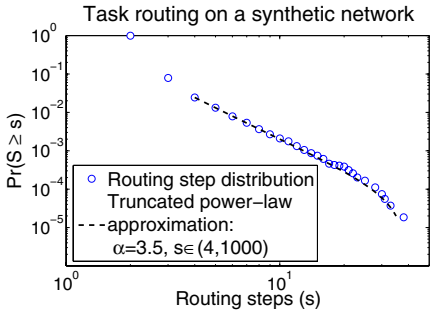

Figure 8: Routing steps distribution in a simulated enterprise network.

degree distribution $P(k) \sim k^{-1.73}$, where $k \in [1, 400]$. For each node in the synthetic network, we randomly select its awareness coefficient and exposure coefficient following the same power-law distribution $P(a) \sim a^{-2.92}$, $P(e) \sim e^{-2.92}$, where $a, e \in [0.047, 0.94]$ as calculated from Eqs.(5)-(6). Similarly, for simulating the Netbeans network, we calculate the power-law distribution for the awareness coefficient and the exposure coefficient as $P(a) \sim a^{-3.36}$, $P(e) \sim e^{-3.36}$, where $a, e \in [0.05, 1.6]$. For the Enterprise network, the awareness coefficient and the exposure coefficient follow the power-law distribution $P(a) \sim a^{-2}$, $P(e) \sim e^{-2}$, where $a, e \in [0.036, 0.72]$. Figure 6 shows that the degree distributions in synthetic networks are very close to those observed in the three real collaborative networks (*i.e.*, Eclipse, Netbeans, and Enterprise), shown in Figure 2.

Besides degree distributions, we need to evaluate the capability of our network model in generating networks with various clustering coefficients. Recall that the clustering coefficient of a collaborative network reflects the existence of expertise domains and the difference between inter- and intra-domain links. Here, we study the same three synthetic networks as shown in Figure 6. In each network, we divide the nodes into K expertise domains and then vary the clustering coefficient through edge swapping. As we vary the value of K, we expect different clustering coefficients. We select the one with the clustering coefficient closest to that of the real network as an approximation.

Figure 7 shows the variations of clustering coefficients of the synthetic networks for different values of K. By increasing the value of K, we observe that the clustering coefficient increases. Hence, by choosing a proper value of K, our network model can approximate a real collaborative network in both the degree distribution and the clustering coefficient. In our study, the Eclipse network is best approximated with 9 domains. The Netbeans network is best approximated with 10 domains. The Enterprise network is best approximated with about 60 expertise domains. We do not have information regarding the number of expertise domains in the Eclipse or Netbeans networks. However, we were able to confirm that the Enterprise network indeed had about 60 expertise domains.

It can also be observed in Figure 7 that, when the network has a power-law degree distribution with a large scaling parameter (*e.g.*, the Netbeans network), the clustering coefficient curve tends to be more flat than for the other networks. The reason is that, in such a network, most nodes have very few connections. Correspondingly, in our network model, most nodes have small awareness and exposure coefficients. Hence, the network is not very well-connected. After dividing the nodes into different domains, the edge swapping process can affect only a small number of cross-domain edges; otherwise, the network will become disconnected. As a result, increasing the value of K has a smaller effect on changing the network clustering coefficient.

5.2 Evaluating the Routing Model

To evaluate the routing model, first we ran task routing simulations guided by SGR on a synthetic network generated by the network model and we demonstrated that the result is consistent with real observations.

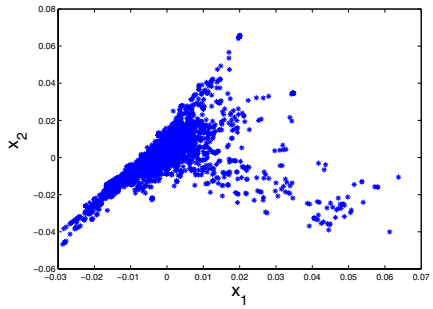

Figure 9: Two-dimensional spectral embedding of the Netbeans network.

We generated a collaborative network with 5,000 nodes to simulate the Enterprise network. The incoming/outgoing degree of the generated network follows a power-law distribution $P(k) \sim k^{-1.5}$, where $k \in [1, 400]$. We divided the network into 60 expertise domains, which leads to a clustering coefficient of 0.37. We generated a set of 100,000 tasks by choosing the initiators and the resolvers. For each task, we choose an initiator node with probability proportional to its outgoing degree, and a resolver node with probability proportional to its incoming degree. As shown in Figure 8, the resulting routing steps distribution again follows a power-law distribution. Its power law factor $\alpha = 3.5$ is very close to the real value $\alpha = 3.53$, which indicates that we can seamlessly combine the two models without inconsistency.

We further ran task routing directly on a two-dimensional representation of real networks to illustrate that it can stand alone for routing simulations. To map a real collaborative network into a two-dimensional space, while preserving the local neighborhood relationships, we adopt the spectral embedding method [23]. The embedding process guarantees that, if two nodes are close to each other in the original

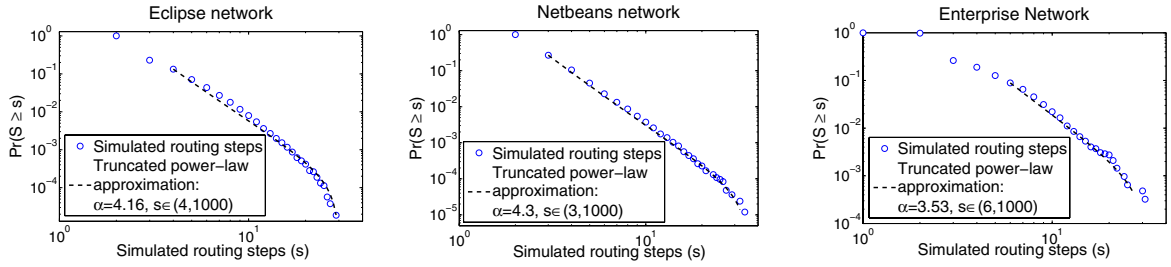

Figure 10: Simulated routing step distributions.

space, they are likely to be close to each other in the embedded space. The closeness between two nodes can be defined by the number of task transfers between them: the more frequent the task transfer, the closer the two nodes.

Figure 9 shows the two-dimensional embedding of the Netbeans network, using the spectral embedding method. The embedding can be regarded as a non-uniform distribution of nodes in an expertise space. Given the embedding, we can assign two-dimensional coordinates (x_1, x_2) to each node in the network, which enables distance measurement between pairs of nodes, a required input to the SGR model. Because we know the initiator and the resolver of each task, we then apply the SGR model to simulate the path of each task routing. The routing step distributions of the simulation for all three networks are shown in Figure 10. The simulated results match the observations well, as Figure 3 shows.

5.3 Combining the Two Models: A Case Study

Our network model simulates the static connectivity of a collaborative network, whereas our SGR model simulates the dynamic user behavior in information routing in a collaborative network. Combined together, these two models provide an unprecedented means of studying existing collaborative networks. It is important to study how the structure changes of a collaborative network can affect the efficiency of task execution, without changing the real-world network structure. This case study demonstrates the simulation method for our network and information routing models.

The case study is the problem management organization of a large IT service provider. To accommodate the evolving workload and human resources, the service provider needs to restructure the service agent network to deliver the optimal performance in resolving the problems reported by its clients. Currently, these restructuring decisions are made manually by experienced managers or consultants, without quantitative analysis as to how the resulting network will perform after the restructuring.

Our models can be used to provide analytical insights to the decision makers. First, one can use our network model to generate new network topologies with different structural constraints that need to be imposed in practice. Then, given a set of tasks, the efficiency of different networks can be evaluated through the task routing simulation guided by the SGR model. Here, we assume that a collaborative network of $5,000$ service agents needs to be restructured. These agents are divided into K pools (expertise domains) based on their expertise. A important question is: "How does one select the optimal number K of pools, to provide the best efficiency in task execution?" Intuitively, a smaller value of K indicates that the agents are more generalized in their

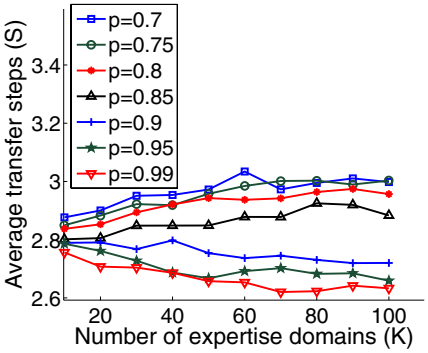

Figure 11: Evaluating the network structures.

domain expertise, whereas a larger value of K suggests that the agents are more specialized in their domain expertise. Furthermore, with more domains, a task is less likely to be initially assigned to the right agent pool, which might lead to longer routing paths, because intra-domain routing is more likely to occur than inter-domain routing.

For our analysis, we generate 10 collaborative networks, with 10 to 100 domains. In each network configuration, we simulate the routing of the same set of $100,000$ tasks. The probability p of correctly assigning the task to the right domain is also taken into account in the simulation. For each task, first we select the resolver node with probability proportional to its incoming degree. Then, with probability p, the task is initiated within the same domain as the resolver; otherwise, the initiator is selected from outside the resolver's domain. We vary the "correct assignment probability" p from 0.7 to 0.99. For each value of p, we route the entire set of tasks in the 10 networks. The results of all simulations are shown in Figure 11. The y-axis shows the average number of transfer steps to the resolver for the entire set of tasks. Each curve shows the routing simulation results for a particular choice of p. Obviously, a lower average number of steps indicates a higher routing efficiency, because it usually takes less time when the tasks are routed to the resolver in fewer steps. As shown in the figure, when more tasks are initially assigned to the right domain, increasing the number of domains leads to better performance. When fewer tasks are initially assigned to the right domain, a smaller number of domains is more favorable.

Achieving a certain value of p, given various numbers of agent pools, has different implications in terms of training the initial assigner of the task: for the same value of p, the training cost typically increases as the number of agent pools increases, because the assigner must have stronger knowl-

edge in matching the task with the correct expertise domain. Configuring the collaborative network into different numbers of expertise domains also has implications on the training cost for the agents. Given these implications, the decision maker can use our method to select the optimal number of agent pools that suits the enterprise's budget or other constraints.

6. RELATED WORK

Previous studies related to our work mainly belong to two categories: those that focus on network generation models, and those that analyze information flows in networks.

Network generation models. Generating synthetic networks that reflect statistics similar to real social networks has been of great interest to researchers in various fields. The Erdös-Rényi random network [9] is a classic random network, where any two nodes are connected according to a fixed probability. A regular lattice network is created with nodes placed on one or more dimensional lattices, *i.e.*, circle or grid, and each node is connected to its n nearest neighbors. Watts and Strogatz [32] added random rewiring to the regular lattice network such that the generated network has a small diameter as observed in a sample of the real social network [29]. Barabasi *et al.* [5] focused on the fact that many complex networks have degrees that follow a heavy-tail distribution and captured this phenomena by incrementally creating a random network, with new edges preferentially attached to already well-connected nodes. To comply with both the small-world effect and the power-law degree distribution, Makowiec [16] and Ree [21] proposed rewiring processes in a constant-size network based on the preferential attachment principle. Serrano *et al.* [26] developed a network generation model to reproduce self-similarity and scale invariance properties observed in real complex networks, by utilizing a hidden metric space with distance measurements. Sala *et al.* [24] studied how well the generated graphs match real social graphs extracted from Facebook.

Different from the existing graph generation models, our method contributes toward understanding how links are established and how members with different expertise interact with each other in real collaborative networks. Both the expertise awareness and expertise exposure of each member are taken into consideration in our model. It not only generates a network topology with statistical characteristics similar to real-world collaborative networks, but also can be seamlessly combined with our routing model to simulate human dynamics in these networks.

Information flow analysis. The spreading of information has been extensively studied under different network settings, *e.g.*, social networks, especially the World Wide Web, the e-mail network, biological networks, *etc.* Examples include the spread of innovations [12, 22, 27, 30], opinions, rumors and gossip [10, 11, 17], computer/biological viruses [15, 25] and marketing [8, 13]. More recently, Wang *et al.* [31] have studied how information propagates from person to person using e-mail forwarding, and Wu *et al.* [33] analyzed the information spreading pattern on Twitter. This type of information flow aims to reach and influence more people and, hence, to achieve a large impact. Most of the work has focused on analyzing patterns of the information spreading process. Kempe *et al.* [13] have addressed the question of how to choose a subset of nodes to initiate information spreading to maximize influence in a network.

In our work, we focus on another type of information flow: task-driven information flow, where the goal is to reach a user who can accomplish a task with a minimal number of transfer steps. Related to our problem, Milgram [19] demonstrated that short paths exist between any pair of nodes in a social network (*a.k.a.*, the small world phenomena). Kleinberg [14] investigated why decentralized navigation is efficient using a synthetic network lattice. Boguna *et al.* [6] studied the navigability of complex networks by running a greedy routing algorithm on synthetic networks generated by a model described in [26]. In the collaborative networks we studied, we observe that these networks exhibit degree distributions quite different from commonly-studied complex networks. Furthermore, the simple greedy algorithm does not provide a good approximation of information flow dynamics in collaborative networks. Thus, we developed the SGR model to evaluate the efficiency of task-driven information flow in such networks.

7. CONCLUSIONS

This study examined a special type of social networks – collaborative networks. Detailed observations of three real-world collaborative networks were presented along with the static network topology and dynamic information routing for each network. The collaborative networks exhibit not only the truncated power-law node degree distributions but also organizational constraints. Information routing in collaborative networks is different from routing in conventional complex networks, such as computer networks and airline networks, because of the random factors in human decision making. The routing steps in collaborative networks also follow a truncated power-law distribution, which implies that a considerable number of tasks travel along long sequences of steps before they are completed. Our results and observations for several independent sources are consistent with each other, and can be generalized to other real-world collaborative networks. They help in understanding the complicated behavior in human collaboration.

Based on real-world data, we developed a graph model to generate networks similar to real collaborative networks, and a stochastic routing algorithm to simulate the human dynamics of collaboration. The models are independently validated using real-world data. We demonstrated that the two models can be used to answer real-world questions, such as: *"How can one design a collaborative network to achieve higher efficiency?"* To the best of our knowledge, our work is the first attempt to understand human dynamics in collaborative networks and to estimate analytically the efficiency of real collaborative networks.

Acknowledgment

The first author, Gengxin Miao, was supported by an IBM Ph.D. Fellowship; she spent the Summer of 2011 at IBM T.J. Watson Research Center, where she conducted this research. This research was sponsored in part by the U.S. National Science Foundation under grant IIS-0917228 and by the Army Research Laboratory under cooperative agreement W911NF-09-2-0053 (NS-CTA). The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to

8. REFERENCES

[1] Bugzilla: http://www.bugzilla.org/.

[2] Eclipse: http://www.eclipse.org/.

[3] Mozilla: http://www.mozilla.org/.

[4] Msr 2011 challenge: http://2011.msrconf.org/msr-challenge.html.

[5] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[6] M. Boguna, D. Krioukov, and K. C. Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2008.

[7] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51:661–703, 2009.

[8] P. Domingos and M. Richardson. Mining the network value of customers. In *SIGKDD*, pages 57–66, 2001.

[9] P. Erdös and A. Rényi. On random graphs I. *Publicationes Mathematicae*, 6:290–297, 1959.

[10] S. Galam. Minority opinion spreading in random geometry. *The European Physical Journal B - Condensed Matter and Complex Systems*, 25(4):403–406, 2002.

[11] S. Galam. Modelling rumors: The no plane Pentagon French hoax case. *Physica A: Statistical Mechanics and Its Applications*, 320:571–580, 2003.

[12] X. Guardiola, A. Diaz-Guilera, C. J. Perez, A. Arenas, and M. Llas. Modeling diffusion of innovations in a social network. *Phys. Rev. E*, 66:026121, 2002.

[13] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146, 2003.

[14] J. Kleinberg. Small-world phenomena and the dynamics of information. In *NIPS*, page 2001. MIT Press, 2001.

[15] A. Lloyd and R. May. How viruses spread among computers and people. *Science*, 292(5520):1316–1317, 2001.

[16] D. Makowiec. Evolving network - simulation study. *The European Physical Journal B - Condensed Matter and Complex Systems*, 48:547–555, 2005.

[17] K. Malarz, Z. Szvetelszky, B. Szekf, and K. Kulakowski. Gossip in random networks. *ACTA Physica Polonica B*, 37, Nov. 2006.

[18] G. Miao, L. E. Moser, X. Yan, S. Tao, Y. Chen, and N. Anerousis. Generative models for ticket resolution in expert networks. In *KDD*, pages 733–742, 2010.

[19] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, Oct. 1992.

[21] S. Ree. Power-law distributions from additive preferential redistributions. *Phys. Rev. E*, 73:026115, Feb. 2006.

[22] E. M. Rogers. *Diffusion of Innovations*. Free Press, 4th edition, 1995.

[23] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[24] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao. Measurement-calibrated graph models for social network experiments. In *WWW*, pages 861–870, 2010.

[25] R. P. Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev.*, 86(14):3200–3203, 2001.

[26] M. Serrano, D. Krioukov, and M. Boguna. Self-similarity of complex networks and hidden metric spaces. *Phys. Rev.*, 078701, 2008.

[27] D. Strang and S. A. Soule. Diffusion in organizations and social movements: From hybrid corn to poison pills. *Annual Review of Sociology*, 24(1):265–290, 1998.

[28] R. Taylor. *Constrained switchings in graphs*. Research report. University of Melbourne, Department of Mathematics, 1980.

[29] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[30] T. W. Valente. Network models of the diffusion of innovations. *Computational and Mathematical Organization Theory*, 2:163–164, 1996.

[31] D. Wang, Z. Wen, H. Tong, C. Y. Lin, C. Song, and A. L. Barabási. Information spreading in context. In *WWW*, pages 735–744, 2011.

[32] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[33] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on Twitter. In *WWW*, pages 705–714, 2011.