SAND2012-8981P

A scalable directed graph model with reciprocal edges*

Nurcan Durak† Tamara G. Kolda‡ Ali Pinar§ C. Seshadhri¶

Abstract

A stronger relation exists between two nodes when they each point to one another (*reciprocal edge*) as compared to when only one points to the other (*one-way edge*). The proportion of reciprocal edges in a network is a good indicator of how tight the relations among the nodes are. The mutual relations could be an indication of friendship in a graph with social behavior or the information flow in a communication network. Despite their importance, reciprocal edges have been disregarded by most directed graph models. In our study, we propose a directed graph model that (i) combines the correct proportions of both reciprocal and one-way edges, (ii) matches the in-, out-, and reciprocal-degree distributions of the fitted graph, and (iii) requires only $O(m)$ work for a graph with m edges, making it scalable to very large graphs. We show the effectiveness of the proposed model on several real-world graphs and compare it to other state-of-the-art models.

1 Introduction

Is the connection between two nodes one-way or two-way (*reciprocal*)? We can infer a lot from the answer to this question. Assume we are given an e-mail exchange network. If two people are sending messages to each other, there is a real connection between them. However, if the message is only being sent in one direction, we cannot infer whether these two people know each other or not (e.g., sender could be a spammer). The high amount of reciprocal edges also is an indicator of a social behaviour in a network.

To study the impact of reciprocal edges, we categorize the directed edges into two types: *reciprocal* and *one-way* (see Figure 1). Formally, we say an edge (u, v) is *reciprocal* if the corresponding edge (v, u) also exists; otherwise, we say it is *one-way*.

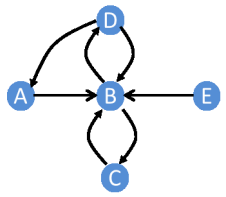

Figure 1: A directed graph with reciprocal (e.g., B-D) and one-way (e.g., D-A) edges.

The reciprocity, r, measures the density of reciprocal edges in a network [15] where r is the ratio of the number of reciprocal edges to the total number of edges. It can be interpreted as the probability of a random edge in a network being reciprocated. The reciprocity ratio is higher in the networks with social flavor (e.g., twitter, flickr, e-mail) whereas it is lower in information networks (e.g., web, news forums) or temporal networks (e.g., citation); see Table 1. It was observed that if a network has more reciprocal edges, viruses or news spread more quickly [15]. The reciprocity measure defined in [15] cannot distinguish networks with different densities (dense networks tend to have higher r values) and it can be exaggerated by self-links. The reciprocity measure can be improved by, for example, removing self-links and normalizing using the density [5].

In another interesting study [12], the formation order of the reciprocal edges is analyzed in some social interaction based networks such as Flickr (which has 68% reciprocal edges), and it is found that 83% of all reciprocal edges are created within 48 hours after the initial edge creation. Twitter is also analyzed in terms of reciprocity and found that 22.1% of the edges are reciprocal [7], which shows that twitter is the mixture of social and information accessing network.

Our interest is in realistic generative models that capture, at a minimum, the degree distributions of real-world networks. It is important to study networks to understand formation behaviours, detect abnormalities, and increase robustness. However, real networks

*This work was funded by the DARPA Graph-theoretic Research in Algorithms and the Phenomenology of Social Networks (GRAPHS) program and by the DOE Complex Interconnected Distributed Systems (CIDS) Program. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

†Sandia National Laboratories, CA, nurcan.durak@gmail.com
‡Sandia National Laboratories, CA, tgkolda@sandia.gov
§Sandia National Laboratories, CA, apinar@sandia.gov
¶Sandia National Laboratories, CA, scomand@sandia.gov

may not be released due to privacy and security issues. Therefore, random network generators reproducing salient features of real networks are required.

One of the most common salient features among networks is that degree distributions of many real networks is heavy-tailed [2]. Existing directed graph models such as Stochastic Kronecker Graphs (SKG) [9, 8] and Forest Fire (FF) [10] roughly match the given in-degree and out-degree distributions. Moreover, these and many other directed graph models do not generate any or many reciprocal edges.

Another concern in random graph generation is scalability. As the sizes of networks get larger and larger, generating random graphs in a scalable manner becomes crucial. Many graph generators add edges incrementally, creating new edges based on previous insertions. Another drawback is finding the right parameters of the models (e.g., SKG, FF, etc.) is very time intensive for large scale graphs. Without costly parameter estimation steps, we need to generate realistic directed graphs with reciprocal edges in a scalable manner.

1.1 Contributions

- Reciprocal edges represent social exchanges in the networks; however, most generative models specifically designed for social networks are missing this crucial behavior. We propose the *Fast Reciprocal Directed* (FRD) graph generator which explicitly matches the reciprocal degree distribution as well as the in- and out- degree distributions.
- A critical component of the FRD is a fast model for generating a directed graph without reciprocal edges. For that purpose, we propose the *Fast Directed* (FD) graph generator, which is a close cousin of the Chung-Lu [3] and edge configuration models [14, 1].
- Neither FRD nor FD require any "model fitting" parameters beyond the target degree distributions.
- Both of our proposed models are *fast*, generating m edges in $O(m)$ time for a constant maximum degree. Our models take less than a minute to generate a graph with multi-million nodes and edges, faster than any comparable models.
- We also explain why the number of degree-1 nodes is much lower than intended in Chung-Lu like models [3, 17] and propose a solution to obtain a better match for the degree-1 vertices.

2 Related Work

In this section, we consider existing directed generative graph models. Most previous models suffer from some combination of the following problems: few or no reciprocal edges, unable to match various degree distributions precisely, lack of scalability in fitting and/or generation (most models require some "history" to pick the next set of edges).

Kleinberg et al. [6] propose the Edge Copying (EC) model for web networks based on the observation that web network has topic-based clusters. In the EC model, when a new node arrives, it selects a random vertex v and copies a specified number of links k of vertex v [6]. The EC model has no mechanism for reciprocal edges since new nodes always point to older nodes. Leskovec et al. proposes the Forest Fire (FF) model [10] in which a new node can connect both to the vertices pointing to vertex v or to the vertices that vertex v points to. The edge creation continues with the neighbors of the neighbors of v, in other words fire spreads in a region. The FF model has forward p_f and backward p_b burning probabilities which are used to specify the density(seriousness) of the fire region. FF is a state-of-the-art model and will be used in our comparative studies. Like EC, FF model cannot create reverse (reciprocal) links between two nodes. Also, both methods are serial in nature because each new set of links for a new vertex depends on the graph that has been created thus far. To fit FF to a real graph, the forward p_f and backward p_b burning parameters must be adjusted to match the number of the edges in the fitted graph. In large scale networks, each fitting attempt with different burning parameters is very expensive. In practice, the required time of the FF fitting is (number of attempts to find the paramaters) $\times$ (FF graph generation time), but we only report the generation time (after fitting is complete) in our studies.

Unlike the EC model and FF model, the Stochastic Kronecker Graph (SKG) model [9, 8] is a scalable model. It is also considered to be state-of-the-art and is used for comparison in our experiments. The SKG model begins with an initiator matrix (typically 2×2) and produces larger graphs by recursive Kronecker product much faster than incremental methods. For large scale graphs, computing the initiator matrix is very time intensive [8]. In fact, we were unable to compute the initiator matrix for the larger graphs in a reasonable time frame; therefore for the remainder, we use the initiator matrices that have been previously reported in the literature [8, 19]. As with FF, we only report the generation time for the graphs.

In another directed model, the growth of Wikipedia is imitated and reciprocal edges are partly supported [21, 20]. This model extends the well-known preferential attachment (PA) model [2] by including reciprocity measure r. Each node arrives at a time and connects to k sink vertices, and then reciprocal links

are formed from those sink vertices to the newly added node with the probability of reciprocity, r. They test their model only for Wikipedia network and show that it matches the in-degree distribution well; however, it deviates sharply from the real out-degree distribution. Like EC and FF, this model is not scalable.

The work of this paper is closely related to the Chung-Lu (CL) model [3], which generates an undirected graph whose degree distribution matches to the given degree distribution. The CL model creates an edge between v_i and v_j proportional to the product of their degrees. In the CL model, each edge creation is done by independent coin flips and therefore, it is not suitable for scaling. A "fast" CL model that behaves like SKG model in terms of several network measures was introduced in [16]. Another model is the Edge Configuration (EdgeCon) model [14, 1] creates a set containing d_i copies of each vertex v_i and then chooses random matching of the elements in the set to create edges.

3 Proposed Directed Graph Models

In this study, we propose two scalable directed graph models: the *Fast Directed* (FD) model which generates a directed graph $G = (V, E)$ with respect to the given in- and out-degree distributions, and the *Fast Reciprocal Directed* (FRD) which explicitly accounts for reciprocal edges as well.

Before going into the details, we present the notation. Given a directed graph G, let n be the number of nodes and m be the number of directed edges. For instance, in Figure 1, $n = 5$ and $m = 7$. We divide the edges into three types:

- $d_i^{\leftrightarrow}$ = reciprocal degree (each reciprocal edge corresponds to a *pair* of directed edges),
- $d_i^{\leftarrow}$ = in-degree (excluding reciprocal edges), and
- $d_i^{\rightarrow}$ = out-degree (excluding reciprocal edges).

We also define the *total* in- and out- degrees, which include the reciprocal edges, i.e.,

- $d_i^{\Leftarrow} = d_i^{\leftarrow} + d_i^{\leftrightarrow}$ = total in-degree, and
- $d_i^{\Rightarrow} = d_i^{\rightarrow} + d_i^{\leftrightarrow}$ = total out-degree.

Most directed graph models consider only the total in- and out-degrees, ignoring reciprocity. As an example of these measures, node B in Figure 1 has $d_B^{\leftrightarrow} = 2$, $d_B^{\Leftarrow} = 2$, $d_B^{\rightarrow} = 0$, $d_B^{\Leftarrow} = 4$, and $d_B^{\Rightarrow} = 2$.

We may also assemble corresponding degree distributions, as follows. For any $d = 0, 1, \ldots,$ define

- $n_d^{\leftrightarrow}$ = Number of nodes with reciprocal-degree d,
- $n_d^{\leftarrow}$ = Number of nodes with in-degree d,
- $n_d^{\rightarrow}$ = Number of nodes with out-degree d,
- $n_d^{\Leftarrow}$ = Number of nodes with total-in-degree d, and
- $n_d^{\Rightarrow}$ = Number of nodes with total-out-degree d.

Let $d_{\max}$ be the maximum of all possible degrees. Then we can express n and m as

$$n = \sum_{d=0}^{d_{\max}} n_d^{\leftarrow} = \sum_{d=0}^{d_{\max}} n_d^{\rightarrow} = \sum_{d=0}^{d_{\max}} n_d^{\leftrightarrow},$$

$$m = \sum_{d=1}^{d_{\max}} d \cdot n_d^{\leftarrow} + d \cdot n_d^{\leftrightarrow} = \sum_{d=1}^{d_{\max}} d \cdot n_d^{\rightarrow} + d \cdot n_d^{\leftrightarrow}.$$

The reciprocity ratio of a graph [15] is

$$r = \frac{\#\text{ reciprocated edges}}{\#\text{ edges}} = \frac{\sum_{d=1}^{d_{\max}} d \cdot n_d^{\leftrightarrow}}{m}.$$

3.1 The Fast Directed Graph Model

In this model, we consider only the total in- and out-degrees, ignoring reciprocity.

To generate the *Fast Directed* (FD) model, we extend the Fast Chung-Lu (FCL) model for undirected graphs [17]. This model is based on the idea that each edge creation can be done independently if the degree distribution is given. The FCL reduces the complexity of the CL model from $O(n^2)$ to $O(m)$, and the same can be done in the directed case.

In the Chung-Lu model [3], after m insertions (and assuming $d_i^{\Rightarrow} d_j^{\Leftarrow} < m$ for all i, j) the probability of edge (i, j) is

$$p_{ij} = \frac{d_i^{\Rightarrow} d_j^{\Leftarrow}}{m}.$$

The naive approach flips a coin for each edge independently. The "fast" approach flips a coin to pick each endpoint. The probability of picking node i as the source is proportional to $d_i^{\Rightarrow}$ and the probability of picking node j as the destination is proportional to $d_j^{\Leftarrow}$.

Our implementation works as described in Alg. 1. We first pick all the source nodes and then all the sink nodes using the weighted vertex selection described in Alg. 2. If we want 500 nodes of out-degree of 2, for example, we create a "degree-2 pool" of 500 vertices and pick from it a total of 1000 times in expectation by doing weighted sampling of the pools. Within the pool, we pick a vertex uniformly at random with the further expectation that each vertex in the pool will be picked 2 times on average. In Alg. 2, the pool of degree-d vertices is denoted by $\mathcal{P}_d$ and the likelihood that the dth pool is selected is denoted by w_d. In all cases except $d = 1$, the size of the pool is defined by the number of vertices of that degree and the weight of the pool is the number of edges that should be in that pool. The one exception is the degree-1 pool which has a *blowup* factor b. For now, assume $b = 1$; we explain its importance further on in §3.3. At the end of Alg. 2, we randomly relabel the vertices so there is no correlation between the degree and vertex identifier.

The FD method can produce repeat edges, unlike the naive version that flips n^2 weighted coins (one per edge). Nevertheless, this has not been a major problem in our experience. Another alternative to Alg. 2 is to put d copies of each degree-d vertex into a long array and then randomly permute it—this is the approach of the *edge configuration* model. This gives the *exact* specified degree distribution (excepting possible repeats) by using a random permutation of a length m_* array. This would produce very similar results to what we show here, and is certainly a viable alternative. We also mention an alternate way of generating Chung-Lu graphs that could be adapted for the directed case [11].

Algorithm 1 Fast Directed Graph Model

procedure FDModel(G,$b^{\Leftarrow}$,$b^{\Rightarrow}$)
  Calculate $\{ n_d^{\Leftarrow} \}$ and $\{ n_d^{\Rightarrow} \}$ for G
  $\{ i_k \} \leftarrow$ VertexSelect($\{ n_d^{\Rightarrow} \}, b^{\Rightarrow}$)
  $\{ j_k \} \leftarrow$ VertexSelect($\{ n_d^{\Leftarrow} \}, b^{\Leftarrow}$)
  $E \leftarrow \{ (i_k, j_k) \}$
  Remove self-links and duplicates from E
  return E
end procedure

Algorithm 2 Weighted Vertex Selection

procedure VertexSelect($\{ n_d \}$, b)
  $n \leftarrow \sum_{d=0}^{d_{\max}} n_d$
  $n_* \leftarrow b \cdot n_1 + \sum_{d=2}^{d_{\max}} n_d$
  $m \leftarrow \sum_{d=1}^{d_{\max}} d \cdot n_d$
  $\mathcal{P} = \{ 1, \ldots, n_* \}$
  for all $d = 1, \ldots, d_{\max}$ **do**
    $w_d \leftarrow d \cdot n_d / m$
    if $d > 1$ **then**
      $\mathcal{P}_d \leftarrow n_d$ vertices from $\mathcal{P}$
    else
      $\mathcal{P}_1 \leftarrow b \cdot n_1$ vertices from $\mathcal{P}$
    end if
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{P}_d$
  end for
  for all $k = 1, \ldots, m$ **do**
    $\hat{d}_k \leftarrow$ Random degree in $\{ 1, \ldots, d_{\max} \}$,
      proportional to weights $\{ w_d \}$
    $i_k \leftarrow$ Uniform random vertex in $\mathcal{P}_{\hat{d}_k}$
  end for
  $\mathcal{P} \leftarrow$ unique indices in $\{ i_k \}_{k=1}^m$
  $\pi \leftarrow$ Random mapping from $\mathcal{P}$ to $\{ 1, \ldots, n \}$
  return $\{ \pi(i_k) \}_{k=1}^m$
end procedure

3.2 The Fast-Reciprocal Directed graph model

The FD model generates a directed graph and matches to the total in- and out-degree distributions. However, it produces virtually no reciprocal edges. To overcome this issue, we propose the *Fast Reciprocal Directed* (FRD) graph model.

Here, the goal is quite simple: capture the reciprocal edges using an undirected model and the remaining directed edges using a directed model. After that, we blend the generated edges from each model in one model. In this case, we explicitly consider the three distributions, $\{ n_d^{\leftrightarrow} \}$, $\{ n_d^{\leftarrow} \}$, and $\{ n_d^{\rightarrow} \}$. The method is presented in Alg. 3.

Algorithm 3 Fast Reciprocal Directed Graph Model

procedure FDModel(G,$b^{\leftrightarrow}$, $b^{\leftarrow}$,$b^{\rightarrow}$)
  Calculate $\{ n_d^{\leftrightarrow} \}$, $\{ n_d^{\leftarrow} \}$, and $\{ n_d^{\rightarrow} \}$ for G
  $\{ i_k \} \leftarrow$ VertexSelect($\{ \frac{1}{2} n_d^{\leftrightarrow} \}, b^{\leftrightarrow}$)
  $\{ j_k \} \leftarrow$ VertexSelect($\{ \frac{1}{2} n_d^{\leftrightarrow} \}, b^{\leftrightarrow}$)
  $E_1 \leftarrow \{ (i_k, j_k), (j_k, i_k) \}$
  $\{ i_l \} \leftarrow$ VertexSelect($\{ n_d^{\rightarrow} \}, b^{\rightarrow}$)
  $\{ j_l \} \leftarrow$ VertexSelect($\{ n_d^{\leftarrow} \}, b^{\leftarrow}$)
  $E_2 \leftarrow \{ (i_l, j_l) \}$
  $E \leftarrow E_1 \cup E_2$
  Remove self-links and duplicates from E
  return E
end procedure

3.3 Fixing the Number of Degree-1 Nodes

Below, we present our arguments for the case of the in-degree, but the same arguments applied to out-degree or reciprocal degree (with slightly more complexity in the reciprocal case which is omitted due to space). We use just the notation d to denote the in-degree, for simplicity.

If we run VertexSelect (Alg. 2) repeatedly, always assigning the same ids to each vertex pool and omitting the random relabeling (π) at the end, each node will get its desired in-degree *on average across multiple runs*. For any single run, however, this will not be the case. In fact, the degrees are Poisson distributed.

Claim 3.1. *The probability that a vertex v in pool $\mathcal{P}_d$ is selected x times is*

$$\text{Prob}\{ v \text{ selected } x \text{ times} \mid v \in \mathcal{P}_d \} = \frac{d^x e^{-d}}{x!}.$$

This claim is easy to see. We expect that pool $\mathcal{P}_d$ will be selected $w_d = d \cdot n_d$ times. Therefore, each element of $\mathcal{P}_d$ will be selected an average of d times, so that is the Poisson parameter. (There may be some small variance in the number of times that each pool is

selected, but the variance should be small enough not to greatly impact the average degree.)

The effect of the Poisson distribution is particularly noticeable in the pool of degree-1 nodes where the probability that a node in $\mathcal{P}_1$ has in-degree $x = 1$ is only 36%. An additional 36% will have an in-degree of $x = 0$ and the remaining 28% will an in-degree of $x \geq 2$. Of course, there will be some contributions from the other pools, e.g., $\mathcal{P}_2$ will produce 27% degree-1 nodes. However, in a power law degree distribution, $n_2 \ll n_1$ so its contribution is small. Nevertheless, we can calculate the expected number of degree-x nodes by summing over the contributions across all degrees pools.

CLAIM 3.2. *Let n'_x denotes the number of nodes that are selected exactly x times. Then*

$$\mathbb{E}(n'_x) = \sum_d n_d \frac{d^x e^{-d}}{x!}.$$

Again, the claim is easy to see and so the proof is omitted.

For many real-world distributions, $n'_1 \ll n_1$. We propose a workaround to this problem — we would like to reduce the number of nodes in $\mathcal{P}_1$ that are selected multiple times. To do this, we increase the size of the pool via a *blowup factor* b, which is used as follows. Let $\mathcal{P}_1$ contain $b \cdot n_1$ nodes. The weight of the pool will not change, meaning that it will still be selected n_1 times. Therefore, we may make the following claim.

CLAIM 3.3. *The probability that a vertex v in pool $\mathcal{P}_1$ with $b \cdot n_1$ elements is selected x times is*

$$\text{Prob}\{\, v \text{ selected } x \text{ times} \mid v \in \mathcal{P}_1 \,\} = e^{-1/b}/(b^x \cdot x!).$$

Furthermore, the expected number of nodes in $\mathcal{P}_1$ that are selected exactly one time is $n_1 \cdot e^{-1/b}$. Hence, letting n'_x denote the number of nodes that are selected exactly x times, we have

$$\mathbb{E}(n'_x) = n_1 \cdot \frac{e^{-1/b}}{b^{x-1} \cdot x!} + \sum_{d>1} n_d \frac{d^x e^{-d}}{x!}.$$

Proof. We still pick pool $\mathcal{P}_1$ a total of n_1 times, so that average (i.e., the Poisson parameter) for this pool is now reduced to $n_1/(n_1 \cdot b) = 1/b$ since there are $b \cdot n_1$ elements.

The next equation comes from the fact that there are $b \cdot n_1$ nodes in the pool, so we multiply the number of nodes with the probability of being picked x times with $x = 1$ to determine the expected number.

Finally, the revised expectation comes from changing the formula for the first pool to account for the enlarged pool size.

If we choose, for example, $b = 10$, then we can expect that $0.9 \cdot n_1$ nodes in $\mathcal{P}_1$ to be selected exactly one time. We show an example of the impact of this modification in Figure 2, where we show the total in-degree for soc-Epinions1 with and without a blowup factor of $b = 10$. The degrees are logarithmicly binned and summed. Note that the match for the number of degree-1 nodes is improved, but there is small penalty in the match for degree-2 nodes. We use $b = 10$ in all experiments reported in this paper.

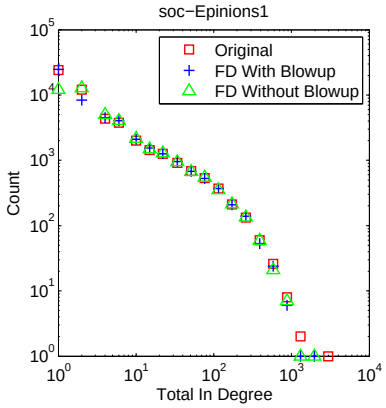

Figure 2: Example of in-degree distribution with and without blowup factor. Note that the model with the blow-up factor matches degree-1 nodes precisely, however, the model without blow-up generates only half of the degree-1 nodes in the original graph.

4 Experimental Studies

We test our models on various directed networks such as citation (cit-HepPh), web (web-NotreDame), and social (soc-Epinions1, soc-LiveJournal) [22]. We also test our models on large scale graphs coming from online social networks (youtube, flickr, liveJournal) [13]. We list the attributes of the networks in Table 1 after removing self-links and making the graph unweighted (simple). Note that reciprocity, r, is very low in the citation network. We elaborate how we fit the models to the real networks below.

Fast Directed (FD) and Fast Reciprocal Directed (FRD) Our proposed models work directly with the appropriate degree distributions of the input graphs. We used a blowup factor of $b = 10$ in all cases.

Forest Fire (FF) We provide the number of nodes n, and the forward and backward burning probabilities p_f and p_b to the SNAP software [22]. To fit FF, we match the generated graph models to the number of edges in the real networks. For each target graph, we search a range of values by incrementing p_f value

Table 1: Networks used in this study. The value of r is the reciprocity measure, p_f is the forward burning parameter for FF, and the last column is the SKG initiator matrix.

Graph Name	Nodes	Edges	Rec. Edges	r	p_f	SKG initiator
cit-HepPh [22]	34K	421K	<1K	0.003	0.37	[0.990,0.440;0.347,0.538] [8]
soc-Epinions1 [22]	76K	508K	206K	0.405	0.346	[0.999,0.532;0.480,0.129] [8]
web-NotreDame [22]	325K	1,469K	759K	0.517	0.355	[0.999,0.414;0.453,0.229] [8]
soc-LiveJournal [22]	4,847K	68,475K	32,434K	0.632	0.358	[0.896,0.597;0.597,0.099] [19]
youtube [13]	1,157K	4,945K	3,909K	0.791	0.335	—
flickr [13]	1,861K	22,613K	14,117K	0.624	0.355	—
LiveJournal [13]	5,284K	77,402K	56,920K	0.735	0.355	—

by $\delta p = 0.001$ in range [0.2-0.5] to find the best model giving the similar number of edges to the original network; the values we use are reported in Table 1. We set $p_b = 0.32$ as described in [10].

Stochastic Kronecker Graphs (SKG) We use the initiator matrices reported by previous studies: [8] for cit-HepPh, soc-Epinions, and web-NotreDame and [19] for soc-LiveJournal. We attempted to generate initiator matrices for large graphs using [22], but the program did not terminate within twenty-four hours. Therefore, we only fit SKG to the networks obtained from SNAP[22] data warehouse. We set the size of the final adjacency matrix is $2^{\lceil \log_2(n) \rceil}$, where n is the number of nodes in the real graph.

We generate all the models in a Linux machine with 12GB memory and Intel Xeon 2.7 Ghz processor. The FD and FRD methods were implemented by us in MATLAB; the SKG and Forest Fire generation code were implemented in C++ from [22]. For fair comparison, we do not time the printing or file saving parts from the Snap software. Graph generation time for each model is listed in Table 2. Among all of the results, FD and FRD are the fastest, in that order. SKG is little bit slower than both FD and FRD models. The forest fire is the slowest even though C++ is much faster than MATLAB code.

Table 2: Graph generation time

Graph Name	SKG	FD	FRD	FF
cit-HepPh	2.17s	0.16s	0.19s	18.80s
soc-Epinions	1.53s	0.29s	0.41s	6.73s
web-NotreDame	4.95s	0.56s	0.62s	29.66s
soc-LiveJournal	6m51s	31.15s	41.75	2h28m32s
youtube	—	2.16s	2.53s	2m22s
flickr	—	10.30s	12.20s	1h11m2s
liveJournal	—	35.30s	59.98s	8h30m18s

We analyze the number of reciprocal edges generated by each model in Table 3. The FF model cannot generate any reciprocal edges. The FD model can generate a few random reciprocal edges but this number is negligible compared to the real number of reciprocal edges. The SKG model generates some reciprocal edges; however, it is also much less than the real number. The FRD model performs the best and generate correct amount of reciprocal edges.

Table 3: Reciprocal Edges created by each model

Graph Name	Orig.	SKG	FD	FRD	FF
cit-HepPh	1071	1160	159	1148	0
soc-Epinions1	31K	835	86	30K	0
web-NotreDame	89K	5K	27	85K	0
soc-LiveJournal	1.5M	14K	171	1.5M	0
youtube	526K	—	18	499K	0
flickr	1.3M	—	205	1.3M	0
liveJournal	4.1M	—	258	4.0M	0

We also analyze the generated degree distributions by each model. The plot are log-binned for each of readability. Figure 3 shows the results on the soc-Epinions1 graph. Here we see that all four methods do fairly well in terms of matching the total in- and out-degree distributions. (The few low values for SKG are due to its well-known cycling behavior [18].) However, only the FRD method matches the reciprocal degree distribution. The FD and SKG methods produce far too few reciprocal edges and FF does not produce any. We see very similar behavior in Figure 4 for soc-LiveJournal, except here the FF and SKG degree distributions do not match the total out-degree distribution very well. Once again, neither FD nor SKG produce many reciprocal edges and FF does not produce any. Figures for cit-HepPh and web-NotreDame are shown in the appendix.

For larger graphs, we have not included SKG due

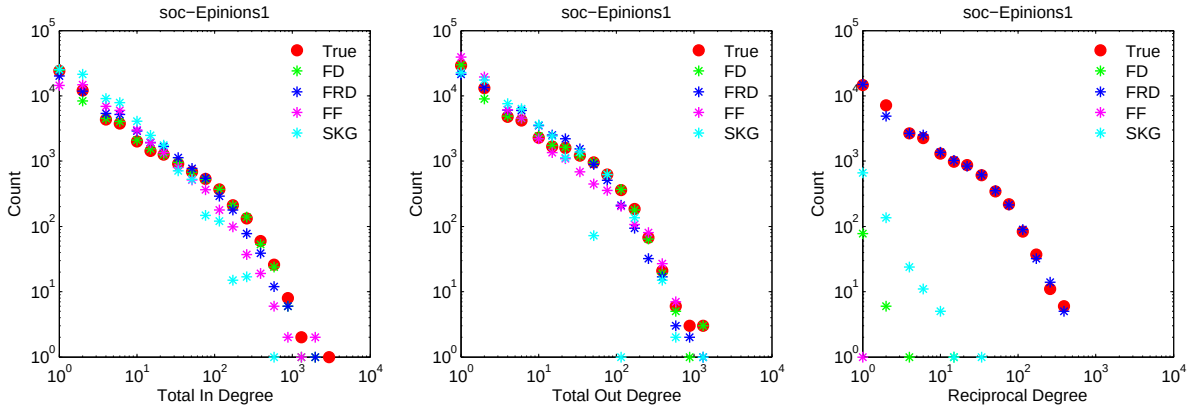

Figure 3: Comparisons of degree distributions produced by various models for graph soc-Epinions1.

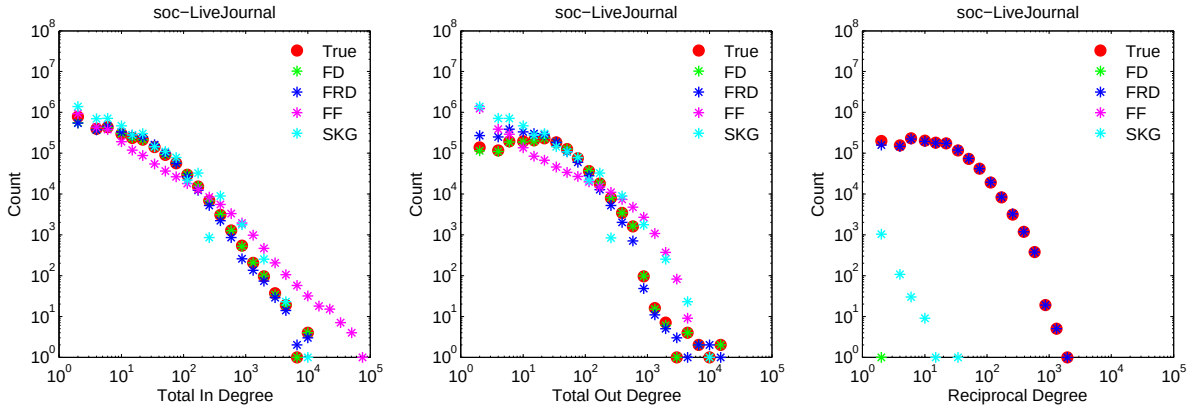

Figure 4: Comparisons of degree distributions produced by various models for graph soc-LiveJournal.

to the expense of fitting the model. We do compare to FF, however, for the youtube and flickr graphs shown in Figure 5 and Figure 6, respectively. After extensive tuning, FF is able to match the total in- and out-degree distributions fairly well. But it of course cannot match the reciprocal degree.

Finally, we shown results just for our methods on the largest graph: livejournal in Figure 7. We observe a very close match for the FRD method in all three distributions.

5 Significance and Impact

Directed networks have not received much attention in terms of generative models. An obvious first-level goal for a generative model would be to match the total in- and out-degree distributions of a given graph. We propose the FD model for this purpose. It is a fast variant of the directed Chung-Lu model [1, 3, 4], picking endpoints for each edge at random, proportional to each node's desired degree. This is a close cousin of the configuration model [14]. It compares favorably in both speed and accuracy to existing state-of-the-art models.

Directed social networks, however, cannot be modelled well without considering reciprocal edges. The proposed FD model generates very few reciprocal edges. In fact, few models explicitly generate such edges. The FF model, for example, does not even have the capability to generate a reciprocal edge since new nodes only connect to older nodes and there is no capacity for the older nodes to add additional links (i.e., back to the new node). Probably the most direct approach to this was the extension of preferential attachment in [20], but it was unable to capture both in- and out-degree distributions.

To also address modeling or reciprocal edges, we propose an effective, if straightforward, FRD approach to modeling these networks by separately building models for the reciprocal edges (modeled as an undirected

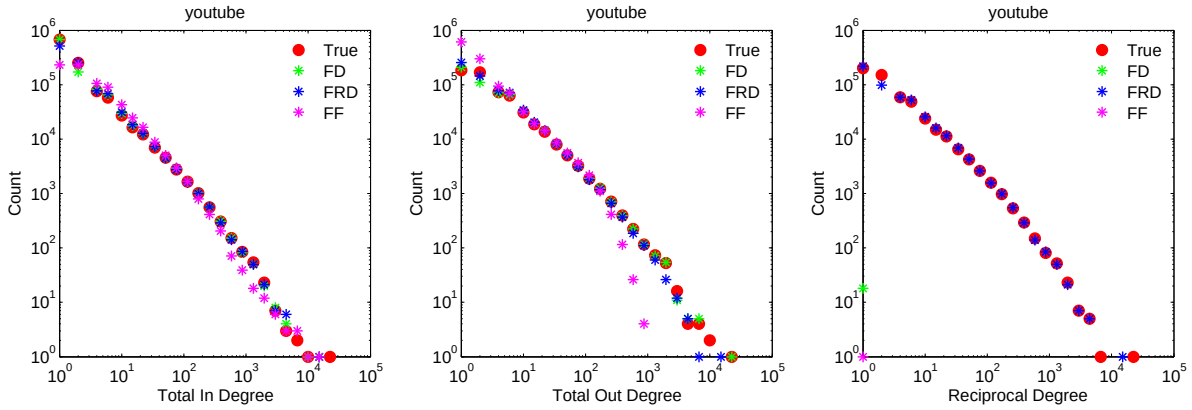

Figure 5: Comparisons of degree distributions produced by various models for graph youtube.

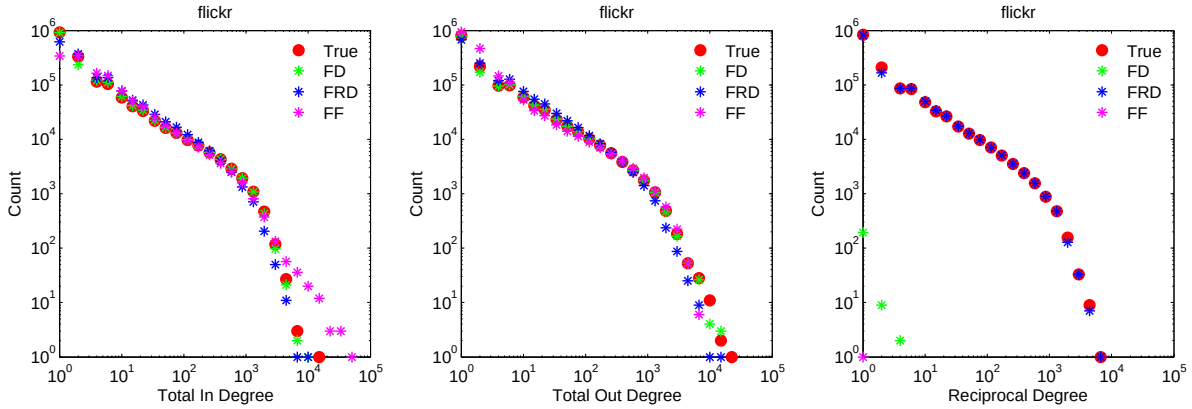

Figure 6: Comparisons of degree distributions produced by various models for graph flickr.

graph) and the non-reciprocal edges (modeled as a directed graph using FD). We combine the two graphs in an unsophisticated way — simply random mapping the nodes in the undirected graph to the set of all nodes. A more sophisticated approach would be to combine the reciprocal and non-reciprocal edges in a way that respected the total in- and out-degree distributions. To our surprise, this did not seem to be necessary, indicating that there is not a strong correlation in reciprocal and non-reciprocal edges in the graphs we studied.

Compared to state-of-the-art directed graph models such as FF [10] and SKG [8], both our FD and FRD methods are significantly faster and give a more accurate match to the various degree distributions. Moreover, there is no fitting procedure to determine the parameters of our FD and FRD methods. Even the best choice for the "blowup" parameter, if one wants to be very exact, can be determined from equations previously outlined. Of course, we would be remiss if we failed to point out that the FD and FRD models have many more parameters (i.e., the entire degree distributions) than FF and SKG, but this is still a very small amount of data compared to the overall sizes of the graphs.

Our FD and FRD methods are highly scalable since edge generation can be done in parallel, on multiple threads or across multiple machines in a distributed setting. The data required to generate each edge is the order of the size of the maximum degree, and so can be easily transmitted to multiple processors.

These models also serve as a baseline for understanding networks. For instance, a community is often defined as a subgraph with more edges than expected as compared to a random model. Similarly, baseline models are generally useful in understand the significance of observed patterns such as directed triangles or more sophisticated patterns. When analyzing social networks, explicit reciprocal edges are important since they are extremely common and clearly play a role in community

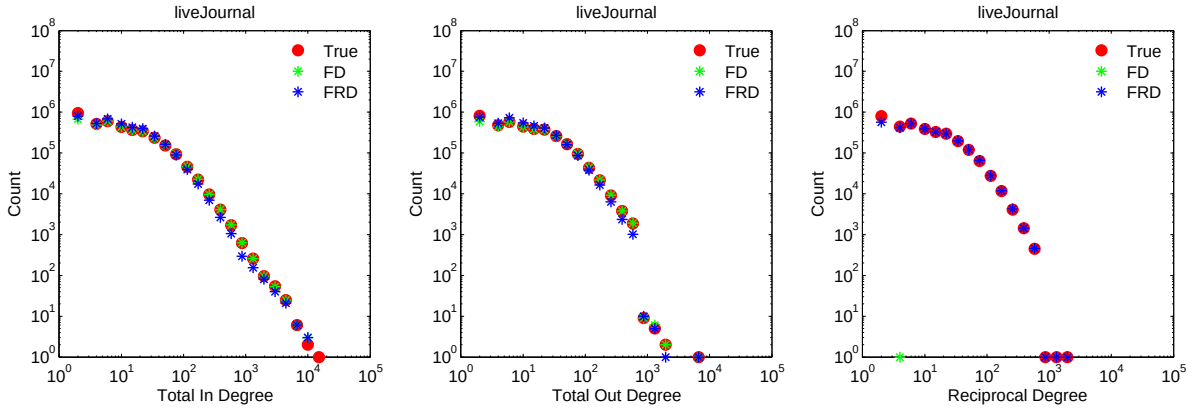

Figure 7: Comparisons of degree distributions produced by various models for graph livejournal.

structure and other observed phenomena.

References

[1] W. Aiello, F. Chung, and L. Lu, *A random graph model for massive graphs*, in STOC'00, ACM, 2000, pp. 171–180.

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

[3] F. Chung and L. Lu, *The average distances in random graphs with given expected degrees*, PNAS, 99 (2002), pp. 15879–15882.

[4] F. Chung and L. Lu, *Connected components in random graphs with given degree sequences*, Annals of Combinatorics, 6 (2002), pp. 125–145.

[5] D. Garlaschelli and M. I. Loffredo, *Patterns of link reciprocity in directed networks*, Phys. Rev. Lett., 93 (2004), p. 268701.

[6] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, *The web as a graph: measurements, models, and methods*, in COCOON'99, Springer-Verlag, 1999, pp. 1–17.

[7] H. Kwak, C. Lee, H. Park, and S. Moon, *What is twitter, a social network or a news media?*, in WWW '10, ACM, 2010, pp. 591–600.

[8] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, *Kronecker graphs: An approach to modeling networks*, J. Machine Learning Research, 11 (2010), pp. 985–1042.

[9] J. Leskovec and C. Faloutsos, *Scalable modeling of real graphs using kronecker multiplication*, in ICML'07, ACM, 2007, pp. 497–504.

[10] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graphs over time: densification laws, shrinking diameters and possible explanations*, in KDD'05, ACM, 2005, pp. 177–187.

[11] J. C. Miller and A. A. Hagberg, *Efficient generation of networks with given expected degrees*, in WAW, Springer, 2011, pp. 115–126.

[12] A. Mislove, H. S. Koppula, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, *Growth of the flickr social network*, in WOSN'08, ACM, 2008, pp. 25–30.

[13] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, *Measurement and analysis of online social networks*, in IMC'07, ACM, 2007, pp. 29–42.

[14] M. Molloy and B. Reed, *A critical point for random graphs with a given degree sequence*, Random Structures & Algorithms, 6 (1995), pp. 161–179.

[15] M. E. J. Newman, S. Forrest, and J. Balthrop, *Email networks and the spread of computer viruses*, Phys. Rev. E, 66 (2002), p. 035101.

[16] A. Pinar, C. Seshadhri, and T. G. Kolda, *The similarity between stochastic Kronecker and Chung-Lu graph models*, in SDM12, SIAM, Apr. 2012, pp. 1071–1082.

[17] C. Seshadhri, T. G. Kolda, and A. Pinar, *Community structure and scale-free collections of Erdős-Rényi graphs*, Phys. Rev. E, 85 (2012).

[18] C. Seshadhri, A. Pinar, and T. G. Kolda, *An in-depth study of stochastic Kronecker graphs*, in ICDM'11, IEEE Computer Society, 2011, pp. 587–596.

[19] C. Xu, *Making Doodle Obsolete: Applying auction mechanisms to meeting scheduling*, PhD thesis, Harvard College, Apr. 2010.

[20] V. Zlatic and H. Stefancic, *Model of wikipedia growth based on information exchange via reciprocal arcs*, EPL (Europhysics Letters), 93 (2011), p. 58005.

[21] V. Zlatić and H. Štefančić, *Influence of reciprocal edges on degree distribution and degree correlations*, Phys. Rev. E, 80 (2009), p. 016117.

[22] *Stanford Network Analysis Project (SNAP)*. Available at `http://snap.stanford.edu/`.