A Scalable Null Model for Directed Graphs Matching All Degree Distributions: In, Out, and Reciprocal

Nurcan Durak, Tamara G. Kolda, Ali Pinar, and C. Seshadhri
Sandia National Laboratories
Livermore, CA USA
Email: nurcan.durak@gmail.com, {tgkolda, apinar, scomand}@sandia.gov

***Abstract*—Degree distributions are arguably the most important property of real world networks. The classic edge configuration model or Chung-Lu model can generate an undirected graph with any desired degree distribution. This serves as a good *null model* to compare algorithms or perform experimental studies. Furthermore, there are scalable algorithms that implement these models and they are invaluable in the study of graphs. However, networks in the real-world are often directed, and have a significant proportion of *reciprocal edges*. 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*). Despite their importance, reciprocal edges have been disregarded by most directed graph models.**

We propose a null model for directed graphs inspired by the Chung-Lu model that matches the in-, out-, and reciprocal-degree distributions of the real graphs. Our algorithm is scalable and requires $O(m)$ random numbers to generate a graph with m edges. We perform a series of experiments on real datasets and compare with existing graph models.

I. Introduction

Ever since the seminal work of Barabási and Albert [1], Faloutsos et al. [2], Broder et al. [3], degree distributions are widely regarded as a key feature of real-world networks. The heavy-tailed nature of these degree distributions has been repeatedly observed in a wide variety of domains. One of the invaluable tools in analyzing heavy-tailed graphs is the ability to produce a random or "generic" graph with a desired degree distribution. The classic edge configuration [4]–[7] does exactly that and is a common method for constructing such graphs. Chung and Lu [8], [9] give more analyzable variants of this model. MCMC methods based on random walks are also used for this purpose [10], [11].

These constructions are useful for testing algorithms and comparing with existing models. It also helps in design of new algorithms. For example, versions of the stochastic block model [12], [13] used for community detection use Chung-Lu type constructions for null models. The classic notion of modularity [14] measures deviations from a Chung-Lu structure to measure community structure. At a higher level, having a baseline model that accurately matches the degree distribution informs us about other properties. Notably, work on the eigenvalue distributions on Chung-Lu graphs [15], [16] suggest the observations on so called "eigenvalue power laws" are simply a consequence of heavy tailed degree distributions. For these reasons, we think of the edge-configuration or Chung-Lu constructions as *null models*.

While all of this work has been extremely useful in advancing graph mining, it ignores the crucial property of *direction* in networks. Most interaction, communication, web networks are inherently directed, and the standard practice is to make these undirected. Furthermore, directed networks exhibit *reciprocity*, where some pairs of vertices have edges in both directions connecting them. For example, in Figure 1, there are two-way connections between some vertices. This indicates a much stronger connection between them.

Newman [17] introduces the reciprocity, r, which measures the density of reciprocal edges in a network. It can be interpreted as the probability of a random edge in a network being reciprocated. The reciprocity is often high in social networks but is lower in information networks; see Table I. It was observed that high reciprocity leads to faster spread of viruses or news [17], [18]. The importance of reciprocal edges is underscored by a study of formation order of these edges [19]. In the Flickr network (which has 68% reciprocal edges), 83% of all reciprocal edges are created within 48 hours after the initial edge creation. The Twitter network has 22.1% of the reciprocal edges [20]. Reciprocity also plays an important role in interactions in massive multiplayer online games [21]. All these studies show that reciprocal edges are quite special, and provide important information about the social processes underlying these graphs. But all graph models and constructions completely ignore these edges.

A key concern with graph generation is simple construction and scalability, as we may want test instances with millions (and more) edges. A key feature for a null model is its scalability and its ability to quickly produce a large graph that matches degree distributions.

A. Contributions

For a directed graph, there are three distinct degree distributions associated with it: the in-degree, the out-degree, and the reciprocal degree distribution. The last can be thought of as the degree distribution of the undirected subgraph obtained by only taking reciprocal edges. A good null model, along the lines of the configuration model or Chung-Lu, must match all three of these. We design the *Fast Reciprocal Directed* (FRD) graph generator that does exactly that.

978-1-4799-0203-3/13/$31.00 ©2013 IEEE

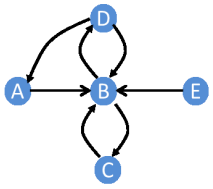

Fig. 1: A directed graph with reciprocal (e.g., B-D) and one-way (e.g., D-A) edges.

- The *Fast Reciprocal Directed* (FRD) graph generator takes as input in-,out-, and reciprocal degree distributions, and produces a random graph matching these. It can be thought of as a generalization of the Chung-Lu model for this setting. We provide a series of empirical results showing how it matches these degree distributions for real datasets.
- Our algorithm is *fast and scalable*. It only requires some minimal preprocessing and the generation of $O(m)$ random numbers. It takes less than a minute to generate a graph with multi-million nodes and edges, faster than any comparable models.
- We compare FRD's degree distributions fits with existing directed graph models. At some level, this is not a fair comparison, since we do not consider our generator to be realistic (while competing methods attempt to match other important graph properties and mimic real world processes). Our model is meant to be a baseline or null model that matches degree distributions. But our comparisons with realistic graph models are quite illuminating. Despite the large number of reciprocal edges in real networks, none of the other models come even marginally close to matching the reciprocal degree distribution.
- As an aside, we explain why the number of degree-1 nodes is much lower than intended in Chung-Lu like models [22], [23] and propose a solution to obtain a better match for the degree-1 vertices. This fix is incorporated in the FRD generator.

II. Related Work

As mentioned earlier, edge configuration models have a long history. Miller and Hagberg [24] discuss faster algorithms for implementing Chung-Lu, while Seshadhri et al. [25] discuss a different parallel version. A directed version of the edge configuration model together with mathematical analyses of connected component structure was given in [26]. Our work is related to this construction.

Reciprocal edges are not taken into account by most common graph models. The Forest Fire (FF) model [27] and Stochastic Kronecker Graph (SKG) model [28], [29] are often used to generate graphs, and do produce directed graphs. They can match in- and out-degree distributions reasonably well, and we use these models for comparisons.

Most common graph models (e.g., preferential attachment [1], edge copying model [30], forest fire [27]) produce directed graphs incrementally to imitate the growth of graphs. They produce heavy-tailed in- and out- degree distributions, but almost no reciprocal edges. Furthermore, they are not scalable to millions of nodes and billions of edges. The Stochastic Kronecker Graph model [28], [29] is scalable, but is also unable to produce reciprocity. In this study, we compare our results with the Forest Fire (FF) model and Stochastic Kronecker Graph (SKG) model.

A notable exception is work of Zlatic et al. [31], [32] that generalizes Preferential Attachment (PA) using reciprocal edges. Unfortunately, it is not scalable and does not match out-degree distributions (in their experiments). Another variant of PA [33] does allow edges between existing nodes (thereby introducing some reciprocity), but the model is not meant to really match real data.

III. The Fast Reciprocal Directed Null Model

We first introduce some 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 of a graph [17] is

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

We will present an extension of the Chung-Lu model that accounts for in- and out-degrees. This will be a part of the final FRD generator.

A. The Fast Directed Generator

In this first step, we consider only the total in- and out-degrees and ignore reciprocity. This can be thought of as a fast implementation of the directed edge configuration model in [26]. We extend the Fast Chung-Lu (FCL) algorithm for undirected graphs [23]. This 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 [22], 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 with 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 §III-C. 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 [24].

B. Introducing reciprocity

The FD model generates a directed graph and matches to the total in- and out-degree distributions. However, it produces virtually no reciprocal edges. The FRD null model explicitly introduces reciprocity using an undirected model and uses FD for remaining directed edges. We blend the two schemes 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 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

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

C. 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 or reciprocal degrees. 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 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 2. *Let n'_x denote 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. *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 a small penalty in the match for degree-2 nodes. We use $b = 10$ in all experiments reported in this paper.

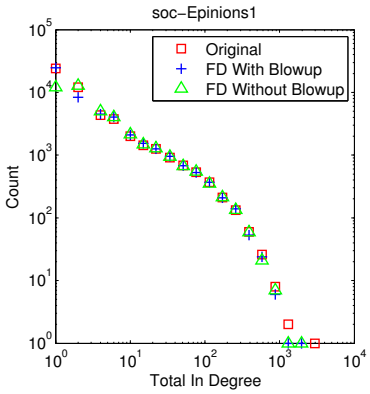

Fig. 2: Example of in-degree distribution with and without blowup factor. Note that the model with the blow-up factor matches degree-1 nodes precisely, however, the model without blow-up generates only half of the degree-1 nodes in the original graph.

IV. EXPERIMENTAL STUDIES

We test our models on various directed networks such as citation (cit-HepPh), web (web-NotreDame), and social (soc-Epinions1, soc-LiveJournal) [34]. We also test our models on large scale graphs coming from online social networks (youtube, flickr, liveJournal) [35]. We list the attributes of the networks in Table I after removing self-links and making the graph unweighted (simple). As expected, the reciprocity r is very low in the citation network. We elaborate how we fit the models to the real networks below.

a) Fast Directed (FD) and Fast Reciprocal Directed (FRD): This only requires the appropriate degree distributions of the input graphs. We used a blowup factor of $b = 10$ in all cases.

b) 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 [34]. To fit FF, we picked parameters that best match the number of edges in the real networks. For each target graph, we search a range of values by incrementing p_f value by $\delta p = 0.001$ in range [0.2-0.5] to find the best parameters, which are reported in Table I. We set $p_b = 0.32$ as described in [27].

TABLE I: Networks used in this study. r is the reciprocity, 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 [34]	34K	421K	<1K	0.003	0.37	[0.990,0.440;0.347,0.538] [29]
soc-Epinions1 [34]	76K	508K	206K	0.405	0.346	[0.999,0.532;0.480,0.129] [29]
web-NotreDame [34]	325K	1,469K	759K	0.517	0.355	[0.999,0.414;0.453,0.229] [29]
soc-LiveJournal [34]	4,847K	68,475K	32,434K	0.632	0.358	[0.896,0.597;0.597,0.099] [36]
youtube [35]	1,157K	4,945K	3,909K	0.791	0.335	—
flickr [35]	1,861K	22,613K	14,117K	0.624	0.355	—
LiveJournal [35]	5,284K	77,402K	56,920K	0.735	0.355	—

c) Stochastic Kronecker Graphs (SKG): We use the initiator matrices reported by previous studies: [29] for cit-HepPh, soc-Epinions, and web-NotreDame and [36] for soc-LiveJournal. We attempted to generate initiator matrices for large graphs using [34], but the program did not terminate within twenty-four hours. Therefore, we only fit SKG to the networks obtained from SNAP [34] data warehouse. We set the size of the final adjacency matrix as $2^{\lceil \log_2(n) \rceil}$, 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 in MATLAB. For SKG and Forest Fire, we used the C++ implementations in [34]. Graph generation time for each model is listed in Table II. For fair comparison, we do not include I/O times. 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++ codes are typically much faster than MATLAB codes.

TABLE II: Graph generation times

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	41.75s	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 III. The FF model cannot generate any reciprocal edges. The FD model can generate a few random reciprocal edges but their number is negligible. The SKG model generates some reciprocal edges; yet a negligible amount. The FRD model performs the best and generate expected amount of reciprocal edges.

We also analyze the generated degree distributions by each model. The plots are log-binned for 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 [37].) 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 produces many reciprocal edges and FF does not produce any.

TABLE III: Number of 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

For larger graphs, we have not included SKG due 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. We also show 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. For completeness, we show results for the citation network cit-HepPh in Figure 8 and web network Web-NotreDame in Figure 9.

V. Conclusion

Reciprocity in directed networks has not received much attention in terms of generative models. A first-level goal for a generative model would be to match specified in-, out-, and reciprocal degree distributions. The FRD generator does exactly that and therefore is a good null model for social network analysis. It is a variant of Chung-Lu that explicitly takes care of reciprocal edges. We find it very intriguing that existing graph models completely ignore reciprocal edges despite the relatively high fraction of such edges. While the main challenge in graph modeling would be to design a realistic model that accounts for reciprocity, we feel that FRD is a first step in that direction.

Acknowledgment

This work was funded by the DARPA Graph-theoretic Research in Algorithms and the Phenomenology of Social

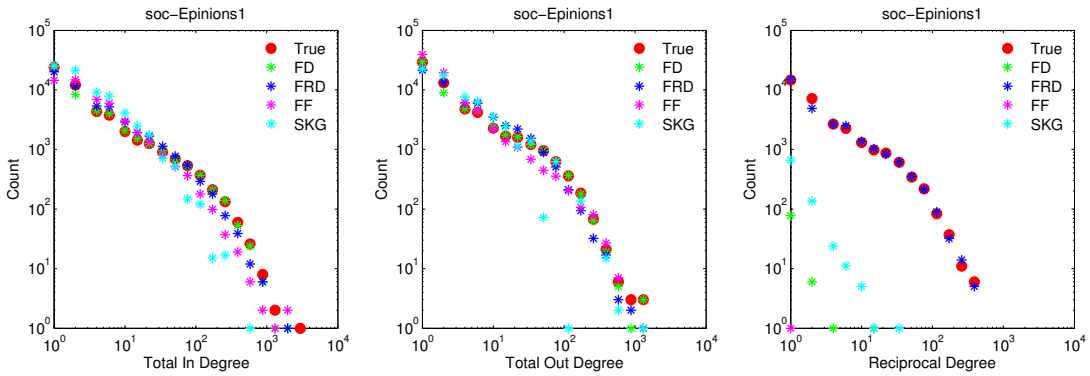

Fig. 3: Comparisons of degree distributions produced by various models for graph soc-Epinions1.

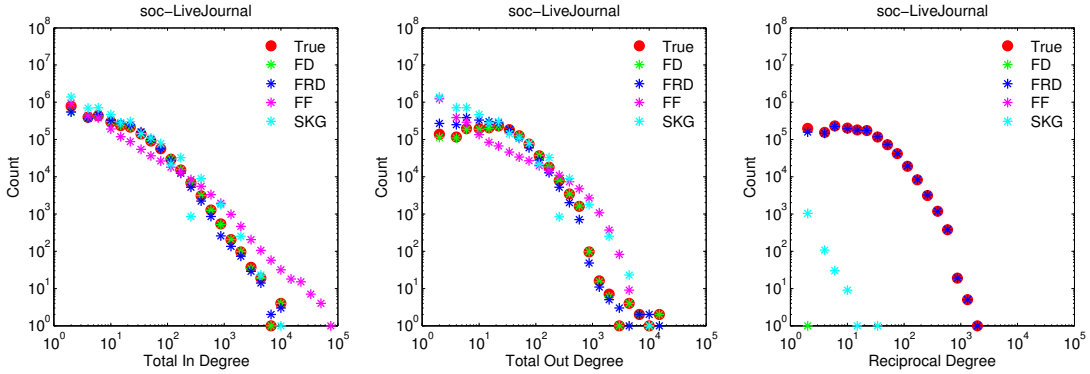

Fig. 4: Comparisons of degree distributions produced by various models for graph soc-LiveJournal.

Networks (GRAPHS) program and by the DOE Complex Distributed Interconnected Systems (CDIS) 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.

REFERENCES

[1] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks", *Science*, vol. 286, pp. 509–512, Oct. 1999.

[2] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos, "On power-law relationships of the internet topology", in *Proceedings of SIGCOMM*, 1999, pp. 251–262.

[3] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web", *Computer Networks*, vol. 33, pp. 309–320, 2000.

[4] E. A. Bender and E.R. Canfield, "The asymptotic number of labeled graphs with given degree sequences", *Journal of Combinatorial Theory A*, vol. 24, pp. 296–307, 1978.

[5] B. Bollobás, "A probabilistic proof of an asymptotic formula for the number of labelled regular graphs", *European Journal on Combinatorics*, vol. 1, pp. 311–316, 1980.

[6] M. Molloy and B. Reed, "The size of the giant component of a random graph with a given degree sequence", *Combinatorics, Probability and Computing*, vol. 7, pp. 295–305, 1998.

[7] M.E.J. Newman, "The structure and function of complex networks", *SIAM Review*, vol. 45, pp. 167–256, 2003.

[8] William Aiello, Fan Chung, and Linyuan Lu, "A random graph model for massive graphs", in *STOC'00*. 2000, pp. 171–180, ACM.

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

[10] R. Kannan, P. Tetali, and S. Vempala, "Simple Markov-chain algorithms for generating bipartite graphs and tournaments", *Random Struct. Algorithms*, vol. 14, no. 4, pp. 293–308, 1999.

[11] C. Gkantsidis, M. Mihail, and E. W. Zegura, "The Markov chain simulation method for generating connected power law random graphs", *ALENEX*, pp. 16–25, 2003.

[12] P. J. Bickel and A. Chen, "A nonparametric view of network models and Newman-Girvan and other modularities", *Proceedings of the National Academy of Sciences*, vol. 106, no. 50, pp. 21068–21073, 2009.

[13] B. Karrer and M. E. J. Newman, "Stochastic block models and community structure in networks", *Physical Review E*, vol. 83, no. 1, pp. 21068–21073, 2011.

[14] M. Girvan and M. Newman, "Community structure in social and biological networks", *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[15] Milena Mihail and Christos H. Papadimitriou, "On the eigenvalue power law", in *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, London, UK, 2002, RANDOM '02, pp. 254–262, Springer-Verlag.

[16] F. Chung, L. Lu, and V. Vu, "Eigenvalues of random power law graphs", *Annals of Combinatorics*, vol. 7, pp. 21–33, 2003.

[17] M. E. J. Newman, Stephanie Forrest, and Justin Balthrop, "Email networks and the spread of computer viruses", *Phys. Rev. E*, vol. 66, no. 3, pp. 035101, Sept. 2002.

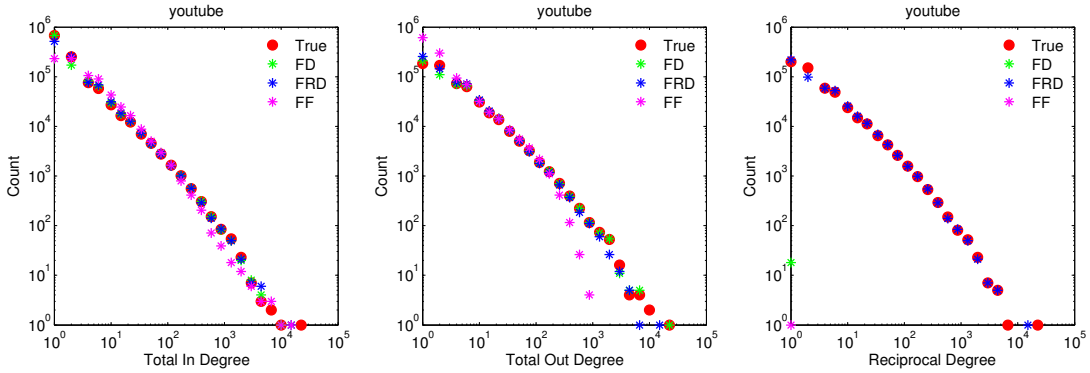

Fig. 5: Comparisons of degree distributions produced by various models for graph youtube.

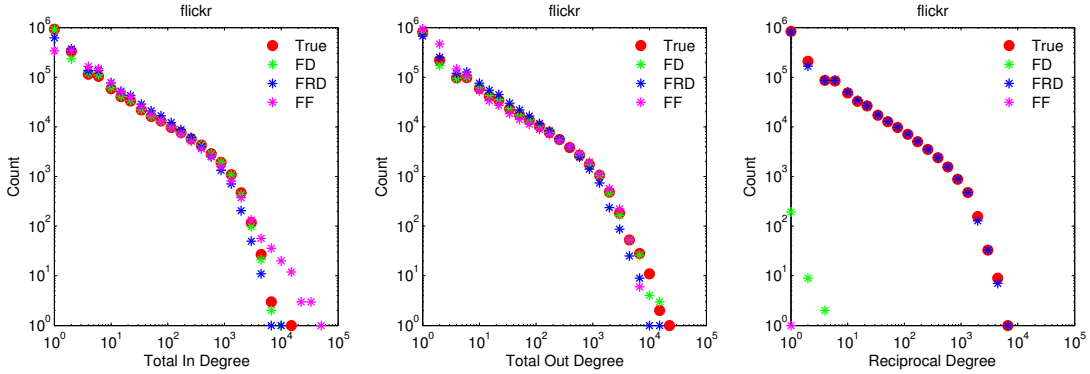

Fig. 6: Comparisons of degree distributions produced by various models for graph flickr.

[18] Diego Garlaschelli and Maria I. Loffredo, "Patterns of link reciprocity in directed networks", *Phys. Rev. Lett.*, vol. 93, pp. 268701, Dec. 2004.

[19] Alan Mislove, Hema Swetha Koppula, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee, "Growth of the flickr social network", in *WOSN'08*. 2008, pp. 25–30, ACM.

[20] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon, "What is twitter, a social network or a news media?", in *WWW '10*. 2010, pp. 591–600, ACM.

[21] K. Subbian, A. Singhal, T. Kolda, A. Pinar, and J. Srivastava, "On reciprocity in massively multi-player online game networks", work in progress, 2013.

[22] Fan Chung and Linyuan Lu, "The average distances in random graphs with given expected degrees", *PNAS*, vol. 99, no. 25, pp. 15879–15882, 2002.

[23] C. Seshadhri, Tamara G. Kolda, and Ali Pinar, "Community structure and scale-free collections of Erdős-Rényi graphs", *Phys. Rev. E*, vol. 85, May 2012.

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

[25] T. Kolda, A. Pinar, T. Plantenga, and C. Seshadhri, "A scalable generative graph model with community structure", arXiv:1302.6636, 2013.

[26] L. A. Meyers, M. E. J. Newman, and B. Pourbohloul, "Predicting epidemics on directed contact networks", *Journal of Theoretical Biology*, vol. 240, pp. 400–418, 2006.

[27] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations", in *KDD'05*. 2005, pp. 177–187, ACM.

[28] Jure Leskovec and Christos Faloutsos, "Scalable modeling of real graphs using kronecker multiplication", in *ICML'07*. 2007, pp. 497–504, ACM.

[29] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani, "Kronecker graphs: An approach to modeling networks", *J. Machine Learning Research*, vol. 11, pp. 985–1042, Feb. 2010.

[30] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins, "The web as a graph: measurements, models, and methods", in *COCOON'99*. 1999, pp. 1–17, Springer-Verlag.

[31] Vinko Zlatić and Hrvoje Štefančić, "Influence of reciprocal edges on degree distribution and degree correlations", *Phys. Rev. E*, vol. 80, pp. 016117, Jul 2009.

[32] V. Zlatic and H. Stefancic, "Model of wikipedia growth based on information exchange via reciprocal arcs", *EPL (Europhysics Letters)*, vol. 93, no. 5, pp. 58005, 2011.

[33] Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan, "Directed scale-free graphs", in *SODA'03*. 2003, pp. 132–139, ACM.

[34] "Stanford Network Analysis Project (SNAP)", Available at http://snap.stanford.edu/.

[35] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee, "Measurement and analysis of online social networks", in *IMC'07*. 2007, pp. 29–42, ACM.

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

[37] C. Seshadhri, Ali Pinar, and Tamara G. Kolda, "An in-depth study of stochastic Kronecker graphs", in *ICDM'11*. 2011, pp. 587–596, IEEE Computer Society.

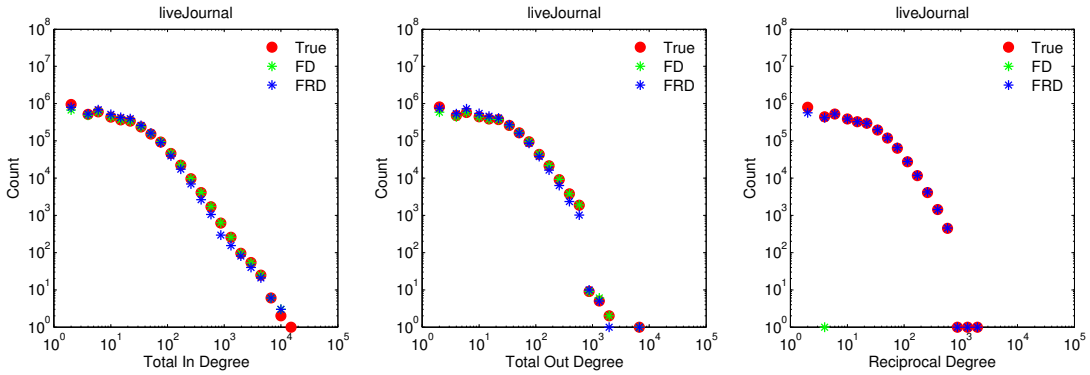

Fig. 7: Comparisons of degree distributions produced by various models for graph livejournal.

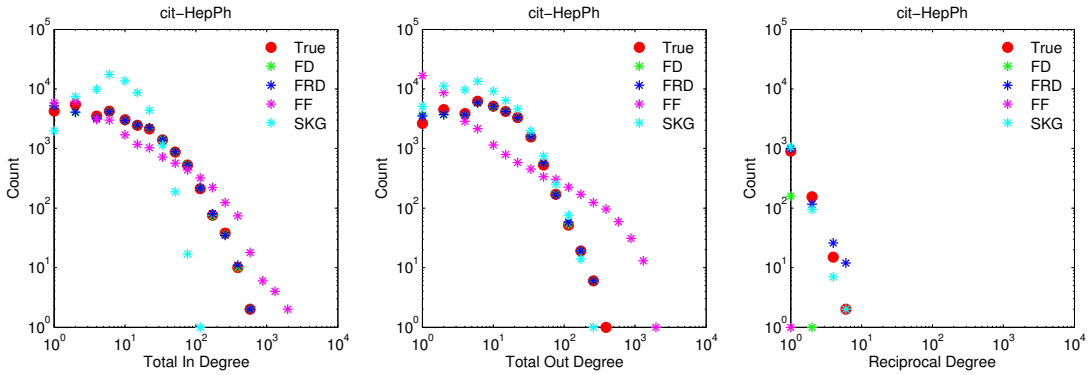

Fig. 8: Comparisons of degree distributions produced by various models for graph cit-HepPh.

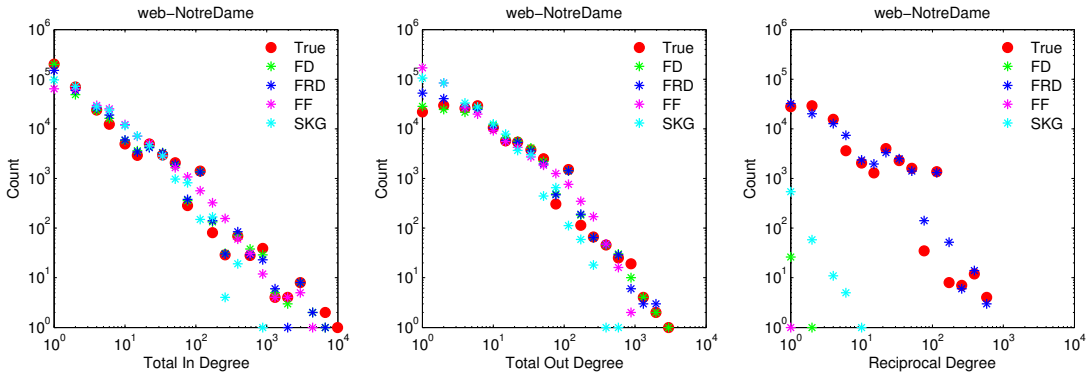

Fig. 9: Comparisons of degree distributions produced by various models for graph web-NotreDame.