Measurement-calibrated Graph Models for Social Network Experiments

Alessandra Sala, Lili Cao, Christo Wilson, Robert Zablit,
Haitao Zheng and Ben Y. Zhao
Computer Science, U. C. Santa Barbara, Santa Barbara, CA 93106, USA
{alessandra, lilicao, bowlin, rzablit, htzheng, ravenben}@cs.ucsb.edu

ABSTRACT

Access to realistic, complex graph datasets is critical to research on social networking systems and applications. Simulations on graph data provide critical evaluation of new systems and applications ranging from community detection to spam filtering and social web search. Due to the high time and resource costs of gathering real graph datasets through direct measurements, researchers are anonymizing and sharing a small number of valuable datasets with the community. However, performing experiments using shared real datasets faces three key disadvantages: concerns that graphs can be de-anonymized to reveal private information, increasing costs of distributing large datasets, and that a small number of available social graphs limits the statistical confidence in the results.

The use of measurement-calibrated graph models is an attractive alternative to sharing datasets. Researchers can "fit" a graph model to a real social graph, extract a set of model parameters, and use them to generate multiple synthetic graphs statistically similar to the original graph. While numerous graph models have been proposed, it is unclear if they can produce synthetic graphs that accurately match the properties of the original graphs. In this paper, we explore the feasibility of measurement-calibrated synthetic graphs using six popular graph models and a variety of real social graphs gathered from the Facebook social network ranging from 30,000 to 3 million edges. We find that two models consistently produce synthetic graphs with common graph metric values similar to those of the original graphs. However, only one produces high fidelity results in our application-level benchmarks. While this shows that graph models can produce realistic synthetic graphs, it also highlights the fact that current graph metrics remain incomplete, and some applications expose graph properties that do not map to existing metrics.

Categories and Subject Descriptors

I.6.4 [**Computing Methodologies**]: Model Validation and Analysis; D.4.8 [**Operating Systems**]: Performance—*Measurements, Modeling and prediction*

General Terms

Experimentation, Measurement

1. INTRODUCTION

Access to realistic measurement data is critical to accurate research results in a variety of network domains. Prior work on wired and wireless networks has shown that the validity of experimental results depends on the accuracy of the data used [12, 16]. This is especially true for the growing area of online social networks, where evaluation of social applications can produce very different results depending on the social graphs used [35]. As researchers gain awareness of the need for real data, they are frustrated by the difficulty of performing measurements on existing social networks, many of whom now take careful steps to prevent automated crawlers. Instead of commiting costly resources to gather real social graphs, researchers seek access to a small number of measurement-generated graphs available to the community[1].

However, the continued distribution and experimental use of social graph datasets faces three significant challenges. First, owners of datasets are increasingly concerned about inadvertently revealing private information with their anonymized datasets. Recent work shows that malicious parties can recover information from anonymized graphs by *de-anonymizing* them using either auxiliary graphs or by identifying unique subgraphs in the anonymized graph [4, 27, 28]. Given recent privacy compromises [7], these concerns act as a strong disincentive against sharing graph datasets. Second, the limited number of available graphs is insufficient to generate meaningful experimental results. Ideally, researchers would like to experiment with multiple real graphs to produce statistically confident results. Finally, social networks are exploding in size, and measured graphs contain millions of nodes and hundreds of millions of edges. Even compressed, graphs from our Facebook study [35] can be over 50GBs in size. Sharing this data requires either a multi-day download over a fast network or shipping hard-drives, neither of which scales with the demand for these graphs.

Synthetic graphs generated by measurement-calibrated graph models offer an attractive alternative to sharing large graph datasets. Trace-driven modeling is popular in research settings where real measurement data is difficult to gather, such as wireless networks [16, 22]. In the social graph context, we can "fit" a graph model to a real social graph, thereby extracting a set of model parameters. We feed these parameters into the graph model to produce randomized synthetic graphs that match the original in statistical properties. If accurate, experimental results from these synthetic graphs will closely match those from experiments performed on the original graph. This approach addresses all of the aforementioned challenges: synthetic graphs are randomized, hence no privacy is compromised; additional graphs can be generated on demand, thus improving statistical confidence; and model parameters are compact, hence cost of sharing models is trivially low.

But which graph model should we use? Instead of creating a new graph model, we choose to determine if any of the numerous

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

[1]Since being available in May 2009, 31 research groups have made use of social graphs from our recent Facebook study [35].

models in literature are suitable. Of the recent graph models, only two (dK [23] and Kronecker graphs [19]) are designed to capture overall structural characteristics of graphs. These *structure-driven* models would be ideal for our use, but they incur very high costs in memory or computation. To apply them to large social graphs, we must first use parameters to reduce their overheads, *i.e.* limiting dK graphs to dK-2, and reducing matrix size and number of iterations for Kronecker graphs. Unfortunately, this also reduces model accuracy. In contrast, *intent-* and *feature-driven* models such as Nearest Neighbor and Forest Fire have significantly lower algorithmic and computational complexity, but are only designed to capture a single graph property. To be inclusive, our work must consider all three types of graph models.

We must answer several other key questions before we can accept the validity of research results using synthetic graphs generated by calibrated models. What challenges are involved in fitting graph models to real social graphs? How accurately can model-generated synthetic graphs capture the statistical metrics of real graphs? We refer to a model's ability to produce statistically similar graphs as its *fidelity*. Which models demonstrate the highest levels of fidelity for today's large social graphs? Do current graph metrics capture all of the meaningful properties of real graphs? And finally, can application-level results obtained by researchers using synthetic graphs match those obtained using the original graphs?

In this paper, we seek to answer these questions by exploring the feasibility of replacing real graphs with synthetic graphs generated from calibrated graph models. We make three key contributions:

1. We examine the challenge of fitting graph models to specific graphs. We explore the problem for several popular graph models, and propose a two-phase parameter search approach guided by a structural graph similarity metric.
2. We use our methodology to examine a set of popular graph models from literature, and evaluate how accurately each model captures statistical metrics from graphs of the Facebook social network ranging from 30,000 to 3 million edges. We find that while most vary significantly in accuracy, two graph models (Nearest Neighbor and dK-2) are consistently accurate for most metrics across all of our test graphs.
3. We examine the impact of using synthetic graphs through simulations of social network applications. We use these application-level tests as "end-to-end" metrics to test the feasibility of substituting real graphs with synthetic graphs. Our results show that the Nearest Neighbor model produces results on synthetic graphs closely matching those of real graphs, thus confirming that model-generated synthetic graphs can be reliably used in research on social networks.

While this work focuses on social network graphs, we believe our methodology is general, and we can use similar techniques to evaluate the feasibility of measurement-calibrated graph models for Internet routers, the web, and biological graphs.

2. METHODOLOGY AND CHALLENGES

Our goal is to identify which graph models, if any, can generate synthetic graphs that are sufficiently representative of real-world social graphs to be suitable for experimental research. We refer to the a model's ability to reproduce statistically similar synthetic graphs as its *fidelity*. Thus, our restated goal is to determine the fidelity of current graph models, and whether any model has sufficiently high fidelity to replace real social graphs in research.

Our approach (shown in Figure 1) consists of three steps: collecting real-world social graphs, fitting graph models to target graphs, and quantifying each model's fidelity by comparing the resulting synthetic graphs against the originals using graph metrics and application benchmarks. Here, we summarize our methodology, identify 4 challenges in the process, and describe our solutions.

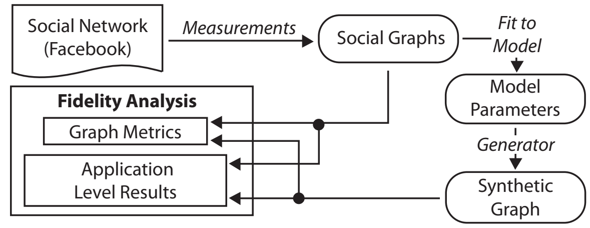


Figure 1: Our methodology for evaluating graph model fidelity.

Graphs	Nodes	Edges
Monterey Bay, CA	6,283	33,969
Santa Barbara, CA	12,814	92,241
Egypt	246,692	1,618,085
New York, NY	377,712	3,616,873

Table 1: Four representative social graphs from Facebook measurements in 2008. They vary in size from very small (Monterey Bay, CA) to extremely large (NY, NY).

Collecting Social Graphs. Facebook is the largest social network in the world with more than 350 million users. We use several Facebook social graphs we obtained through detailed measurements between March and May 2008 [35]. Our data includes anonymized social graphs encompassing more than 10 million users with over 940 million social links from the 22 largest regional networks, as well as several smaller regional networks. These Facebook social graphs are attractive as target graphs for this work for two reasons. First, our prior analysis [35] showed that they are *representative* of measured social graphs, *i.e.* they display graph properties similar to measurements of other popular social networks such as Orkut [25]. Second, the availability of a wide range of Facebook graphs means we can choose multiple graphs of different sizes. Ultimately, we chose to use 4 representative regional networks (listed in Table 1) ranging from 6000 nodes and 30,000 edges to 300,000 nodes and 3 million edges.

Fitting Models to Target Graphs. We compare the fidelity of different models to determine their suitability as replacements for measured social graphs. We consider several well-known graph models of social networks developed from the fields of mathematics, physics and computer science. For each model, we use social graphs from Facebook as targets, and determine the optimal model parameters that provide the best fit for the model and a given graph. We then use these parameters to generate randomized graphs that attempt to match the target graph's salient graph properties.

We face three challenges in this phase of our work. First, existing graph models cannot be used directly to generate synthetic graphs because their output is restricted in some manner. For example, Nearest Neighbor [33] has been analytically shown to generate graphs with Power-law coefficients always > 2, and Random Walk [33] produces directed graphs that may be disconnected. We modify these models to produce general graphs matching our social graph datasets. For models such as dK [23] and Kronecker graphs [19], we also tune parameters to trade off accuracy for model complexity. Model modifications are described in Section 3.

Our second challenge lies in fitting the models to our target graphs, *i.e.* for a given graph G and model M, determining the optimal parameters for M to generate graphs that best match G. However, this leads us to the third challenge: "how do we quantify similarity between graphs?" As we will explain in Section 4, we build a graph similarity metric using the dK-series [23], a structure-based graph metric that, given sufficient space, can uniquely identify a target graph. Using this metric, we perform adaptive precision search through the parameter space until we find the best fit parameters.

Evaluating Model Fidelity. Finally, once we have computed the best fit parameters for a given model and target graph, we can compute the fidelity of the model with respect to a given metric. The final challenge is identifying the correct metric(s) that capture the properties of interest to experimental research. We start with a comprehensive set of accepted social graph metrics, including the power-law degree distribution, node separation, clustering coefficient and assortativity. For each target graph, we quantify a model's fidelity by measuring the Euclidean distance between the target's graph metrics and those of the synthetic graphs.

These metrics may not tell the whole story, however. Ultimately, we do not yet understand how these graph metrics are related, or whether existing graph metrics completely capture all properties of a graph. Therefore, the only reasonable way to determine the fidelity of a graph model for experimental research is to feed the original and synthetic graphs into "application-level" tests and examine the difference in their results. We perform a suite of simulations of well-known social network applications, including Sybilguard [36], Reliable Email [13], and Social Shields for anonymous communication [30]. Examining the "error" between application results from original graphs and those of synthetic graphs provides an end-to-end test that answers two questions: do current graph metrics capture the features of graphs "important" to social networking research, and can researchers safely rely on synthetic graphs to produce meaningful and accurate experimental results?

3. SOCIAL GRAPH MODELS

In this section, we briefly describe six well-known graph models that we consider as potential models to replace real social graphs. We divide these models into three classes based on their methodology. We classify the classical Barabasi-Albert model [5] and the Forest Fire [20] model as *feature-driven*, since they focus on reproducing statistical features of a graph such as power-law distribution and dynamic changes in graph density. *Intent-driven* graph models such as Random Walk [33] and Nearest Neighbor [33] focus on emulating the creation process of the original graphs. Finally, *structure-driven* models capture statistics from the graph structure, allowing a corresponding generator to reproduce random graphs with the same structural constraints. This class includes Kronecker graphs [19] and dK-graphs [23].

We omitted a number of graph models from our study. The Watts and Strogatz model [34] generates small-world graphs, but is unsuitable for our study because it does not produce graphs with power-law degree distribution. Other models were omitted because they are similar to models in our chosen set. This includes variants of the Nearest Neighbor [32], Random Walk, and Forest Fire models, such as the copying model [17], the duplication divergence model [33] and the random surfer model [8].

3.1 Feature-driven Models

Barabasi-Albert. Barabasi and Albert [5] proposed the classical model which produces graphs with power-law degree distributions missing from random graphs [11]. This model proposed an incremental growth model for graph construction, and preferential attachment: the idea that new nodes tend to attach to existing nodes with non-uniform probability. The model has two parameters: n, number of nodes in the graph, and m, number of edges introduced from each new node to existing nodes. Given its impact on other models, we include it as a baseline measure.

Forest Fire (modified). Leskovec et al. observed increases in density and decreases in diameter over time in graphs such as the patent citation graph and Internet AS connectivity graph [20]. To capture these dynamic effects, they propose the Forest Fire model, where the graph grows with each new node connecting to a set of existing nodes. After the new node connects to an existing node, it randomly connects to some of the node's neighbors. This process is executed recursively, imitating the "burning" of forest fires.

Since the Forest Fire model generates directed graphs, we make a simple modification for it to generate undirected graphs for our study. Specifically, we always create undirected edges and follow the edges in both directions in the "burning" process. This model has two parameters: n, number of nodes in the graph, and p, the rate which decides the number of neighbors "burned" in each recursion.

3.2 Intent-driven Models

Intent-driven models attempt to capture how power-law graphs form and grow by emulating the processes behind link formation between nodes, *e.g.* formation of friendships in offline social networks and adding links on a webpage to other sites.

Random Walk (modified). The Random Walk model [33] emulates the randomized walk behavior of friend discovery in online social networks. Each new node performs a random walk starting from a randomly chosen node in the graph. As the walk traverses the graph, the new node probabilistically attaches itself to each visited node. The original model creates directed graphs. We modify the model to create undirected edges, and allow the random walk to traverse edges in any direction. The model has three parameters: n, the number of nodes, q_e, the probability of continuing the walk after each step, and q_v, the probability of attaching to a visited node. The original Random Walk model can generate disconnected graphs. We fix $q_v = 1$ for each new node's first edge, in order to ensure a generated graph with a single connected component.

Nearest Neighbor (modified). Another model based on social behaviors is the Nearest Neighbor model [33]. It follows the observation that two people sharing a common friend are more likely to become friends. Each new node added to the graph is connected to a random existing node. Additionally, random pairs of 2-hop neighbors around the new node are connected. The original model has two parameters: n, the number of nodes, and u, a probability that determines at each step if a new node is added or if a pair of 2-hop neighbors are connected.

Analysis shows that the original model always produces graphs with power-law exponent greater than 2 [33]. This does not match known measurements of social networks such as Facebook, YouTube, Flickr and Orkut, which all have power-law exponents between 1.5 and 1.75 [25, 35]. Thus, we modify the model by adding a parameter k. Each time a new node is added, we also connect k pairs of existing nodes randomly chosen from the graph.

Because the power-law exponent scales with the intensity of preferential attachment in random graphs, adding edges between node pairs selected uniformly at random reduces the level of preferential attachment, and thus the power-law exponent. By extending the analysis in [33] to our modified model, we show that the power-law exponent γ is a function of k and u: $\gamma \approx 1 + \frac{1}{\beta}$, where

$$\beta = \frac{u}{2(1-u)}\left(-1 + \sqrt{1 + 4\frac{(k+1)(1-u)}{u}}\right).$$

	Monterey	S. B.	Egypt	New York
dK-1	102	153	416	385
dK-2	4,230	9,477	45,245	51,238
dK-3	258,871	884,931	6,919,578	10,475,401

Table 2: The number of values required to represent our graphs using different dK-series.

This produces the desired reduction in Power-law distribution while maintaining the intuition behind the model.

3.3 Structure-driven Models

Unlike models that focus on single properties or incremental growth of the graph, structure-driven models focus purely on capturing the physical characteristics of the target graph. Thus they have no graph formation parameters, only parameters that trade off accuracy for model complexity. Given their design goals, these models should provide the most representative synthetic graphs. While they are likely to capture a graph's structure, however, they are also likely to incur high costs in computation and/or memory, thus limiting their achievable accuracy in practical settings.

Kronecker Graphs. Leskovec et al. proposed using Kronecker graphs to approximate real graphs [19]. Kronecker graphs are generated by the recursive evolution of an initiator graph. This evolution process, called Kronecker multiplication, is able to approximate real graph structures [20]. KronFit is an algorithm that generates synthetic graphs that are structurally similar to a given target graph. The similarity is measured by a maximum likelihood value, *i.e.*, the probability that this model will generate a graph identical to the original.

KronFit includes parameters that tradeoff optimality and computation complexity. The most important parameters are the size of the initiator matrix i_{kro}, the sample size for estimating the likelihood and its gradient s_{kro}, and the maximum number of gradient descent iterations in the search g_{kro}. Larger parameters map to higher accuracy as well as higher complexity. With guidance from the authors of [19], we chose the following values to keep the running time comparable to other graph models (less than 48 hours for Egypt and New York graphs on a server with 32GB of RAM): i_{kro} = 3, s_{kro} = 500,000, g_{kro} = 50.

dK**-graphs.** Finally, dK-graphs are a systematic way of extracting subgraph degree distributions from a target graph, so that similar synthetic graphs can be generated with identical degree distributions [23]. As the value of d increases, dK incorporates degree distributions of increasingly large subgraphs. For example, the dK-1 metric captures the node degree distribution, dK-2 captures the joint degree distribution, and dK-3 captures the clustering coefficient. As d increases beyond 3, the distribution becomes increasingly likely to uniquely define the target graph [23].

dK models' running time and computation state size increase rapidly as d increases. As shown in Table 2, the amount of state to capture a graph grows rapidly from dK-2 to dK-3, and becomes prohibitively costly for large graphs like New York. In this work, we present results using dK-2 model for two reasons: to avoid extremely large memory requirements for large graphs like New York, and because graph generators for dK-3 do not yet exist.

4. FITTING MODELS TO GRAPHS

We now describe our efforts to fit each graph model to Facebook social graphs. All six models are parameterized by n, number of nodes in the graph, and four models require additional parameters.

4.1 A Case for Parameter Sampling

Maximum likelihood estimation (MLE) is the best-known statistical method for fitting a statistical model to data and estimating a model's parameters. For different parameters, it calculates the maximum probability that a parameterized model generates the data exactly matching the original, and chooses the parameters that maximizes such probability. Applying MLE to graph model fitting, however, is very difficult. For large graphs like ours, there are no efficient solutions to determine if two graphs are physically identical. This is the well-known graph isomorphism problem whose difficulty has been proven in prior work [14].

Instead, we propose to use a parameter-search based solution by scanning the possible parameter space, and guiding the search using a statistical similarity metric between the target graph and the model generated graphs. We choose this solution for our graph models because it produces good results within tractable computation times, despite the massive sizes of our graphs. This is a first attempt at a practical solution, and we leave the search for more efficient solutions for future study.

Implementing this solution requires us to solve two technical challenges. First, we need a metric to measure the statistical difference between graphs, for which we propose to leverage the dK series, a graph distribution that captures subgraph degree distributions [23]. Second, we need an efficient strategy to search through the large parameter space to quickly locate near-optimal parameters. For this we propose a space-sampling solution with adaptive precision.

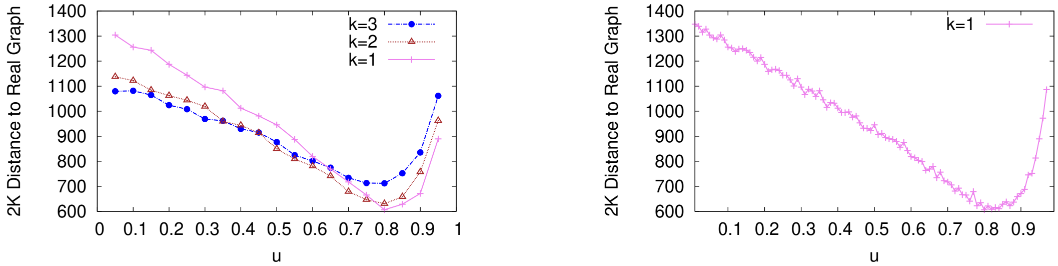
Structure-Driven Graph Comparison. We consider the problem of quantifying the similarity between any two graphs. A naive solution is to organize the accepted social graph metrics (Section 5.1) into a vector, each with a weight. We can define the statistical difference between graphs as the distance between the vectors derived from each graph. The problem with this approach is that it assumes our graph metrics are comprehensive, and that we can assign the "right" weight to each metric.

Rather than focusing on known graph metrics, we propose to use the dK-series as a single similarity metric to capture a graph's physical characteristics. We do so for two reasons. First, with increasing d, dK can progressively capture increasingly complex graph properties [23] until the graph is uniquely defined by the dK model. Second, the dK-series captures significantly more detail of graph structures than alternative metrics. For a given d, we calculate the distance between two graphs as the square distance between the dK vectors of the two graphs.

Given the memory and time complexities of dK for higher values of d (see Table 2), we limit ourselves to the dK-2 series. Using dK-2, parameter fitting a Nearest Neighbor model to our New York graph requires 2 days of computation on a quad CPU server. More accurate dK-3 requires orders of magnitude more values to represent each graph, making it impractical for larger graphs.

Parameter Sampling with Adaptive Precision. When it comes to parameter fitting models, local search algorithms such as hill climbing [31] are the most widely used solutions. Hill climbing starts from a random (potentially poor) solution, and iteratively improves the solution by making small changes until no more improvements are found. Hill climbing does not work well for non-convex spaces, however, since it will terminate when it finds a local maxima. We have experimented with hill climbing in our model fitting problem, and confirmed that it produces suboptimal results because the similarity metric (dK or others) is not strictly convex.

To overcome this problem, we apply a sampling method that finds the best fit parameters by uniformly scanning the possible



(a) with $k = 1, 2, 3$ and coarse sampling of u.   (b) with $k = 1$ and fine sampling of u.

Figure 2: Two-level parameter sampling for the Nearest Neighbor model on the Monterey Bay graph. Using dK-2 as the graph similarity metric, (a) shows the first level sampling with varying values of k (values of $k > 3$ are not shown). Error (distance from the target graph) is shown on the Y-axis. After choosing k=1 from (a), (b) fine-tunes for the parameter u. The final parameters are $k = 1$ and $u = 0.8$.

parameter space given a reasonable constraint on the level of precision. We choose this simple approach because it requires minimum information on parameter statistics. In contrast, advanced sampling methods such as Gibbs sampling [1] require knowledge of the conditional distribution of each parameter, which is very costly to compute. We also apply techniques to improve efficiency and accuracy. First, we use theoretical analysis to narrow down the parameter space based on the statistics of the real graph. We partition the remaining space uniformly to avoid local maxima. Second, we apply an initial coarse sampling to identify candidate parameter regions and then perform fine-grained sampling within these regions. Despite the simplicity of this approach, results in Section 5 confirm that it locates model parameters that produce synthetic graphs that closely approximate metrics of the original graph.

4.2 Fitting Algorithms in Detail

We now apply the fitting algorithm to each model.

Nearest Neighbor. Our modified Nearest Neighbor model has two parameters: $0 < u < 1$, and $k = 1, 2, 3, \cdots$. Since u and k determine the power-law exponent γ, we also examine the resulting γ to eliminate unsuitable choices of u and k. In the remaining two-dimensional parameter space, we apply a multi-level sampling method. First, we vary k from 1 to 10 and for each k sample u coarsely with $\Delta u = 0.05$. Using the Monterey Bay graph as an example, Figure 2(a) shows the dK-2 based distance between the original and Nearest Neighbor-generated graphs. This model produces graphs with minimal dK-2 distance from the target when $k = 1$. Next, having fixed $k = 1$, we apply a fine sampling on u with $\Delta u = 0.01$. Results in Figure 2(b) show that the fine grain sampling avoids a significant number of local maxima. The final parameters for Monterey Bay are $k = 1$ and $u = 0.8$.

Random Walk. The modified model has two parameters to tune: $0 < q_e < 1$ and $0 < q_v < 1$. Both are real numbers and their contributions are inter-related. Our sampling takes two steps. We start from a coarse sampling on q_e with $\Delta q_e = 0.1$, and for each q_e we sample q_v with the same precision $\Delta q_v = 0.1$. Using these results we identify multiple candidate intervals where the synthetic graphs are closer to the original real graph. Next, we apply a fine-grained sampling across these intervals with $\Delta q_v = 0.01$ and $\Delta q_e = 0.01$, and choose the best configuration that minimizes the dK-2 distance.

Forest Fire. The Forest Fire model has only a single parameter, p the burn rate. For each target graph, we apply a fine-grained sampling ($\Delta p = 0.01$) across its range to find the best fit p.

Barabasi-Albert. This model has only one unknown parameter m, the number of edges introduced from each new node to existing nodes. Since m is a static parameter, we compute it as $m = |E|/n$. This follows naturally because given n nodes, the total number of edges in the graph is $n \cdot m$.

Computational Costs. Our experience shows that the parameter search approach is computationally tractable for large graphs. Running experiments on a Dell 2900 server w/ 32GB of RAM, most models can be fit to the largest of our graphs (New York, 3.6M edges) within 48 hours. In all cases, model fitting runtime is dominated by the time required to generate candidate graphs as we search through the model parameter space. Computing the dK-2 distributions is also a factor, but rarely contributes more than 1 hour to the total fitting time. The time required to compare dK-2 distributions is negligible (a few ms). Finally, we found that Kronecker graphs and the Forest Fire model are the most computationally intensive to fit.

5. FIDELITY UNDER GRAPH METRICS

Having extracted the best parameters for each model and Facebook graph combination, we can evaluate the fidelity of the models by comparing the properties of the Facebook graphs against their synthetic counterparts. We first evaluate the fidelity of graph models using graph metrics described in literature. In the rest of the paper, we identify a graph G as $G = (V, E)$ where V is the set of vertices representing social network users, and E is the collection of undirected edges representing links among users.

We evaluate the six graph models using the Facebook graphs listed in Table 1. For each target graph, we apply the fitting mechanism described in Section 4 to compute the best parameters for each model. We generate 20 randomly seeded synthetic graphs from each model for each target graph, and measure the differences between them using several popular graph metrics. We examine model fidelity by computing the Euclidean distance between metrics derived from the target and model-produced graphs. We represent node degree distribution, clustering coefficient and joint degree distribution as functions of the node social degree, and compute each metric's Euclidean distance as the average square root of the total squared errors in metric values. All results are averages from comparing the 20 synthetic graphs against the original. Standard deviation values are consistently low relative to the values themselves, and are omitted for clarity.

5.1 Social Graph Metrics

We now summarize the suite of graph metrics we use to determine the fidelity of our graph models.

Node Degree Distribution (NDD). Social degree refers to the number of friends (or edges) each node has. Measurements show that node degree distributions in social graphs follow a power-law distribution, *i.e.* the fraction $P(k)$ of nodes in the graph having k connections to other nodes grows as $P(k) \sim k^{-\gamma}$, where γ is a constant exponent. We compute γ by fitting a graph's degree distribution to a power-law using the method described in [9].

Joint Degree Distribution. There are several different ways to capture the joint degree distribution, including the k_{nn} function, assortativity, and the s-metric [23]. k_{nn} computes the correlation between a node's degree and the average degree of its neighbors. A graph's assortativity coefficient AS is a value in [-1,1] calculated as the Pearson correlation coefficient of the degrees of all connected node pairs in the graph. A positive value means that nodes tends to connect with others with similar degrees, and a negative value means the contrary [29]. Finally, the s-metric captures the joint degree distribution of a graph as the probability that high degree nodes inter-connect with each other [21]. We compute all three metrics for all synthetic and target graphs. Given the consistency of our results, we omit results for the s-metric, which can be viewed as a subset of the assortativity distribution.

Clustering Coefficient (CC). Clustering coefficient measures whether social graphs conform to the small-world principle [34]. It is defined as the ratio of the number of links that exist between a node's immediate neighborhood and the maximum number of links that could exist. For a node x with degree d_x, at most $d_x(d_x - 1)/2$ edges can exist among x's friends (when they form a complete clique). Let k_x be the actual number of edges among x's friends, the clustering coefficient of node x is $\frac{2k_x}{d_x(d_x-1)}$. A graph's CC is the mean CC of all nodes. Intuitively, a high CC means that nodes tend to form highly connected subgraphs with their neighbors.

Node Separation. The degree of node separation is measured through three metrics: average path length, network radius and network diameter. Average path length refers to the average of all-pairs-shortest-paths on the social graph. The radius and diameter are calculated using the eccentricity of each node in the social graph. Eccentricity is defined as the maximum shortest-path distance between a node and any other node in the graph. Radius is the minimum of all eccentricities, while diameter is the maximum. Because computing all-pairs-shortest-paths is computationally infeasible given the size of our social graphs, we estimate the radius, diameter and average path length by determining the eccentricity of 1000 randomly selected nodes in each graph.

5.2 Graph Metric Results

We now describe the results of our fidelity tests. All of our graph computations are performed on a cluster of Dell Poweredge 1750 servers with dual-core Xeon processors. Memory intensive computations were performed on 2 Dell 2900 servers, each with a quad-core Xeon CPU and 32GB of memory.

Node Degree Distribution. We examine the node degree distribution in terms of the cumulative distribution function (CDF). The results across the four Facebook graphs are consistent. Hence in Figure 3 we only plot the results for Santa Barbara and New York. We see that dK and Nearest Neighbor are the two models that produce the most accurate degree distributions, which is confirmed by the Euclidean distance results in Table 4 under "NDD." An interesting observation is that ≈40% users have social degree of 10 or less, which is not captured by the Barabasi Albert model.

We also examined whether the node degree in the synthetic graphs follows the power-law distribution. Using the power-law curve fitting method in [9], we derive the exponent and the fitting error. Table 3 summarizes the results for Santa Barbara and New York. Again, dK-2 and Nearest Neighbor outperform other models.

Graph	Model	Estimated Power-law exp.	Power-law Fitting error
Santa Barbara 12,814 nodes 92,241 edges	Real Graph	1.50	0.27
	dK-2	1.50	0.27
	Nearest Neighbor	1.50	0.28
	Random Walk	1.50	0.37
	KronFit	1.50	0.24
	Forest Fire	1.53	0.19
	Barabasi-Albert	2.82	0.007
New York 377,712 nodes 3,616,873 edges	Real Graph	1.50	0.33
	dK-2	1.50	0.33
	Nearest Neighbor	1.50	0.37
	Random Walk	1.50	0.46
	KronFit	1.50	0.37
	Forest Fire	1.51	0.18
	Barabasi-Albert	2.86	0.006

Table 3: Examining the power-law effect in node degree distribution. dK-2 and Nearest Neighbor are the two most accurate models.

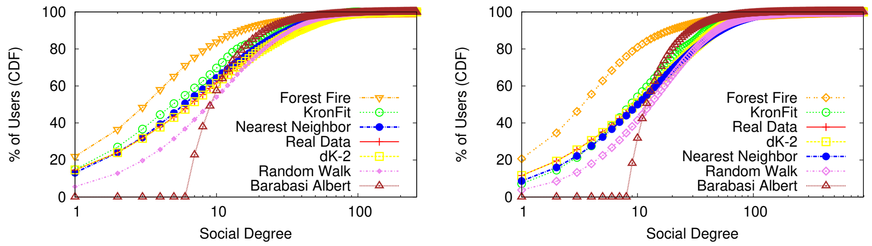
Joint Node Degree Distribution. Next we compare the real and synthetic graphs in terms of node connectivity, particularly on metrics of joint node degree distribution (k_{nn} and assortativity). These metrics have been used to verify whether a graph displays *scale-free* and *small-world* properties. We represent k_{nn} as a vector over the node social degree, and use the vector-based Euclidean distance values in Table 4 to represent the statistical difference between the real and synthetic graphs.

Assortativity is a scalar value representing the same property as k_{nn}, and reflects the same relative results. For scalar metrics like assortativity, network diameter, and average path length, we compare the actual values of both synthetic and target graphs in Table 5.

The k_{nn} and assortativity results show that the dK-2 model is consistently accurate across all four target graphs, while the Nearest Neighbor model displays some visible differences. A closer look at the detailed k_{nn} values (omitted for brevity) shows that the Nearest Neighbor model displays a similar trend in k_{nn} as the real graphs, but produces larger k_{nn} values at nodes with higher social degree. This implies that the model tends to inter-connect nodes with higher social degrees, which can be explained by the fact that the model connects nodes to 2-hop neighbors and most popular nodes are within 2 hops from each other. This is confirmed by its higher assortativity values in Table 5.

Clustering Coefficient. Prior work [34] shows that social networks have a local clustering structure, *i.e.* neighbors of a node in the social graph tend to connect to each other as well. This is particularly true for nodes with low social degrees. Figure 4 shows the clustering coefficient as a function of the social degree for three networks. We omit the results for Monterey Bay because they are highly consistent with those of Santa Barbara. The Euclidean distance values associated with all four networks are shown in Table 4 under the column "CC."

In this case, Nearest Neighbor, Random Walk and Forest Fire are the top three models. They all follow the general trend in the target graphs: nodes with small social degrees experience heavier clustering, and the degree of clustering decreases with the node degree. This is unsurprising, given the model definitions where all three encourage forming local triangles by connecting 2-hop neighbors (Nearest Neighbor) or connecting new nodes to well-connected subgraphs (Random Walk and Forest Fire). On the other hand, the other three models (dK-2, KronFit and Barbasi-Albert)



(a) Santa Barbara (b) New York

Figure 3: CDFs of the node degrees of the real Facebook graphs and those generated by the six network models. dK-2 and Nearest Neighbor models closely match the real data.

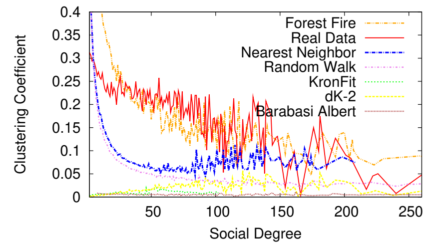


Figure 4: CC in Santa Barbara as a function of social degree. The best models are Forest Fire, Nearest Neighbor and Random Walk.

all produce flat clustering coefficient around 0.05 or less, and fail to capture any local clustering effects.

Node Separation Metrics. Finally, we look at how models capture the separation between nodes through the network diameter and average path length metrics. We find that dK-2 is highly accurate, while Forest Fire also performs well. Nearest Neighbor, however, produces significantly more clustered graphs, resulting in shorter path lengths and a smaller network diameter. We attribute this to Nearest Neighbor's focus on preferential attachment which increases connections between highly connected nodes.

5.3 Summary of Observations

The above results do not tell us in absolute terms how significantly different synthetic graphs are compared to the original graphs. Relatively speaking, however, we see that the dK-2 and Nearest Neighbor models provide a relatively accurate representation of the target graphs. On the other hand, some models do not accurately capture particular individual metrics well. The dK-2 model is especially accurate in capturing the individual and joint degree distributions, but fails to capture the key feature of local clustering. The Nearest Neighbor model is consistently accurate in terms of the degree distribution and clustering coefficients, but is biased towards inter-connecting high-degree nodes, and produces graphs with significantly shorter path lengths and network diameter. This results in higher assortativity values that may diverge from graphs with more heterogeneous connections like Egypt[2].

Our results are promising, because they show that despite variances across models and graphs, two models (Nearest Neighbor and dK) stand out for their ability to capture graph metrics. If these metrics are indicative of application performance, then we expect these two to also show high fidelity in our application benchmarks.

6. APPLICATION FIDELITY BENCHMARKS

Since we do not yet understand how graph metrics impact different social applications, the final measure of a model's fidelity must still rely on application-level benchmarks. We implement the algorithms from several social network applications, run tests with both target graph and synthetic graphs as input, and compare the results to quantify each model's fidelity. In addition, these results allows us to identify whether the existing social graph metrics fully capture the "important features" in social networks.

We chose "Reliable Email" [13], "Sybilguard" [36] and a "Social Shield Anonymous System" [30] as representative social network applications. All are recent research systems that leverage graph properties in social networks to address network security problems. Compared to known graph metrics, all these application tests present new perspectives, since their performance on a particular graph cannot be easily correlated with a single graph metric[3]. Examining how synthetic graphs compare in these application tests versus their original counterparts sheds light on whether today's models are accurate enough to replace actual social graphs with graph models for experimental research.

RE: Reliable Email. RE [13] is a whitelist system for email that securely marks emails from a user's friends and friends-of-friends as non-spam messages, allowing them to bypass spam filters. Friends in a social network securely attest to each others' email messages while keeping users' contacts private.

A meaningful evaluation experiment is to examine the level of potential impact on RE users if accounts in the social network were compromised using phishing attacks. Compromised accounts can flood spam email through the RE system, since their spam bypasses filters and directly reaches user's inboxes. Our RE simulation measures the portion of the entire user population receiving spam as we increase the number of compromised accounts.

We perform these experiments on our Facebook social graphs, and plot the results for Santa Barbara, Egypt, and New York in Figure 5, and list Euclidean distance values for all 4 graphs in column 7 of Table 4. Comparing results across all graphs, Nearest Neighbor produces the overall best results, with dK-2 and Random Walk as the next best models. It is notable that the best model varies across our graphs, perhaps due to specific structural features in each of the Facebook graphs. One take-away from this experiment is that application level results cannot be easily explained using a single graph metric. In general, the accuracy of the RE experiment's performance on a synthetic graph is not strongly correlated with any of the metrics we track in Table 4.

Sybilguard. A malicious user in an online community can launch a Sybil attack [10] by creating a large number of virtual identities. These identities can then work together to provide the owner with some unfair advantage, by outvoting legitimate users in consensus systems, corrupting data in distributed storage systems, or manipulating incentive systems or reputation systems to perform fraud. SybilGuard [36] proposes a way to detect these Sybil identities using social networks. The main insight of this defense relies on the fact that it is difficult to make multiple social connections between Sybil identities and legitimate users. Because of this concrete obstacle, Sybil identities tend to form a strongly connected subgraph with a small number of links to honest users.

Using Sybilguard, a node A seeking to determine if node B is a Sybil identity sends a number of random walks. Node B does

[2]Unlike Egypt, all of our more than 20 Facebook graphs have AS values between 0.05 and 0.25.

[3]The authors of Sybilguard credit their functionality to the Mixing Time property of graphs. We cannot confirm this, since current Mixing Time algorithms do not scale to large graphs.

Graph	Models	Euclidean Distance: Target vs. Synthetic Graphs				
		Graph Metrics			Applications	
		NDD	K_{nn}	CC	RE	Sybil.
	dK-2	**21.43**	**62.82**	2.24	117.97	22.13
Monterey Bay	Nearest Neighbor	30.43	77.73	1.42	**59.62**	**15.23**
6,283 nodes	Random Walk	51.96	80.31	1.94	313.07	22.15
33,969 edges	KronFit	34.30	108.91	2.35	233.81	23.05
	Forest Fire	53.24	267.78	**0.91**	350.73	58.67
	Barabasi-Albert	93.03	132.52	2.22	529.38	17.28
	dK-2	**1.70**	**25.72**	2.15	134.31	7.90
Santa Barbara	Nearest Neighbor	16.73	155.58	1.43	**77.36**	5.56
12,814 nodes	Random Walk	38.64	146.86	1.71	310.00	**5.30**
92,241 edges	KronFit	44.66	173.08	2.06	139.71	8.87
	Forest Fire	93.67	336.97	**1.01**	815.09	76.45
	Barabasi-Albert	103.34	167.85	2.13	757.64	10.95
	dK-2	**15.56**	**178.58**	1.35	**306.57**	6.73
Egypt	Nearest Neighbor	29.35	1147.31	**0.76**	399.90	**4.68**
246,692 nodes	Random Walk	64.03	537.51	0.98	1673.20	6.36
1,618,085 edges	KronFit	80.16	7680.46	1.34	2113.03	30.39
	Forest Fire	68.51	6932.16	2.33	2785.04	72.78
	Barabasi-Albert	98.60	399.31	1.25	2990.49	24.24
	dK-2	**1.34**	106.45	1.53	**392.85**	5.99
New York	Nearest Neighbor	15.63	1281.98	**0.97**	410.58	13.89
377,712 nodes	Random Walk	42.18	1760.65	1.11	1462.96	28.91
3,616,873 edges	KronFit	31.83	373.65	1.46	416.47	30.08
	Forest Fire	133.01	6741.84	1.67	5177.69	8.02
	Barabasi-Albert	117.39	**92.49**	1.44	3686.74	**5.78**

Table 4: Euclidean distances between model-generated graphs and the original graphs for several graph metrics and application benchmarks. Each point is the average of 20 **synthetic to original graph comparisons. For each metric, the value with the lowest error is underlined and in bold, while the second best model is underlined. Overall, Nearest Neighbor is consistently accurate for most metrics.** dK**-2 is highly accurate for node degree distribution (NDD) and joint node degree distribution (**k_{nn}**), but not for clustering coefficient (CC) and application benchmarks.**

Graphs	Exact metric values		
	AS	Diam.	Path Leng.
Monterey Bay	0.29	14	5.09
dK-2	**0.28**	11.95	**4.94**
Nearest Neighbor	0.32	8.2	3.55
Random Walk	0.08	8.55	3.35
KronFit	0.01	8.45	3.79
Forest Fire	0.15	**15.95**	4.77
Barabasi-Albert	-0.03	5	3.23
Santa Barabara	0.24	13	4.31
dK-2	**0.24**	**11.3**	**4.76**
Nearest Neighbor	0.38	7.35	3.26
Random Walk	0.07	8.15	3.20
KronFit	0.15	9.9	3.83
Forest Fire	0.15	16.0	4.69
Barabasi-Albert	-0.021	5.0	3.02
Egypt	0.006	15	4.91
dK-2	**0.005**	12.9	**5.36**
Nearest Neighbor	0.40	8.05	3.43
Random Walk	0.05	9.6	3.46
KronFit	0.084	11.45	4.22
Forest Fire	0.098	**14.65**	4.03
Barabasi-Albert	-0.009	5.0	3.50
New York	0.19	16	4.75
dK-2	**0.18**	12.7	5.42
Nearest Neighbor	0.45	7.95	3.36
Random Walk	0.034	8.45	3.23
KronFit	0.035	8.8	**4.10**
Forest Fire	0.10	**14.8**	4.05
Barabasi-Albert	-0.006	5.0	3.21

Table 5: Comparing the original graphs and their synthetic counterparts w.r.t. assortativity, network diameter and average path length. Results shown are the exact metric value, and results for synthetic graphs are averages over 20 **graphs.**

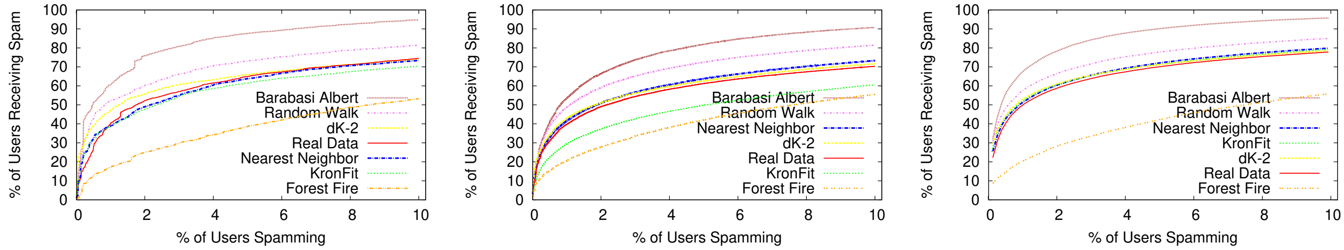
the same, and A records the number of intersections between these random walks. If B is a Sybil identity, it is likely to be in a local subgraph with a small number of paths to A, resulting in a small number of walk intersections. Otherwise, the number of intersections will be high. The rate of success depends on the length of the random walks. If the walks are too short then they might not intersect, and A would have less information about B. Our experiment looks at the portion of random walks that result in intersections as a function of the length of random walks.

Looking at the resulting plots in Figure 6 and Table 4, we see that Nearest Neighbor again performs very well, producing the most accurate synthetic graphs for 3 of the 4 target graphs. We note that the simple Barabasi-Albert model, which consistently produces inaccurate graphs (measured by graph metrics), actually performs relatively well in the Sybilguard tests with Euclidean distances on par or even lower than other models. Again, our results reinforce the idea that application-level benchmarks do not easily map to known graph metrics, and they must be included in any attempts to understand the fidelity of graph models.

Social shields for anonymous communication. Puttaswamy et al. propose using social neighborhoods to protect users of anonymous communication protocols against passive logging attacks [30]. Most anonymous routing protocols provide anonymity by forwarding traffic through a random sequence of relay nodes. In practice, however, malicious relays that observe traffic in the network over long periods can probabilistically guess the identity of the communication source. To protect themselves, a communication source in the proposed system first relays traffic through a random sequence of friend nodes, such that any passive logging attack will not be able to distinguish it from its friend nodes. The solution provides the strongest protection when the user is in a large clique in the social network [30].

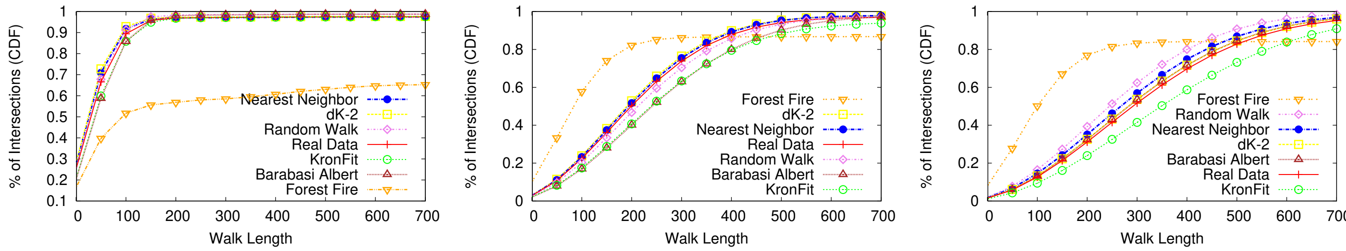
Our experiment measures the size of the largest clique each user is a part of. Again, this application exploits a graph property that is not captured by any of the previously analyzed metrics. The closest related metric is the clustering coefficient, which quantifies the level of connectivity within each user's one-hop neighborhood.

Figure 7 shows two interesting results. First, we see that dK-2 consistently failed to capture the formation of larger cliques in its synthetic graphs. This is somewhat intuitive, since dK-2 captures only joint degree distribution, and not the clustering coefficient. Given its poor correlation with the clustering coefficient (Figure 4), it is clear that dK-2 forms fewer and smaller cliques than the other models. We assume that if a graph generator existed for the dK-3 model, it would do a much better job of capturing clustering coefficients as well as clique properties. Second, our results for Santa Barbara show that the Forest Fire model produces clique properties most similar to the original graph. This is due to the large number of local connections that Forest Fire introduces with each new node. However, this heavy local clustering significantly skews other metrics, making Forest Fire the least accurate of all our models for both the Sybilguard and RE tests. In fact, Forest Fire produces so many local edges that our relatively efficient maximal clique search algorithm failed to complete on Forest Fire graphs modeled after the Egypt and New York graphs. For each of these graphs, our algorithm takes more than 2.5 weeks to produce a result. In comparison,



(a) Santa Barbara (b) Egypt (c) New York

Figure 5: Penetration of spam in a social graph running RE as function of number of spammers in the network.



(a) Santa Barbara (b) Egypt (c) New York

Figure 6: Portion of all random walks resulting in intersections in Sybilguard, as a function of random walk length.

our clique search algorithm running on other models for Egypt and New York all completed in less than 30 minutes. Thus, while we expect Forest Fire to outperform other models for the clique test in Egypt and New York, its large errors in RE and Sybilguard make it unsuitable for our purposes.

Nearest Neighbor, instead, provides a more consistent accuracy both across statistical metrics and application level results, which is also confirmed in this experiment. Although it is not extremely precise in the number of maximum cliques, it performs consistently well over multiple datasets.

Interaction Graphs. We have also investigated another dataset, the "interaction graph" from the New York region, as proposed in [35]. An interaction graph is a social graph in which edges that do not receive interactions between the two endpoints are culled. In our case, interactions are defined as wall posts or photo comment activity. Intuitively, this process removes unimportant edges from the graph, leaving only active edges that are relevant when designing and testing "user-driven" applications.

The interaction graph for NY has 254599 nodes and 926165 edges (a 75% reduction in edges compared to the full social graph). We do not present detailed results regarding the accuracy of the models on this graph, because they lead us to the same conclusions derived from other analyzed datasets. For instance, both clustering coefficient and degree distribution of the interaction graph look extremely similar to those of the full social graph for Egypt. In general, dK-2 and Nearest Neighbor consistently produce the best results on the interaction graph.

Final Considerations. A final take-away from our tests is that despite significant variance in model accuracy, we find that Nearest Neighbor consistently outperforms its competitors in producing synthetic graphs that not only capture the majority of known graph metrics, but also accurately predict the performance of application-level tests such as RE, Sybilguard, and Social Shields. Based on our graph metric and application-level tests, we conclude that the Nearest Neighbor model is a viable candidate for researchers looking to replace real graphs with model-generated graphs.

7. RELATED WORK

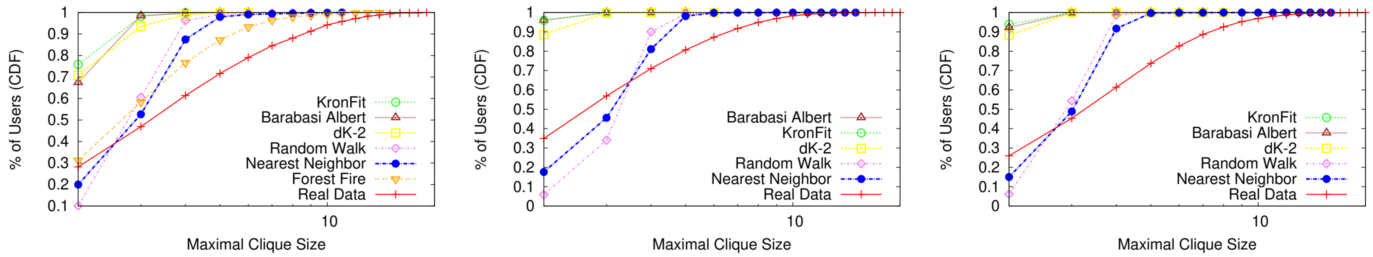
Trace-driven models. Trace-driven network models are popular in research areas where active measurements are difficult to perform, including wireless networks [16], mobile networks [15], and Internet backbone traffic [6]. Researchers continue to rely on synthetic model-generated traffic traces for experimental research, even as they recognize and continue to reduce the inherent error introduced by these models [15].

OSN measurements. Several important measurement studies of online social networks helped derive and shape the key graph metrics we use in our study. Some of them focus on static properties by looking at data sets collected at a single time point [2, 3, 25], while others investigate dynamic properties from a series of data sets over time [18, 20, 26]. These studies found a collection of remarkable properties such as the power-law scaling characteristics, the small-world phenomena, and clustered community structures.

Graph similarity. A number of techniques have been proposed to quantify graph similarity, including graph isomorphism, edit distance [24], common subgraphs and supergraphs, and statistical measurements of graph structure. We chose to use a statistical approach for our study because most of the alternative methods were computationally intractable for our large graph datasets.

8. DISCUSSION AND CONCLUSIONS

We began this work as a search for practical solutions to challenges we faced while distributing measured social graphs to colleagues in the research community. While experiments using trace-driven models are common in the study of both wired and wireless networks, an analogous approach has not been applied to research on social graphs. It became clear to us that measurement-calibrated graph modeling faced a number of challenges due to the inherent complexity and scale of graphs. Our most important contributions are proposing this approach to experimental research on social graphs, identifying inherent challenges, and proposing a number of simple but feasible solutions.



(a) Santa Barbara (b) Egypt (c) New York

Figure 7: CDF graphs showing the size of the largest clique each user belongs to in the Monterey Bay, Santa Barbara, and New York graphs.

Through empirical experimentation, we find that structure-driven models such as dK and Kronecker are limited by high computational and memory complexity. The most consistently accurate model is our modified version of Nearest Neighbor, which despite its simple algorithm, manages to successfully capture key graph metrics of the original graphs. It also produces generally accurate results in our application-level tests, making it a viable candidate for researchers looking to replace real graphs with model-generated graphs. We conclude that graph models can be adequate replacements for real social graphs, and that current graph metrics cannot completely capture properties used by social network applications.

More work needs to be done to further validate and expand these initial findings. A logical next step is to investigate these applications to better understand the properties they rely on for success, and if these properties correspond to yet unknown graph metrics. We also need to verify our conclusions using simulations of more social applications. Finally, more work needs to be done to minimize the operational overheads of structure-driven models such as dK. Then we will learn if, given sufficient computational resources, they can generate the most representative synthetic graphs.

Acknowledgments. We thank Jure Leskovec for sharing code and guidance on calibrating graphs in the Kronecker and Forest Fire models, and the anonymous reviewers for their helpful comments. This work is supported in part by NSF Grants CNS-0916307, IIS-0847925, CNS-0832090, CNS-0546216. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

[1] *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 2004.

[2] ADAMIC, L. A., BUYUKKOKTEN, O., AND ADAR, E. A social network caught in the web. *First Monday 8*, 6 (2003).

[3] AHN, Y.-Y., ET AL. Analysis of topological characteristics of huge online social networking services. In *Proc. of WWW* (May 2007).

[4] BACKSTROM, L., DWORK, C., AND KLEINBERG, J. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW* (May 2007).

[5] BARABASI, A.-L., AND ALBERT, R. Emergence of scaling in random networks. *Science 286* (1999), 509–512.

[6] BARAKAT, C., ET AL. A flow-based model for internet backbone traffic. In *Proc. of Internet Measurement Workshop* (2002).

[7] BARBARO, M., AND ZELLER, T. A face is exposed for AOL searcher no. 4417749, August 2006. NY Times.

[8] BLUM, A., CHAN, T.-H. H., AND RWEBANGIRA, M. R. A random-surfer web-graph model. In *Proc. of ANALCO* (2006).

[9] CLAUSET, A., SHALIZI, C. R., AND NEWMAN, M. E. J. Power-law distributions in empirical data. *SIAM Review 51*, 4 (2009), 661–703.

[10] DOUCEUR, J. R. The Sybil attack. In *Proc. of IPTPS* (March 2002).

[11] ERDOS, P., AND RENYI, A. On the evolution of random graphs. *Mathematical Institute of the Hungarian Academy of Science* (1960).

[12] FLOYD, S., AND KOHLER, E. Internet research needs better models. In *Proc. of HotNets* (Oct. 2002).

[13] GARRISS, S., ET AL. Re: Reliable email. In *Proc. of NSDI* (2006).

[14] IVONA, B., ADAM, K., AND RAHUL, S. Graph model selection using maximum likelihood. In *Proc. of ICML* (2006).

[15] KIM, M., KOTZ, D., AND KIM, S. Extracting a mobility model from real user traces. In *Proc. of INFOCOM* (April 2006).

[16] KONRAD, A., ZHAO, B. Y., JOSEPH, A. D., AND LUDWIG, R. A markov-based channel model algorithm for wireless networks. *ACM Wireless Networks 9*, 3 (May 2003), 189–199.

[17] KUMAR, R., ET AL. Stochastic models for the web graph. In *Proc. of FOCS* (2000).

[18] KUMAR, R., NOVAK, J., AND TOMKINS, A. Structure and evolution of online social networks. In *Proc. of ACM KDD* (2006).

[19] LESKOVEC, J., AND FALOUTSOS, C. Scalable modeling of real graphs using kronecker multiplication. In *Proc. of ICML* (2007).

[20] LESKOVEC, J., KLEINBERG, J., AND FALOUTSOS, C. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proc. of ACM KDD* (2005).

[21] LI, L., ET AL. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Math 2*, 4 (2005), 431–523.

[22] LUDWIG, R., KONRAD, A., AND JOSEPH, A. Optimizing the end-to-end performance of reliable flows over wireless link. In *Proc. of MobiCom* (Seattle, WA, 1999).

[23] MAHADEVAN, P., ET AL. Systematic topology analysis and generation using degree correlations. In *Proc. of SIGCOMM* (2006).

[24] MESSMER, B., AND BUNKE, H. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Trans. Pattern Analysis and Machine Intelligence 20* (1998), 493–504.

[25] MISLOVE, A., ET AL. Measurement and analysis of online social networks. In *Proc. of IMC* (San Diego, CA, Oct 2007).

[26] MISLOVE, A., ET AL. Growth of the flickr social network. In *Proc. of WOSN* (Seattle, WA, August 2008).

[27] NARAYANAN, A., AND SHMATIKOV, V. How to break anonymity of the netflix prize dataset. In *Proc. of IEEE S&P* (May 2008).

[28] NARAYANAN, A., AND SHMATIKOV, V. De-anonymizing social networks. In *Proc. of IEEE S&P* (May 2009).

[29] NEWMAN, M. E. J. Mixing patterns in networks. *Physical Review E 67-026126* (2003).

[30] PUTTASWAMY, K. P. N., SALA, A., AND ZHAO, B. Y. Improving anonymity using social links. In *Proc. of NPSec* (October 2008).

[31] RUSSELL, S., AND NORVIG, P. *Artificial Intelligence: A Modern Approach*, second ed. Prentice Hall, 2003.

[32] TOIVONEN, R., ET AL. A model for social networks. *Physica A: Statistical and Theoretical Physics 371*, 2 (Nov. 2006), 851–860.

[33] VAZQUEZ, A. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E 67-056104* (2003).

[34] WATTS, D. J., AND STROGATZ, S. Collective dynamics of 'small-world' networks. *Nature*, 393 (1998), 440–442.

[35] WILSON, C., BOE, B., SALA, A., PUTTASWAMY, K. P. N., AND ZHAO, B. Y. User interactions in social networks and their implications. In *Proc. of EuroSys* (April 2009).

[36] YU, H., ET AL. Sybilguard: defending against sybil attacks via social networks. In *Proc. of SIGCOMM* (September 2006).