Time-Based Sampling of Social Network Activity Graphs

Nesreen K. Ahmed, Fredrick Berchmans, Jennifer Neville, and Ramana Kompella

Computer Science Department, Purdue University
West Lafayette, IN 47907
{nkahmed, fjohnber, neville, kompella}@cs.purdue.edu

ABSTRACT

While most research in online social networks (OSNs) in the past has focused on static friendship networks, social network activity graphs are quite important as well. However, characterizing social network activity graphs is computationally intensive; reducing the size of these graphs using sampling algorithms is critical. There are two important requirements—the sampling algorithm must be able to preserve core graph characteristics and be amenable to a streaming implementation since activity graphs are naturally evolving in a streaming fashion. Existing approaches satisfy either one or the other requirement, but not both. In this paper, we propose a novel sampling algorithm called Streaming Time Node Sampling (STNS) that exploits temporal clustering often found in real social networks. Using real communication data collected from Facebook and Twitter, we show that STNS significantly out-performs state-of-the-art sampling mechanisms such as node sampling and Forest Fire sampling, across both averages and distributions of several graph properties.

Keywords

Online Social Networks, Graph Sampling, Link Analysis

1. INTRODUCTION

Online social networks (OSNs) have witnessed tremendous growth and popularity over the recent years, with several OSNs (*e.g.*, Myspace [2], Facebook [1]) routinely comprising millions of users today. The huge success and increasing popularity of social networks makes it important to characterize and study their behavior in detail, as such studies can drive the development of appropriate tools and systems to manage and maintain these OSNs effectively, as well as lead to the development of new communication models and behavioral theories in social sciences.

Recent work (*e.g.*, [22, 20, 18, 16, 5, 6]) in analyzing online social network data has focused primarily on either static social network structure (*e.g.*, a fixed network of friendship links) or evolving social networks (*e.g.*, a network where friendship links are added over time). However, OSNs are more than just a record of social network ties—popular OSN sites provide infrastructure for community formation and mechanisms to maintain community over time by facilitating communication (*e.g.*, Wall postings), content sharing (*e.g.*, photographs), and other forms of activities.

Studying social network activity graphs (communications graphs overlaid on the friendship graphs) is as important as, if not more than, studying static friendship networks. First, activity graphs are often more relevant since they reflect the current state of the social network [26]. User activity is also a better predictor of the strength of ties between users [13]. Second, activities within OSNs directly impact the systems infrastructure, in terms of server workloads and Internet traffic conditions; thus studying activity graphs is critical for infrastructure provisioning and management. Third, they may possess very different characteristics compared to the friendship networks [27]. For example, a particular user may have a large number of links in the friendship network, but may communicate only with a few.

Studying social network activity graphs can be quite challenging. Although social activity graphs typically only comprise a subset of nodes from the original graph, the number of edges can be orders of magnitude larger, as users interact repeatedly over time. Facebook, for instance, recently reported that the number of chat messages has exceeded a billion per day [10]. The large size of these graphs makes it computationally challenging to process and characterize (*e.g.*, compute degree, path length and other distributions) such graphs in detail. It is, therefore, important to decrease the size of the graph in order to contain the complexity of analysis; a standard approach to achieving this is to *sample* the original graph.

There are essentially two main requirements for any sampling algorithm for social network activity graphs. First, sampling algorithm should preserve (most of) the characteristics of the original graph, otherwise, the complexity benefits are not worth the reduction in accuracy. Second, since social network activity graphs often evolve in a *streaming* fashion as users communicate over time, an activity graph is simply a stream of edges between nodes (users) over time. It is, therfore, computaionally inefficient for any sampling algorithm that requires to access the entire graph which covers the full activity timeline; incrementally computable sampling algorithms are therefore more preferred.

Unfortunately, most existing algorithms are good at either preserving graph properties well or are easily amenable to streaming implementations, but *not* both. For instance, simple intuitive algorithm such as edge sampling while is easy to implement in a streaming fashion, is not good at preserving graph properties, because it is heavily biased towards high-volume nodes. Node sampling while is capable of preserving some statistics efficiently (*e.g.*, cluster coefficient), is not amenable to a streaming implementation.

MLG '10 Washington, DC USA

Random-walk-based sampling algorithms such as Forest Fire [19] perform much better than both node and edge sampling. However, they also require the entire graph to begin with.

In this paper, we propose a new sampling algorithm called Streaming Time Node Sampling (STNS) that is based on key observations made from a year worth of data collected from Facebook Wall data, involving users in the Purdue University network. These observations include, 1) the extent of activity is positively correlated with degree; 2) overall clustering is relatively consistent over time; 3) many users are infrequently active; and, 4) communication among friends is relatively sparse.

We evaluate our proposed algorithm on real-world activity graphs from both Facebook and Twitter. In comparison with existing schemes such as FFS, we find that STNS has significantly lower error. Even across distributions of various graph metrics, we find that samples produced by STNS typically are more closer to original graphs than those obtained by FFS.

Thus, the main contributions of our paper are as follows.

- We present a detailed characterization study (in Section 3) of the communication behavior by collecting a year worth of Facebook Wall data from the Purdue campus network.
- We propose a new sampling algorithm called STNS (described in Section 4) that is based on the key insights derived from the above characterization study. STNS is simple, preserves important graph characteristics well, and can be implemented in a streaming fashion.
- We present detailed comparison (in Section 5) of our algorithm with node sampling and Forest Fire [19]. We find that STNS performs much better both point statistics (*e.g.*, mean) and entire distributions.

2. BACKGROUND

In this section, we formally state the sampling problem and outline briefly a few state-of-the-art sampling mechanisms.

2.1 Problem definition

Let $G(V, E_{(0,T)})$ represent the social activity graph, where V is the set of nodes and $E_{(0,T)}$ is the set of edges in the graph. Let $(0, T)$ represent the activity timeline, defined as the time interval during which these activity edges occur. Each edge $e \in E_{(0,T)}$ can be described as a tuple of the form (v_i, v_j, t) where $v_i, v_j \in V$ and $t \in (0, T)$ represents the time step at which this activity edge occurred within the activity timeline. Given a sampling fraction ϕ, the goal is to create a sample graph $G_s(V^s, E^s_{(0,T)})$ such that $|V_s|/|V| = \phi$, that preserves or scales the properties of the original network.

We focus mainly on the degree, path length and clustering coefficients. Further, we are interested not only in the point statistics such as mean, but also in the distributions of these properties. The degrees of nodes of the network is the easiest way to characterize a graph. The distribution of clustering coefficient captures the local topological features of the graph. The distribution of path length captures the global topological features of the graph. Our goal is to devise a sampling method which preserves these properties in the sampled graphs.

2.2 Current sampling methods

Traditional sampling techniques can be broadly classified as node-based, edge-based, and topology-based techniques. We summarize these techniques in Table 1 in terms of their ability to preserve graph properties and to implement in a streaming fashion.

Algorithm	Preserves Properties	Amenable to Streaming
Edge sampling	XX	✓
Node sampling	X	X
Forest Fire	✓	X
STNS (this paper)	✓✓	✓

Table 1: Summarizing properties of sampling algorithms.

Node sampling (NS). In classic node sampling, nodes are chosen independently and uniformly at random from the original graph for inclusion in the sampled graph. Then the sample include all the edges among the sampled nodes. Clearly, node sampling as described is not amenable to a streaming implementation because, at any given instant a node is selected, all the edges incident on the node are required, even those that have happened in the past.

The work in [24] shows that uniform node sampling does not accurately capture power-law degree distributions. Similarly, [17] shows that although node sampling captures degree distributions due to its inclusion of all edges for a chosen node set, the original connectivity is less likely to be preserved.Thus, NS can lead to biased estimates for clustering and path length. In many other situations, however, researchers have observed that NS produces good samples [19]. Thus, we consider this in our comparison.

Edge sampling (ES). Edge sampling focuses on the selection of edges rather than nodes to populate the sample. Edges are chosen independently and uniformly at random, then the two incident nodes and the edge are added to the sample. No additional edges between the sampled nodes are added except those chosen during the random edge selection process; edge sampling therefore can be easily implemented in a streaming fashion. Edge-based sampling can accurately capture the path length distributions, due to its bias towards high degree nodes and the inclusion of both end points of selected edges. However, overall clustering and connectivity is less likely to be preserved due to the independent sampling of edges [17]. In addition, the method generally produces sparse graphs. Due to these limitations, ES typically does not produce accurate samples, thus we do not consider ES further in this paper.

Topology-based sampling. There are many topology-based sampling methods. One example is snowball sampling, which selects nodes using breadth-first search from a randomly selected seed node. Snowball sampling accurately maintains the network connectivity within the snowball , however it suffers from a *boundary bias* in that many peripheral nodes (*i.e.* those sampled on the last round) will be missing a large number of neighbors.

Random-walk based sampling methods are considered as another class of topology-based sampling methods, which use the natural connectivity of the graph to select nodes and edges. In [19], Leskovec *et al.* analyze various sampling algorithms for sampling large graphs. They propose a Forest Fire Sampling (FFS) method, based on their previous work analyzing temporal graph evolution [21]. FFS is a hybrid combination of snowball sampling and random-walk sampling that has been shown to produce quite accurate samples in practice. We therefore focus on FFS as the main competing approach.

The FFS algorithm starts by picking a node uniformly at random and adding it to the sample. Then the algorithm 'burns' a fraction of the outgoing links with the nodes attached to them. The fraction is a random number drawn from a geometric distribution with mean $p_f/(1-p_f)$). (For the experimental comparisons in this paper, we use $p_f = 0.7$ as recommended by the authors [19]). This process is recursively repeated for each neighbor that is burned until no new

Graph Metric	Facebook	Twitter
Nodes	50096	8581
Edges	1388122	45933
Size of giant component	49893	8214
Diameter	12	16
Avg. path length	4.33	5.18
Density	0.0003	0.0007
Clustering coefficient	0.061	0.061

Table 2: Characteristics of Facebook and Twitter networks.

node is selected to be burned. If that occurs, a new node is chosen at random from the graph to start the process again. The process continues until we reach the sample size.

3. ACTIVITY GRAPH ANALYSIS

While conventional graph sampling algorithms are applicable for many types of graphs, their ability to accurately preserve graph properties can depend on the properties of the underlying graph structure (*e.g.*, random vs. scale-free), or other characteristics of the domain (*e.g.*, rate of evolution, observability). In this work, we consider two subnetworks of real-world social activity graphs, where the subnetwork is based on a particular context in the larger graph (*e.g.*, all members of a particular group). We focus on subnetworks rather than a random sample of the larger network since they are likely to exhibit the same properties of the larger whole network. Note however that the sampling algorithms developed in this paper are generally applicable even for the larger networks.

In this section, we describe the characteristics of two different data sets we have collected to help us in the development and comparison of a new sampling algorithm. We first describe the broad characteristics of the data sets, and then focus on various temporal properties of communication activity within our data sets.

3.1 Data sets for analysis

Facebook Wall data. The first network consists of ‘mini’ communications among the set of 56,061 publicly visible Facebook users in the Purdue University network as of March 2008. Each Facebook user has a public message board called the ‘Wall’ on which other users can post small messages. From the Wall postings in the period 03/01/07–03/01/08, we construct a *Wall graph* with links from senders to different receivers. Each link is associated with the timestamp of the communication.

The (public) Purdue friendship graph is slightly larger than the Wall graph (56,061 vs. 50,096 nodes, 3 million vs. 1.4 million edges); however, in both networks the giant component consists of approximately 50,000 nodes. Table 2 gives the high level structural statistics of the Wall graph.

Twitter data. The second network consists of ‘micro’ communications among the set of 23,842 Twitter users actively involved in discussion surrounding the *United Nations Climate Change Conference* held in Copenhagen in December 2009. We consider the communication network formed by the set of 74,227 reply-to messages in the #cop15 Twitter hashtag. Each reply-to message contains a sender id, a receiver tag (*i.e.*, *@janesmith*), and a timestamp. The tweets occur over a two-week period from 12/07/09–12/18/09, which spanned the entire conference session. We construct a communication network from the set of nodes with at least one incoming reply-to message. This results in a network of 8,581 nodes and 45,933 edges. Table 2 gives the high level structural statistics of the filtered Twitter network.

Notice that while the two data sets are in distinct domains, they have an underlying commonality, namely, both these networks comprise a strong contextual component that joins together users. Given the ensuing similarity, we posit that conclusions drawn across these two data sets are likely more indicative of something fundamental about the nature of these contextual social networks.

3.2 Analysis results

To investigate the characteristics of these communication networks, we conduct a three-part measurement analysis. First, we analyze the temporal characteristics of the social network structure in both Facebook and Twitter. Second, we analyze the temporal activity of users in Facebook. Finally, we analyze temporal communication behavior between pairs of users in Facebook by characterizing the distribution of the sizes and durations and inter-arrival times of ‘conversations’ between different user pairs.

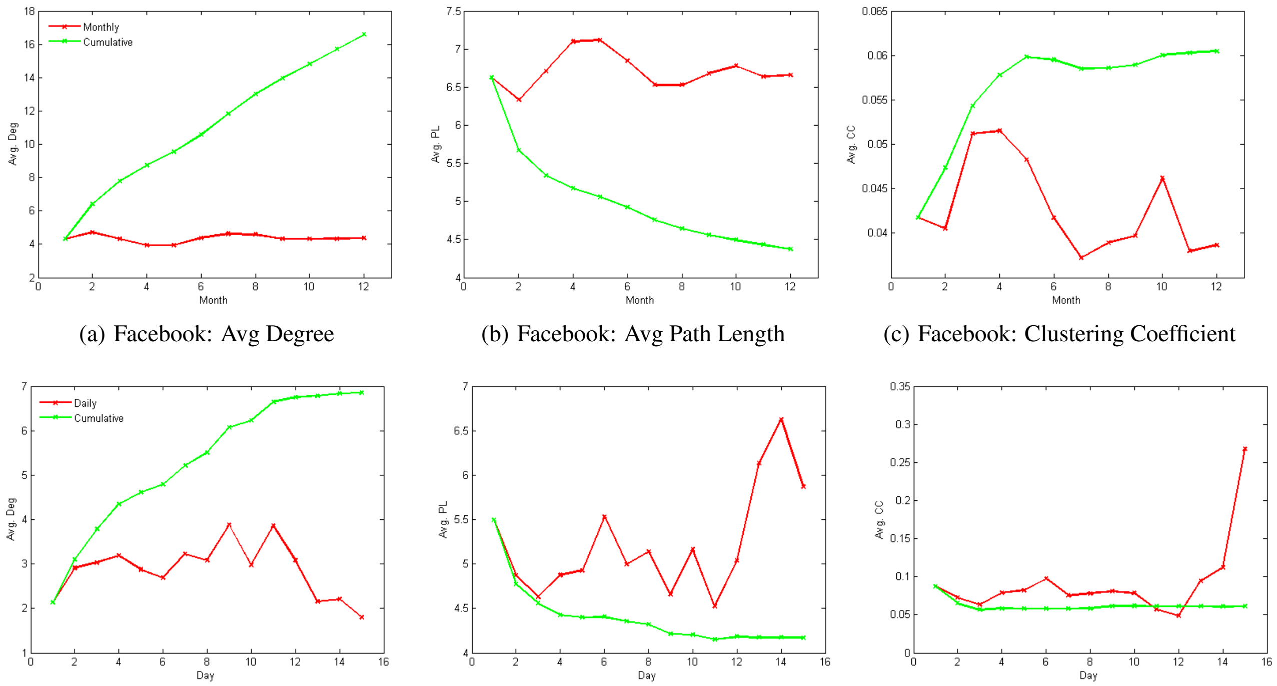
Temporal characteristics. To explore the dynamics of the network structure, we measured the temporal variation of graph statistics in both the cumulative graph as well as monthly snapshots. Figure 1 shows the temporal variation of the three point statistics: average degree, average path length, and average clustering coefficient. From Figure 1, we can see that in the cumulative graphs, average degree increases over time and average path length decreases. This is evidence of the *densification* and shrinking diameter that was observed by Leskovec *et al.* in [21]. However, we can observe that the average degree and path length remain consistently similar when we bin the activity into months. While we have not thoroughly investigated this further, we believe this is indicative of the fact that cumulative affects may be the real reason for the apparent densification.

In contrast to the change in degree and path length, clustering coefficient stabilizes after the first initial four months, and remains relatively consistent in the cumulative graph. This implies that although new nodes (and edges) increase global connectivity (*i.e.*, decrease path length), they do not increase local connectivity among neighbors.

Activity analysis. To investigate user activity over time, we measure the number of users who posted or received at least one message. (For brevity, we focus on the results involving Facebook data alone for this and the next experiment.). In order to reflect the true number of active people in the system, we age out users who do not post a message in a k week period, for $k = 6, 12$ and 18. We plot, for different values of k, the variation of number of active people in any given week in Figure 2(a). From the figure, we can observe that a large number of users are consistently present. However, as we can observe from Figure 2(b) that users are typically infrequently active, with about 80% users active for less than 5 days, not necessarily contiguous, over the entire year.

In Figure 2(c), we evaluated the correlation between users' activity and their degree. From the figure, we can observe a strong positive correlation of activity with degree ($corr = 0.85$), indicating that highly active users talk to many people.

Communication analysis. Our final characterization involves the size, duration and inter-arrival times of ‘conversations’ among users in the Facebook Wall data. We define a *conversation* to be the largest temporally ordered sequence of Wall message postings, with each subsequent message sent within a given delay Δ. This definition of a conversation groups together messages that are close in time. We consider conversations across both user-pairs and users. ‘User-pairs’ refers to conversations between each pair of users where as ‘users’ refers to conversations where we cluster together all messages which contain a given user as either the sender or the receiver



(a) Facebook: Avg Degree (b) Facebook: Avg Path Length (c) Facebook: Clustering Coefficient

(d) Twitter: Avg Degree (e) Twitter: Avg Path Length (f) Twitter: Clustering Coefficient

Figure 1: Graph statistics over time in the Facebook and Twitter networks.

with (possibly several) other senders or receivers.

Due to space limitations, we avoid showing the actual graphs and briefly outline our main observations from the analysis. In our data, we observed that more than 90% of the conversations have a duration of less than 6 hours for user-pairs, and 10 hours for users. Almost all conversations (either between user-pairs or users) are completed within a day. We also considered the number of conversations to study the long-term distribution of user activity. We observe that more than 50% of user-pairs have less than one conversation in a year. One possibility could be that the users have used the Wall as a means to start a conversation and then switch to some other means of communication (such as instant messaging, or face-to-face). The second possibility is that there could be disparity in the number of conversations a user has with different *types* of friends, depending on their rapport. One final measure we considered is the distribution of inter-conversation time. We observed that for two users who had a conversation on any given day, about 90% of pairs will start another conversation within a period of 100 days. We made a similar observation on the users' curve.

4. STREAMING TIME NODE SAMPLING

In this section, we will exploit key characteristics of temporal activity graphs discussed in the previous section to develop a time-based sampling algorithm called Streaming Time Node Sampling (STNS). STNS combines the strengths of node-based sampling for capturing degree distributions, and edge-based sampling for capturing path length distributions, while also accurately preserving overall clustering in the graph.

Specifically, the main observations from our data analysis in the previous section can be summarized as follows.

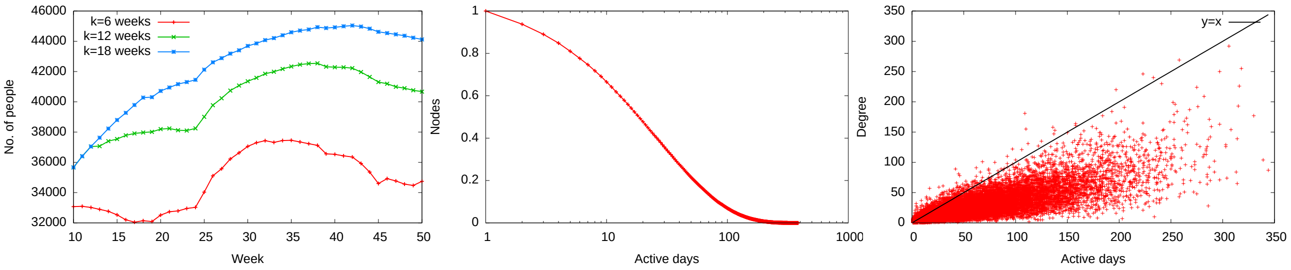
- Extent of activity throughout time is positively correlated with number of friends.
- Overall clustering is relatively consistent over time.
- A large number of users are active consistently over time, but each user individually is infrequently active. (*i.e.*, posting approximately every 6-12 weeks).
- Communication among friends are relatively sparse (*i.e.*, median inter-arrival times is 11 days).

Based on these observations, we conjecture that a time-based sampling approach can accurately capture the overall clustering of the original graph, and combined with aspects of node and edge sampling, will also accurately capture degree and path length distributions.

In our approach, nodes are selected proportionally based on their communication activity. We first sample all edges within a randomly selected time window and then populate our node set with the corresponding nodes. This edge-based sampling of nodes biases our sample towards more active nodes, but since we select both nodes involved in a communication, a set of less active nodes are also likely to be included. In addition, the selection of edges within a given time window is designed to select a subgraph with more clustering/connectivity than if we selected randomly from the entire dataset. Next, among the selected node set, we accumulate additional edges that occur due to communication outside the time window. This is based on our observation of relatively infrequent communication activity and is designed to increase the connectivity of the node set by considering their communication patterns over a longer time frame (in the future).

4.1 STNS algorithm

We consider the activity edges to be events that occur over time and arriving in a streaming fashion at specific timesteps. We define a selection time window τ of specific duration. For this work, we choose τ=1 day for the Wall network, and τ= 1 hour for the Twitter



(a) Active users with aging (b) Nodes active days distribution (c) Correlation of activity with degree

Figure 2: Posts per user and number of active users with aging on a weekly basis.

network. We start by scanning the activity timeline in a streaming fashion. We select a timestamp t on the activity timeline according to a Bernoulli process with mean $p_s = m/T$, where T is the total number of timestamps and m is the average number of timestamps needed to achieve the sampling fraction ϕ. We select the nodes incident on activity edges that occur in the next time window (*e.g.*, $[t, t+\tau]$). The process is repeated in a streaming fashion until we have collected the appropriate fraction of nodes as specified by the sampling task.

We associate a sample time st_v for each node v when it is added to the sample. Then the algorithm chooses the edges $e = (v_i, v_j, t)$ such that $t \geq st_{v_i}$ and $t \geq st_{v_j}$. In other words, we only sample activity edges that involve a node *after* it has been added to the sample. This choice enables STNS to be implemented in a streaming fashion without requiring access to the full temporal extent of the activity graph (*i.e.*, without requiring to remember all edges in the past). We formally describe the steps involved as follows.

1. Let sample node set $V_s = \emptyset$ and sample edge set $E_s = \emptyset$.
2. Scan the activity timeline for $t = 1 : T$ till $\dfrac{|V_s|}{|V|} = \phi$.
   (a) Select a timestamp t with probability $p_s = m/T$.
   (b) Let $E_{(t,t+\tau)}$ be the set of edges with timestamps in the range $[t, t+\tau]$.
   (c) For each edge $e = (v_i, v_j) \in E_{(t,t+\tau)}$:
   If $v_i \notin V_s$: $V_s = V_s \cup \{v_i\}$; Set $st_{v_i} = t_e$.
   If $v_j \notin V_s$: $V_s = V_s \cup \{v_j\}$; Set $st_{v_j} = t_e$.
3. $E_s = \{e'(v_i, v_j) \in E : v_i \wedge v_j \in V_s$ and $t_{e'} \geq st_{v_i}$ and $t_{e'} \geq st_{v_j}\}$

The sampled graph is then $G_s = (V_s, E_s)$.

4.2 Hypothetical variants of STNS

In addition to the STNS algorithm described before, we also consider two hypothetical variants for our comparison to bring out the advantages and limitations of specific design choices in STNS.

Random Time Node Sampling (RTNS). RTNS is exactly the same as STNS except for step 3. The sample node set V_s is selected in the same manner, but the edge set consists of *all* edges associated with the nodes in V_s, i.e., edges that occur both in the past and in the future: $E_s = \{e'(v_i, v_j) \in E : v_i \wedge v_j \in V_s$.

This method is included to assess the streaming aspect of STNS. In STNS we are sampling edges forward in time, yet we are using a fixed time window of edges to illustrate the performance of the algorithm, so RTNS provides a sense of where the algorithm would converge to if it acquires edges well into the distant future.

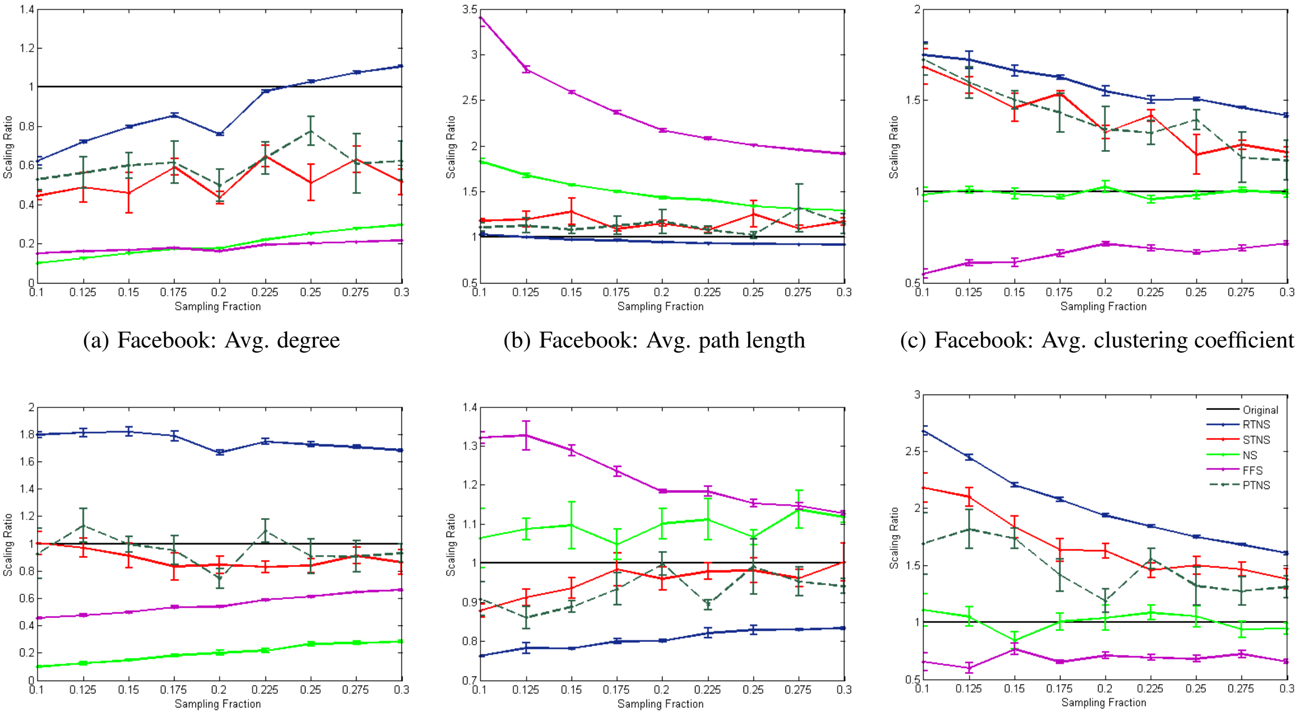
Permuted Time Node Sampling (PTNS). To isolate the effect of picking temporally co-located activity edges, we compute the STNS algorithm on a modified activity graph where the timestamps on the activity edges are randomly permuted. We call this method Permuted Time Node Sampling (PTNS). Clearly, this is a hypothetical algorithm that is designed to bring out whether choosing edges within a particular time window is necessary, or, we could just pick edges at random within the network (similar to edge sampling), include the nodes in the sample set, and any future edges that involve nodes within the sampled set.

5. EXPERIMENTAL EVALUATION

In the experimental evaluation, we mainly compare STNS to node sampling and forest fire sampling, as well as to its baseline variants, RTNS and PTNS. As mentioned before, our evaluation is primarily along three main metrics—degree, path length, and clustering coefficients. Notice that all these three metrics consist of both point statistics as well as distributions, together reflecting the core structure of a graph; our measure of goodness of a sampling algorithm includes its ability to preserve both these statistics. In contrast, metrics such as density and reciprocity are just point statistics alone; thus, we do not explicitly consider these metrics in our goodness metrics.

In our experiments, we focus on obtaining a sample between 10–30% ($\phi = 0.1$ to 0.3) of the original graph. For each sample size, instead of plotting the absolute value of the statistic, we compute the scaling fraction ρ as the ratio of the value of the statistic θ_s in the sampled graph to value in the original graph θ_o. Thus, in each of these plots, the line $\rho = 1$ will represent the value of the statistic on the original graph and is plotted for quick reference. For each sample fraction, we experiment with five different samples (generated using different random seeds) and plot the obtained scaling fractions with error bars showing the standard error. We consider node sampling (NS), forest fire sampling (FFS), and STNS. We first compare them in terms of point statistics, then in terms of distributions. Finally, we compare STNS with its variants, RTNS and PTNS.

Point-statistics. In Figure 3, we compare the average degree, path length and clustering coefficient scaling fractions of different algorithms, as we vary the sampling fraction for both Facebook and Twitter data. From the figures, we can make the following observations. First, we can observe that while all algorithms result in samples with much lower average degree compared to the actual graphs, NS and FFS produce samples that have much markedly lower average degree, although the average degree exhibits a slight increasing trend as sample size increases. This result is somewhat expected since NS does not capture all edges associated with a node



(a) Facebook: Avg. degree

(b) Facebook: Avg. path length

(c) Facebook: Avg. clustering coefficient

(d) Twitter: Avg. degree

(e) Twitter: Avg. path length

(f) Twitter: Avg. clustering coefficient

Figure 3: Comparing sampling methods on Facebook and Twitter data for different point statistics, at sampling fractions of 10-30%.

(unless the other node is also sampled). Similarly, FFS will also tend to miss out on several edges that are not burned resulting in smaller average degree. Our algorithm STNS, in contrast, performs better than the rest, especially for the Twitter data, at all sampling fractions. However, it exhibits a lot of variation due to the randomness involved in choice of the day from where it starts picking the edges for the nodes. If it picks more days at the beginning (end), the scaling fractions would be better (worse).

In terms of the average path length (shown in Figure 3(b) and 3(e)) we observe that the scaling fraction of STNS is close to one, meaning samples preserve average path length, across different sampling fractions. In contrast, the average path length in samples is about 2-3 times higher for FFS and 1.5-2 times higher for node sampling. The reason for this is that, they may not succeed in capturing all the high-degree nodes in the original graphs, leading to longer and more circuitous routes between nodes. STNS' ability to pick nodes in pairs, similar to edge sampling (discussed in Section 2), helps preserve the level of connectivity required to ensure similar path lengths as the original. Notice that while the trends appear similar across both Facebook and Twitter data sets, the exact ratios are different. In Twitter, we observe that NS and FFS only produce samples with only slightly higher average path length as opposed to STNS, compared to the Facebook data where the ratios are between 1.5–3. In addition, STNS results in slightly lower average path lengths because Twitter data is slightly more bursty in time than Facebook due to its microblogging nature.

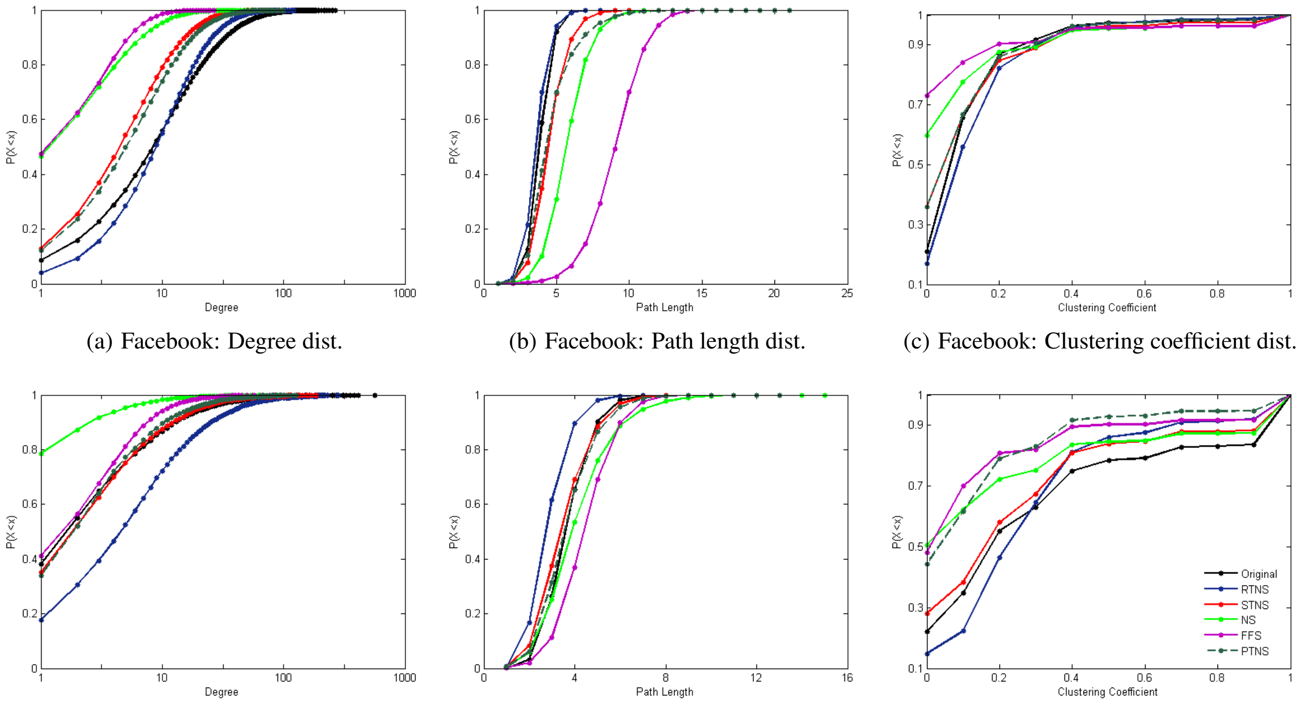
Finally, comparing the average clustering coefficients across Twitter and Facebook, we find that STNS produces graphs that are more clustered than the original one, as indicated by the higher average clustering coefficient in Figure 3(c) and 3(f). Similar to the observation in the case of average path length, we observe a slightly higher over-estimation of the clustering in the case of Twitter (almost 2×) compared to Facebook (about 1.5×). In both cases, as we move toward higher sampling fractions, clustering appears to converge toward the true value for STNS. In all cases, NS appears to be the best at preserving average clustering coefficient, but as we shall see later in Figure 4(c), the distributions don't quite match up with the original graph for NS, leading us to believe that the good results for NS are purely coincidental. FFS in all cases does comparably to STNS, except that it undersamples clustering compared to oversampling for STNS.

Distributions. While point statistics by themselves are important, they do not convey the full picture. Thus, we plot the distributions of these metrics in Figure 4 for both Facebook and Twitter at 20% sampling fraction. We note that the distributions at other sampling fractions showed similar behavior to the point statistics—at 10% sampling fraction the results were more extreme, at 30% less extreme.

Figure 4(a) and 4(d) shows the degree distribution. As discussed before, we can observe from the figures that NS picks a larger fraction of low degree nodes for inclusion in its sample. FFS also exhibits a similar characteristic in the case of Facebook, although not much for Twitter. Across both data sets, we observe STNS is more accurate at preserving degree distributions than either of NS or FFS.

In terms of path length distributions (Figure 4(b) and 4(e)), we observe that FFS samples have a much higher fraction of larger length paths compared to either NS or STNS. Among the three, STNS appears to preserve these distributions the closest, although it has a slightly higher percentage of low length paths, as we discussed during our comparison of point statistics, in the case of Twitter compared to Facebook.

The main advantage of our time-based sampling approach used



(a) Facebook: Degree dist.

(b) Facebook: Path length dist.

(c) Facebook: Clustering coefficient dist.

(d) Twitter: Degree dist.

(e) Twitter: Path length dist.

(f) Twitter: Clustering coefficient dist.

Figure 4: Comparing sampling methods on Facebook and Twitter data for distributions of graph properties at 20%

Approach	Degree	Path Length	Clustering Coeff.
Facebook			
NS	0.5033	0.6103	0.3888
FFS	0.5425	0.9264	0.5214
STNS	**0.2314**	**0.2401**	**0.1483**
Twitter			
NS	0.4037	0.1422	0.2863
FFS	0.0777	0.2885	0.3519
STNS	**0.0317**	**0.1110**	**0.0608**

Table 3: KS distances for different sampling algorithms.

in STNS is perhaps best illustrated by the distribution of clustering coefficient in Figure 4(c) and 4(f). The samples produced by STNS match clustering distribution well with the original graph, more so than FFS and NS, both of which capture a higher fraction of low clustered nodes. While NS preserved the average clustering coefficient very well (as discussed before), the distributions however are quite off compared to the original graph.

To further corroborate the fact that STNS outperforms competing sampling algorithms, we also compute the Kolmogorov-Smirnov (KS) distance between the samples output by NS, FFS and STNS for both Facebook and Twitter data in Table 3. KS distance is used to measure the agreement between two distributions; it is computed as $D = max_x|F_1(x) - F_2(x)|$, where x represents the range of the random variable and F_1 and F_2 represent two distributions. From the results in Table 3, we can observe that STNS samples represent the closest in terms of the KS distance for both Facebook and Twitter data sets. For Twitter, distance of NS from original distribution is similar to that of STNS for the path length distribution, while the KS distance of FFS is close for the degree distributions; neither, however, consistently performs well across all metrics.

Comparing STNS with RTNS and PTNS. In all the graphs, we have also plotted point statistics and distributions of the samples output by the two hypothetical variants of STNS, namely RTNS and PTNS.

Recall that RTNS include edges back in time after a particular node is selected, in contrast to STNS that considers edges only in the future. This should mean that samples produced by RTNS should have higher degree, higher clustering coefficient and lower path lengths than STNS, since STNS samples are a strict subset of RTNS. We can observe these trends clearly in the graphs in Figure 3 and Figure 4. On the whole, we observe that RTNS does worse than STNS in preserving clustering and average path lengths. On the degree distributions (Figure 4(a) and 4(d)), we find that RTNS performs better in the Facebook case than Twitter where it samples more high-degree nodes. By including edges back in time, RTNS tends more toward edge sampling; hence, its performance is not as good as STNS. By selecting edges only in the future, STNS somewhat compensates for the temporally clustered selection of edges that naturally favors the inclusion of high degree nodes.

To test whether selecting temporally clustered edges and nodes helps explain the better accuracy of STNS, we consider the variant PTNS that randomly shuffles all the edges in the activity timeline before sampling in the same fashion as STNS. Both in terms of point statistics as well as distributions, we do not observe any major differences between STNS and PTNS. The only exception is the distribution of clustering coefficients for Twitter, where we can observe that PTNS favors sampling more numbers of low clustered nodes as compared to RTNS. Curiously, there is not much difference in the case of Facebook. We believe that the reason is that activity within Twitter is often more temporally clustered than Facebook; preserving temporal locality matters more to Twitter than Facebook.

Of course, PTNS by itself is a hypothetical algorithm that cannot be implemented in practice. Still, because the differences are minor, we believe that a more realistic incarnation of PTNS, say edge-based node sampling, where we pick edges at random, include nodes that are incident on these edges, and all other edges between selected nodes in the forward direction (just like STNS) would work almost as well as STNS in practice.

6. RELATED WORK

While our work is related to the problem of sampling and analyzing large graphs in the OSN context, the problem of sampling graphs has been of interest in many different fields of research. The work in [17, 28, 24] studies the statistical properties of samples of complex networks produced by traditional sampling algorithms like node sampling, edge sampling and random-walk based sampling and discusses the biases in estimates of graph metrics due to sampling. The peer-to-peer networks research community [25, 11, 8] has used sampling methodologies to quickly explore and obtain a good representative sample of the network topology, as these networks are hard to explore completely and, are quite dynamic with significant amounts of churn in their topology. The WWW information retrieval community has focussed on random walk based sampling algorithms like PageRank [23], HITS [14] and other variations to explore the WWW and rank Web pages. The Internet modeling research community [15, 7, 4] has used different sampling methodologies to reduce the power-law based Internet topologies for faster simulations. Thus, sampling graphs has received significant attention in the past.

In the domain of social network research, however, graph sampling has been less studied due to unavailability of large data and privacy concerns. So people studied auxiliary social networks such as citation networks, affiliation networks etc. Leskovec *et al.* in [19] proposed sampling algorithms to produce samples that match the temporal evolution of the underlying social network. Hubler *et al.* in [12] proposed Metropolis graph sampling based on the idea to compute properties of the original graph for the actual sampling step. Eldardiry *et al.* in [9] proposed a method for resampling from a graph using a subgraph sampling approach to preserve the local relational dependencies while generating a pseudosample with sufficient global variance. Due to the popularity of online social networks such as Facebook [1] and Twitter [3], there has been a lot of work [22, 20, 18, 16, 5, 6] studying the growth and evolution of these online social networks. While most of them have been on static graphs, recent works [27, 26] have started focusing on interactions in social networks.

7. CONCLUSIONS

Online social networks have enjoyed tremendous success and popularity over the years, and have become a major platform for communication activity. The huge size of these networks, however, makes it extremely hard to study and analyze the structure of social network activity graphs. While sampling algorithms can potentially be used to generate representative samples of the original graphs, but most existing algorithms either do not preserve properties of the original graph, or are not suitable for a streaming implementation, which is a requirement since communication activity naturally follows a stream model. We proposed a new sampling algorithm STNS that exploits several key observations from an extensive study of Facebook and Twitter data. We have demonstrated that STNS performs significantly better than previous algorithms in several canonical point statistics as well as across distributions consistently across both data sets.

There are several future directions we wish to with this work. First, we have briefly mentioned in the paper that densification laws proposed by prior researchers appear to not hold when we consider activity within a time window. We wish to empirically validate this phenomenon extensively across data sets. Second, we wish to extend STNS to sample evolving graphs, for example, by placing time window constraints (collecting edges only for a certain time and aging out users).

Acknowledgements

We thank Kyle Bowen for providing us with the Twitter data and Indika Kahanda for his help in Facebook data analysis.
This research is supported by DARPA, ARO, NSF under contract number(s) NBCH1080005, W911NF-08-1-0238, and IIS-0916686. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, ARO, NSF, or the U.S. Government.

8. REFERENCES

[1] Facebook. `http://www.facebook.com/`.

[2] Myspace. `http://www.myspace.com/`.

[3] Twitter. `http://www.twitter.com/`.

[4] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore. On the bias of traceroute sampling: or, power-law degree distributions in regular graphs. In *ACM STOC*, pages 694–703, 2005.

[5] Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *WWW*, pages 835–844, 2007.

[6] H. Chun, H. Kwak, Y. Eom, Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in volume vs interaction: a case study of cyworld. In *ACM/USENIX IMC*, pages 57–70, 2008.

[7] L. Dall 'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani. Exploring networks with traceroute-like probes: Theory and simulations. *Theoretical Computer Science*, 355(1):6–24, 2006.

[8] S. Datta and H. Kargupta. Uniform data sampling from a peer-to-peer network. In *Proceedings of ICDCS'02*, page 50, 2007.

[9] H. Eldardiry and J. Neville. A resampling technique for relational data graphs. In *SNA-KDD'08: Proceedings of the second workshop on Social Network Mining and Analysis*, 2008.

[10] Facebook. Chat reaches 1 billion messages sent per day. `http://www.facebook.com/note.php?note_id=91351698919`, 2009.

[11] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *IEEE INFOCOM*, 2004.

[12] C. Hubler, H.-P. Kriegel, K. M. Borgwardt, and Z. Ghahramani. Metropolis algorithms for representative subgraph sampling. In *ICDM*, 2008.

[13] I. Kahanda and J. Neville. Using transactional information to predict link strength in online social networks. In *AAAI Conference on Weblogs and Social Media*, 2009.

[14] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[15] V. Krishnamurthy, M. Faloutsos, M. Chrobak, J. Cui, L. Lao, and A. Percus. Sampling large Internet topologies for simulation purposes. *Computer Networks*, 51(15):4284–4302, 2007.

[16] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *SIGKDD*, pages 611–617, 2006.

[17] S. Lee, P. Kim, and H. Jeong. Statistical properties of sampled networks. *Physical Review E*, 73:016102, 2006.

[18] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *SIGKDD*, 2008.

[19] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *SIGKDD*, pages 631–636, 2006.

[20] J. Leskovec and E. Horvitz. Worldwide Buzz: Planetary-Scale Views on an Instant-Messaging Network. In *WWW*, 2008.

[21] J. Leskovec, J. M. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *SIGKDD*, pages 177–187, 2005.

[22] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *ACM/USENIX IMC*, 2007.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1998.

[24] M. Stumpf, C. Wiuf, and R. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102(12):4221–4224, 2005.

[25] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. On unbiased sampling for unstructured peer-to-peer networks. In *IMC*, pages 27–40, 2006.

[26] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *WOSN*, August 2009.

[27] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *EuroSys*, pages 205–218, 2009.

[28] S. Yoon, S. Lee, S.-H. Yook, and Y. Kim. Statistical properties of sampled networks by random walks. *Phys. Rev. E*, 75(4):046114, Apr 2007.