Crawling Online Social Graphs

Shaozhi Ye
Department of Computer Science
University of California, Davis
sye@ucdavis.edu

Juan Lang
Department of Computer Science
University of California, Davis
jilang@ucdavis.edu

Felix Wu
Department of Computer Science
University of California, Davis
wu@cs.ucdavis.edu

***Abstract*—Extensive research has been conducted on top of online social networks (OSNs), while little attention has been paid to the data collection process. Due to the large scale of OSNs and their privacy control policies, a partial data set is often used for analysis. The data set analyzed is decided by many factors including the choice of seeds, node selection algorithms, and the sample size. These factors may introduce biases and further contaminate or even skew the results. To evaluate the impact of different factors, this paper examines the OSN graph crawling problem, where the nodes are OSN users and the edges are the links (or relationship) among these users. More specifically, by looking at various factors in the crawling process, the following problems are addressed in this paper:**

- **Efficiency: How fast different crawlers discover nodes/links;**
- **Sensitivity: How different OSNs and the number of protected users affect crawlers;**
- **Bias: How major graph properties are skewed.**

To the best of our knowledge, our simulations on four real world online social graphs provide the first in-depth empirical answers to these questions.

I. Introduction

Online social networks (OSNs) have become a major Internet service in the past few years. Their popularity is highlighted by the millions of users they attract and the huge amount of interactions they boost among these users. For example, Facebook was ranked as the 4th most visited website in the world as of Aug. 2009 [1] with "more than 300 million active users." [2] This global phenomenon has generated lots of interest in many disciplines including sociology and computer science.

Extensive work has been conducted on top of OSNs, in many cases using partial social networks. There are several reasons partial social networks are used. First of all, social data is among the most valuable assets to the OSN providers and is protected by privacy regulations/laws, therefore it is hard to get such data directly from the OSN providers. Secondly, it is a great challenge for crawlers to collect millions of contact lists, profiles, pictures, videos, etc. from OSNs. Many OSNs use a large number of dynamic pages containing AJAX and DHTML effects, and it is not trivial to develop a parser to deal with such complex pages efficiently. Moreover, OSN users are often provided with the flexibility to customize the layout of their pages, which further complicates the design and implementation of the parser. To make things worse, rate limiting is enforced by most OSNs, preventing crawlers from making many requests within a short period of time. Thirdly, as more users become concerned about their privacy in OSNs, many of them choose not to reveal their information to strangers, hence become "black holes" for crawlers.

Although partial datasets have to be used, most previous work does not provide detailed analysis on how the data collection process affects their observations/conclusions. While there has been some research on sampling social graphs [1], [2], most of them assume some prior knowledge of the underlying social networks, which is not available for crawlers in reality. Moreover, it is not clear whether their conclusions on traditional social graphs such as affiliation networks apply to OSNs as well. This paper examines the OSN data collection problem from the crawler's perspective, by evaluating different crawlers with four real world OSN graphs. More specifically, the social graph crawling problem investigated here consists of the following three issues.

- **Efficiency**: How fast nodes/links are discovered through a crawl;
- **Sensitivity**: How OSNs and "black hole" users affect the crawling;
- **Bias**: How statistical properties of the crawled subgraphs are different from those of the whole graph.

The factors we evaluate in this paper include:

- **Choice of seeds**: Seeds are the starting point of a crawl. For Web crawling, it is important to select proper seeds and order the crawling queue to avoid low quality pages [3]. Our experiments show that the small world effect of OSNs makes the choice of seeds less critical.
- **Node selection algorithms**: Node selection algorithms decide which node to crawl next. In this paper we evaluate four node selection algorithms, including the widely used breadth-first search (BFS) and greedy algorithms.
- **Protected users**: There are concerns that as more and more users adopt the access controls to protect their social data, the crawlers may miss a large portion of the social graph. To complement existing social network resilience analysis [4], we show how different crawlers perform in the presence of protected users on real OSNs.
- **Different OSNs**: Different OSNs have their unique properties even though they provide similar services. In this paper we evaluate different crawlers with four OSN graphs collected by Mislove et al. [5]: Flickr, LiveJournal,

[1] http://en-us.nielsen.com/rankings/insights/rankings/internet
[2] http://www.facebook.com/home.php#/press/info.php?statistics

Orkut, and YouTube.

A major contribution of this paper is to formalize how we evaluate the crawling bias and what parameters need to be considered. The two metrics we examined here, mean degree and clustering coefficient, are fundamental statistics for graph analysis. As the first paper to investigate the social graph crawling problem, we believe that it is important to look at them carefully. We are aware of many interesting metrics such as average path length, diameter, and power law parameters while choose not to report them in this paper due to the page limit and computation cost.

The rest of this paper is organized as follows. Section II briefly reviews the prior work on crawling and sampling social graphs. Section III formally introduces the social graph crawling problem and the node selection algorithms tested in this paper. Section IV discusses the simulation setup and the factors we evaluate. Section V and VI presents the observations we get in our experiments. Finally, we describe future work and conclude in Section VII.

II. Related Work

In this section, we review the prior work on social network crawling and social graph sampling.

Despite the huge number of social network publications, few have been dedicated to the data collection process. Chau et al. [6] briefly describe using a parallel crawler running BFS to crawl eBay profiles quickly. The measurement conducted by Mislove et al. [5] is, to the best of our knowledge, the largest OSN crawling study ever published. From four popular OSNs, Flickr, YouTube, LiveJournal, and Orkut, 11.3M users and 328M links are collected. Their analysis confirms most well known properties of OSNs, such as a power-law degree distribution, a densely connected core, strongly correlated in-degree and out-degree, and small average path length.

Most studies are based on subgraphs, thus it is important to know how similar the sampled subgraphs and the original graphs are. Leskovec and Faloutsos [1] evaluate many sampling algorithms such as random node, random edge, and random jump. The datasets used in [1] are citation networks, autonomous systems, the arXiv affiliation network, and the network of trust on epinions.com, the largest of which consists of 75K nodes and 500K edges. These are of much smaller scale than the datasets we examine in this paper. More importantly, none of the networks studied in [1] is an OSN. epinions.com may be the closest to an OSN, but as a review website its main subject is products instead of users and it generates little interaction between users. Many sampling methods considered in [1] require some knowledge of the original graph. For example, the random edge sampling method needs to select an edge at random, which is not supported by most OSNs. The random PageRank node sampling method needs to know the PageRank of a node in advance, while to compute the PageRank, one needs to know the whole graph.

Ahn et al. [7] obtain the complete network of a large South Korean OSN site named CyWorld directly from its operators. They evaluate the *snowball* sampling method (which is in fact breadth-first search) on this 12M node, 190M edge graph. Their results indicate that a small portion ($< 1\%$) of the original network sampled in snowball fashion approximates some network properties well, such as degree distribution and degree correlation, while accurate estimation of clustering coefficient is hard even with 2% sampling. We revisit the estimation of clustering coefficient in Section VI-B.

Gjoka et al. [8] propose a sampling method to select nodes uniformly without knowledge of the entire network, and use this method on a large sample (1M node) of the Facebook graph. The basic idea is that given the current node u, randomly select one of its neighbors v, move the random walker to v with probability $\min(1, k_u/k_v)$, where k_u and k_v are the degrees of u and v respectively. They compare their method with BFS and simple random walk to show it generates a more uniform node sample. Based on this method, it is possible to better estimate clustering coefficient than the estimator in [1]. Evaluating this estimator with the same setting we have here will be an interesting direction for future work. It is also worthwhile to note that, to decide whether to move the random walker, it needs to know k_v, which requires to crawl v in general. Thus when evaluating its cost, we need to take this into account.

The link privacy problem raised by Korolova et al. [9] concerns how an attacker discovers the social graph. The goal of the attacker is to maximize the number of nodes/links it can discover given the number of users it *bribes* (crawls). Several attacks evaluated in [9] actually correspond to node selection algorithms for crawling, such as BFS and greedy attacks. The same problem is considered by Bonneau et al. [10]. They first collect a nearly complete subset of the Facebook network consisting of 15K Stanford students, then test several crawlers including a greedy crawler, a random crawler, and a targeted attack of highest degree nodes (assuming the degrees are known ahead of time.) They show that the targeted attack is most efficient, the random crawler is nearly so, while the greedy crawler isn't as efficient as the other two. Maximizing the number of victims is the sole objective in [9] and [10] therefore neither of them examines other issues such as biases.

III. Crawling Social Graphs

Table I summarizes the list of notations used in this paper.

TABLE I
List of Symbols

Notation	Definition
$\text{Out}(v)$	The set of nodes which are linked by v (directed graph).
V_{Seen}	The set of nodes which are found by the crawler.
V_{Crawled}	The set of nodes which are crawled by the crawler.
E_{Seen}	The set of links which are found by the crawler.

A. Definition

An OSN can be modeled as a graph with users as nodes and the relationship between users as edges. This paper focuses on how to crawl this social graph, which can be naturally divided into crawling nodes and crawling edges.

Crawling nodes: There is often extra information associated with a node, such as personal information (profile), photos, posts, and list of friends. The crawler must crawl a node to collect such information, whereas the crawler may become aware of the existence of a node without having crawled it. The difference is captured in V_{Crawled} and V_{Seen}: V_{Crawled} is the set of crawled nodes, whereas V_{Seen} is the set of nodes of whose existence the crawler is aware.

Crawling edges: While we distinguish between crawled and seen nodes, we do not have such distinguish between crawled and seen edges. First of all, few edge attributes are provided by most OSNs, such as when the edge is created and how a node categorizes this edge (friends, classmates, etc.). Secondly, these attributes often come with the list of multiple edges directly instead of requiring the crawler to ask for a particular edge. Therefore in many cases there is either no edge attributes to crawl or no way to crawl a specific edge. Hence we do not consider E_{Crawled} in this paper.

The process for crawling a graph can be outlined as follows.

1) Put seeds into a queue.
2) Select a node v from the queue.
3) Crawl the node.
4) Add the newly found nodes in Out(v) into the queue.
5) Go to Step 2 or terminate if stop conditions are met.

B. What decides the crawled subgraphs

Given an OSN, the partial graph crawled is decided by the following three factors.

- Seeds
- Node selection algorithm
- Size of the crawled subgraph

Seeds are where the crawler starts. In reality we have to rely on the recommendation service provided by OSNs or use some manually collected/voluntarily contributed seeds, neither of which is feasible for getting large number of seeds.

The size of the crawled subgraph is decided by how fast the crawler can go and when the crawler stops. It is subject to real world resource constraints such as network bandwidth, time, machines, and the rate limits enforced by OSN providers.

Node selection algorithms decide which node to select from the crawling queue. Various information such as locations and occupations is available for sophisticated node selection algorithms while here we focus on the graph crawling problem, i.e. only the abstraction of nodes and edges is considered.

There are lots of node selection algorithms, many of which fit better in the context of graph sampling, such as random node and random jump in [1]. In this paper, we focus on those which are designed for crawlers. More specifically, such a node selection algorithm needs to satisfy the following requirements.

- No prior knowledge on the graph is needed. For example, selecting nodes uniformly at random (random node in [1]) requires the knowledge of the entire graph.
- For fair comparison on efficiency, no crawled node can be discarded. The uniform sampling method proposed by Gjoka et al. [8] may choose to reject a node after examining its degree, therefore is not considered here.

These constraints can be relaxed in some cases. For example, when updating or re-crawling a graph, we have some knowledge from the previous crawl. Crawled nodes may also be discarded when we are willing to trade bandwidth for certain sampling properties, although sampling over a crawled graph is more common since it allows applying different sampling methods to the same graph for multiple purposes.

In this paper, we consider the following four node selection algorithms, which are widely used in practice.

- **BFS**: Simply selecting the first item in the queue, breadth-first search is probably the most popular one.
- **Greedy**: The crawler selects the node with the largest degree in the queue. Since the nodes in the queue are not crawled yet, their degrees on the crawled subgraph $G(V_{\text{Seen}}, E_{\text{Seen}})$ are used.
- **Lottery**: The crawler selects a node in the queue with probability proportional to its degree. This algorithm prefers nodes with large degrees, while also selecting nodes with small degrees to reduce sampling bias. Similar to the greedy algorithm, the degree here is computed on the crawled subgraph.
- **Hypothetical greedy**: The crawler always selects the node with largest degree in the queue, while the *degree* here is the degree on the whole graph, i.e. the true degree. A typical application scenario for this algorithm is to sample a subgraph from a large graph. Or we can assume an algorithm which makes accurate estimation of which node in the queue has the largest degree based on various information, such as the crawled partial graph and user profiles. This algorithm serves as a baseline in this paper.

IV. Simulation Setup

A. Summary of Data Sets

Instead of using synthetic datasets generated by small world models, we use four real world social graphs collected by Mislove et al. [5]. These four OSNs have different focuses. Flickr specializes in photo sharing, YouTube focuses on video sharing, and LiveJournal and Orkut are general social websites. Table II summaries the basic statistics of these four graphs.

TABLE II
Basic statistics of the social graphs used in this paper

Graph	Total Nodes	Total Links	Mean Degree	Clustering Coefficient
Flickr	1,657,846	22,613,981	13.6	0.209
LiveJournal	4,929,069	77,402,652	15.7	0.278
Orkut	3,072,441	223,534,301	72.8	0.164
YouTube	1,099,764	4,945,382	4.5	0.098

Being published in 2007, these four graphs have been widely used in OSN studies. As of Oct. 2009, Google Scholar reports 168 citations to [5], many of which either analyze this data set directly or refer to conclusions based on it. Therefore it is important to take a careful look at this data.

B. Parameters to Investigate

Here are the factors we consider in this paper.

- **Social graphs**: The same crawler may exhibit different behaviors on different social graphs.
- **Choice of seeds**: We want to know how critical the number of seeds and their degrees are.
- **Crawling size**: We are interested in how different crawlers behave as more and more nodes are crawled.
- **Number of protected users**: We want to evaluate how different crawlers explore the graph in the presence of protected users.
- **Node selection algorithms**: We evaluate the four algorithms discussed in Section III-B.

The combination of these factors generates a large problem space to explore. Moreover, multiple tests are required to produce reliable results. With a cluster of 36 PCs (two AMD Opteron 2.6GHz CPU/4GB memory per node), it cost us three weeks to finish all the simulations with various data structure and algorithm optimizations.

V. Crawling Efficiency

Node coverage and link coverage are important indicators for how well the crawler is able to find new nodes/links. Given the number of crawled nodes, they are defined as follows [9].

- **NC** (node coverage): $\frac{|V_{\text{Seen}}|}{|V|}$, i.e. the number of nodes seen by the crawler versus the number of nodes in the graph.
- **LC** (link coverage): $\frac{|E_{\text{Seen}}|}{|E|}$, i.e. the number of links seen by the crawler versus the number of links in the graph.

Node/link coverage is also a metric for how close the subgraph $G(V_{\text{Seen}}, E_{\text{Seen}})$ is to the whole graph $G(V, E)$. When $G(V, E)$ is fixed, the node/link coverage is a monotonic function of the number of nodes being crawled. As the entire graph being crawled, the node/link coverage approaches 1. In many cases where it is not feasible to crawl the whole graph, maximizing the node/link coverage is usually one objective for the partial crawling, i.e. with the same number of nodes being crawled, find as many nodes/links as possible.

This section focuses on how efficient different crawlers are in terms of node/link coverage.

A. Crawling on Different Social Graphs

The online social graphs studied in this paper are different in terms of their size, the services provided, user base, and definition of *friends*. It is interesting to see that in many cases there is little difference between the results obtained from these graphs, which is a good indicator for applying our findings here to other online social graphs. Due to the page limit, we omit the similar results and indicate the outliers when necessary.

Mislove et al. [5] crawled these graphs with BFS, which inevitably introduces biases. Fortunately the LiveJournal sample is the large weakly connected component from LiveJournal and covered 95.4% of the entire LiveJournal network in 2006 when it is crawled, therefore its results should be less biased. Interestingly, although the crawled population is 26.9% and unknown for Flickr and YouTube respectively, their results are similar to those of LiveJournal. This indicates that our conclusions here are not heavily affected by the bias in the source data. The Orkut graph, on the other hand, consists of 11.3% nodes on the Orkut network and is the only strongly connected graph of the four, which becomes the single outlier in some cases.

B. Choice of Seeds

As we argued in Section III-B, the choice of seeds is limited in practice. Here we consider two criteria, the number of the seeds and the degree of the seeds.

To see how the number of seed nodes affects the crawler, we randomly select 100, $1,000$, and $10,000$ nodes as seeds, and run different crawlers with the same set of seeds.

With multiple tests, we find that the difference is small for all four graphs, even when only 10% of the graph is crawled. Furthermore, such difference diminishes quickly as more nodes are crawled. This indicates that OSNs are tightly coupled compared to other graphs such as Web.

To see if starting from nodes with large degrees improves the node/link coverage, we select random seeds with different minimum degree requirements ranging from 10 to 60. With BFS, the improvement is less than 5% for both node and link coverages. With node selection algorithms such as greedy and hypothetical greedy, the improvement is even smaller.

Implication 1: Node/link coverage is sensitive to neither the number of seeds nor the degree of seeds.

Unless explicitly specified, the results reported in the rest of this section are all based on 100 random seeds with no minimum degree requirements.

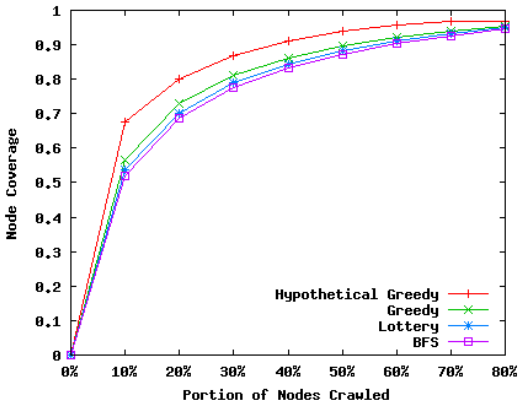
C. Node Selection Algorithms

Figure 1 shows the results of crawling on the LiveJournal graph. With 10% nodes crawled, all crawlers are able to discover more than 55% nodes and 30% links. It is a strong sign of the small world phenomenon: lots of nodes are tightly coupled together within a few hops of each other, therefore crawling a small portion of the network is sufficient to reveal most nodes/links.

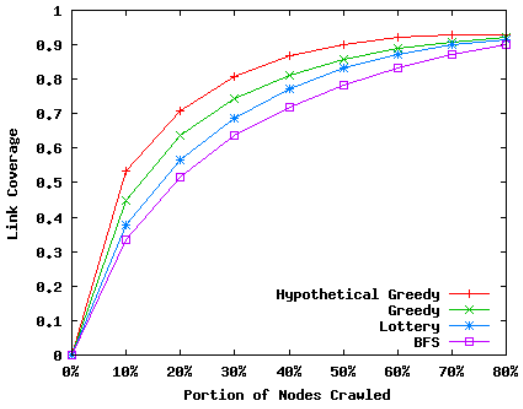
Experiments on YouTube and Flickr show similar results. For node/link coverage, we have hypothetical greedy $>$ greedy $>$ lottery $>$ BFS, although the difference between them is small. The only outlier to this order is node coverage on Orkut (link coverage on Orkut is consistent with others). Shown as Figure 2, BFS gets node coverage close to the hypothetical greedy crawler, and outperforms both lottery and greedy crawlers by $10\% - 30\%$ when 10% of nodes are crawled. Investigating the graph structures which fail greedy crawlers is an interesting direction for future work.

Implication 2: Greedy crawlers are likely to get higher node/link coverage, while BFS crawlers are more robust.

We further examine the results for BFS with 10% nodes crawled, shown as Table III. The largest node coverage is found on the Orkut graph, which is probably a result of its small radius and diameter [5]. Since the mean degree of the



(a) Node coverage



(b) Link coverage

Fig. 1. Node/link coverage of different node selection algorithms (LiveJournal).

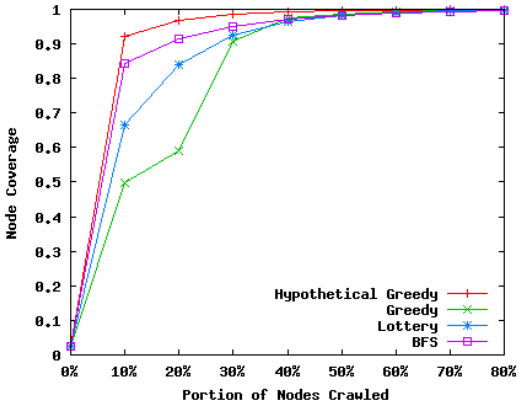


Fig. 2. Node coverage on Orkut graph.

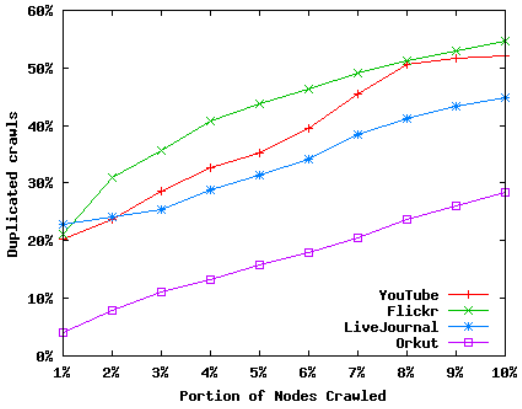


Fig. 3. Duplicate crawls when crawling in parallel without coordination.

crawled nodes, shown as Figure 6, is close to that of the original graph, the low link coverage suggests that there are many links between the uncrawled nodes. The results on Flickr and LiveJournal are counterintuitive. LiveJournal has a slightly larger mean degree (15.7) than Flickr (13.6). Although the crawler gets a larger node coverage on LiveJournal (52.0% vs 49.0%), it gets a much smaller link coverage (33.3% vs 72.7%). This indicates that the Flickr graph has a much smaller, tightly coupled core compared to the LiveJournal graph.

TABLE III
NODE AND LINK COVERAGE AFTER CRAWLING 10% NODES (BFS).

Graph	Node coverage	Link coverage
Flickr	49.0%	72.7%
LiveJournal	52.0%	33.3%
Orkut	77.4%	25.8%
YouTube	57.8%	51.7%

Implication 3: Crawling a small number of nodes is sufficient to discover a large portion of the OSN.

This observation has a direct impact on parallel crawlers. Figure 3 shows the percent of duplicated crawls when running 4 crawlers in parallel without coordination. With only 10% of the network being crawled, $20\% - 50\%$ crawls are wasted on the nodes we have crawled before. The percent of duplicated crawls increases when a larger portion of the network is crawled. In most cases the crawling rate is limited by the social network service, increasing the impact of duplicated crawls.

Implication 4: When crawling an OSN in parallel, coordination between crawlers is required to avoid huge amount of duplicate crawls.

D. Number of Protected Users

As many users being aware of the OSN privacy issues, some users choose to make their profiles, posts, contact lists, etc. visible to a limited number of users, such as their friends or the users within a certain social group. These users appear on the social graph as black holes from the crawler's perspective. Here we perform initial investigations of how these nodes impede the crawling process.

To address the impact of protected users, we compute the change of node coverage: Given a graph G, $\Delta NC = NC - NC'$, where NC' is the node coverage when a certain number of nodes on G are protected. Similarly we have ΔLC.

On the YouTube graph we randomly select 100K nodes to be protected, i.e. their contact lists can not be seen by the crawler. Shown as Figure 4, the node/link coverage drops less than 7% for all four crawlers with 9% nodes being protected.

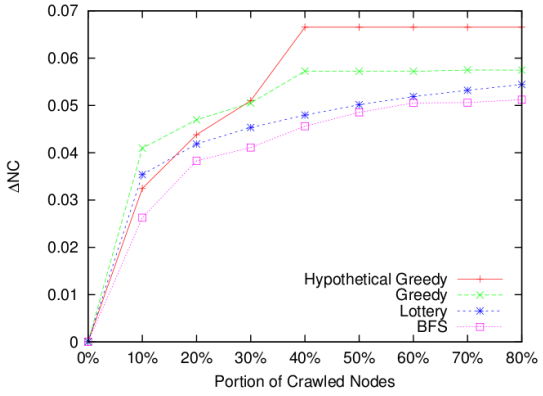


Fig. 4. Crawling in the presence of 100K protected users. (YouTube)

We have similar observations on other graphs. Shown as Table IV, crawlers can easily get around the black holes caused by protected users, therefore the decrease in node/link coverage is small compared to the fraction of protected users on these graphs. Crawling on larger OSN graphs (LiveJournal and Orkut) is more resilient to protected users. Experiments with greedy and lottery crawlers verify this observation too.

TABLE IV
ΔNC, ΔLC WITH 200K PROTECTED USERS (BFS)

Graph		Flickr	LiveJournal	Orkut	YouTube
Users being protected		12.1%	4.1%	6.5%	18.2%
10% users crawled	ΔNC	3.3%	0.6%	0.3%	6.4%
	ΔLC	8.8%	1.4%	2.1%	9.4%
80% users crawled	ΔNC	6.4%	0.9%	0.2%	10.5%
	ΔLC	10.8%	3.6%	6.1%	15.4%

Implication 5: A small portion of protected users does not hurt the node/link coverage of OSN crawlers especially for large social graphs.

VI. CRAWLING BIAS

A. Mean Degree

If we consider crawled subgraphs for the LiveJournal graph only, we have the mean degree shown in Figure 5. It is not surprising to see that the mean degree reported by each crawler follows the order of hypothetical greedy > greedy > lottery > BFS: the hypothetical greedy crawler and the greedy crawler choose higher degree nodes over lower degree ones. BFS chooses randomly, but the high betweenness of high degree nodes means they get picked with higher probability than low-degree ones [11]. The lottery crawler lies somewhere between the two extremes. Naturally the differences between these crawlers, as more nodes are crawled, becomes small.

Figure 6 shows the estimates of mean degree from BFS on each of the graphs. The initial point, with 0% of the nodes crawled, is based on the degree of the seeds. The seeds are

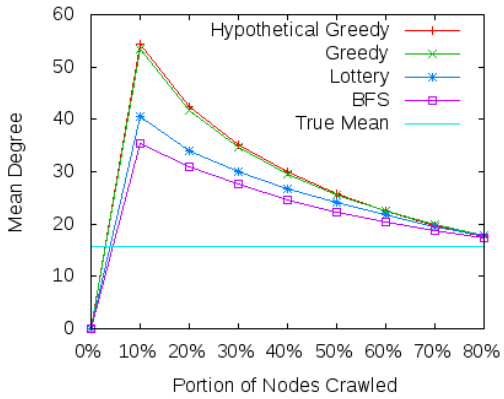


Fig. 5. Mean degree *vs.* different node selection algorithms. (LiveJournal)

chosen at random, and therefore represent a random sample of the graph. For the Flickr and YouTube graphs, the initial degree estimate isn't radically different from the true mean degree, and the degree estimates based on crawled subgraphs remain relatively accurate. The Orkut and LiveJournal graphs are rather different, however. The LiveJournal estimate of mean degree based on randomly chosen nodes is relatively accurate, but as more of the network is crawled, the estimate grows rapidly, reaching a peak when 10% of the graph is crawled. The estimate then steadily declines as more of the graph is crawled, yet doesn't approaching the true mean degree until a significant portion of the network has been crawled. The Orkut estimate of mean degree based on randomly chosen seeds is far lower than the mean degree of the graph. As the graph studied here is a 11.3% sample crawled by BFS, the true mean degree of the original Orkut network is likely to be smaller (thus closer to the mean degree estimated by the seeds) due to the bias BFS has toward high degree nodes.

Implication 6: Only by crawling a substantial portion of the network is the degree estimate guaranteed to be accurate.

To get better degree estimation with small amount of crawls, a sampling over the crawled subgraph is required, such as the random node sampling we use for selecting seeds and the sampling method proposed by Gjoka et al. [8].

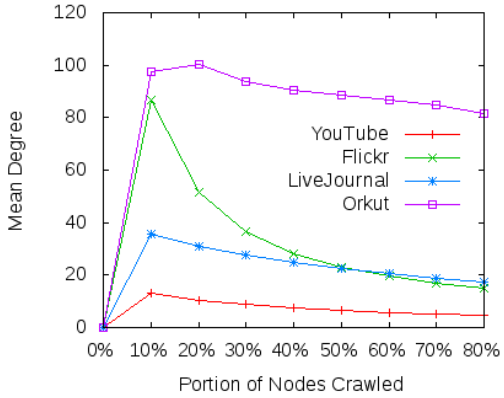


Fig. 6. Mean degree on different social graphs. (BFS)

B. Clustering Coefficient

Shown as Figure 7, clustering coefficient increases as the crawler completes the network except for the YouTube graph, where the clustering coefficient remains stable. The underestimation of clustering coefficient comes from oversampling nodes with large degrees. Given a node v, its clustering coefficient is defined as $\frac{|\{e_{i,j}\}|}{k_v(k_v-1)}$, where $i, j \in \text{Out}(v)$ and k_v is the degree of v [12]. In other words, it is the number of links between v's neighbors versus the maximum possible number of such links. Hence oversampling large degree nodes often leads to small clustering coefficients, which is also reported by Ahn et al. [7].

The accurate estimate of clustering coefficient on YouTube indicates that high degree nodes have their neighbors tightly connected. In other words, the YouTube graph exhibits a stronger "birds of a feather flock together" effect.

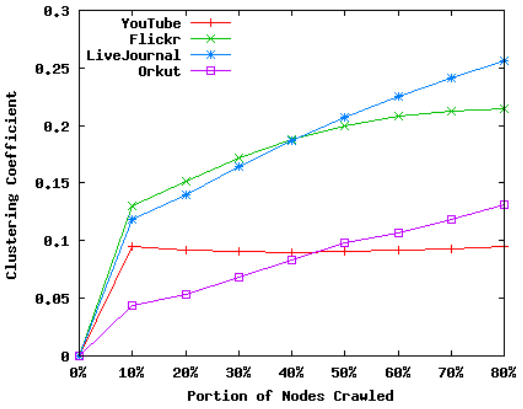


Fig. 7. Clustering coefficient on different social graphs. (BFS)

Experiments on different crawlers with the same graph show little difference on clustering coefficient. For all four graphs, hypothetical greedy crawlers result in the largest clustering coefficients, followed by greedy crawlers, then lottery crawlers. BFS crawlers always give the smallest ones.

Implication 7: Clustering coefficients are easily underestimated.

VII. Conclusions and Future Work

Attempting to crawl the entire graph is tempting, because more data usually leads to better analysis. As online social networks grow in size, the practicality of crawling the entire graph decreases. This paper investigates the influences of seeds, sample size, node selection algorithms, and the graph being crawled. We believe that the implications we conclude here shed light on future OSN studies, which will increasingly rely on crawled subgraphs. The inconsistency across different crawlers and graphs, as well as our bias analysis, suggests future studies to pay more attention to the data collection process and the introduced biases to make their conclusions applicable to more general networks.

In the future, we plan to evaluate other OSN graphs we have crawled entirely. We also plan to make these graphs available to the social network research community.

An interesting question we did not consider in sufficient detail is where the transition between high degree nodes and low degree nodes is. For example, in the LiveJournal crawl, the estimate of mean degree rises dramatically initially, and declines almost equally dramatically. Investigating this transition is a topic of future work.

Another challenging problem is how to update a crawled subgraph efficiently. As OSNs grow rapidly, nodes may join or leave the network, and links may also be created or deleted. Identifying nodes whose statuses may have changed recently saves bandwidth and makes it possible to track the dynamics of larger graphs. One area of ongoing work is applying link prediction [13] to estimate how frequently a node changes its status in order to schedule the crawling to maximize the freshness of the crawled subgraphs.

Acknowledgements

This research is funded in part by National Science Foundation under CNS-0832202. The authors would like to thank Alan Mislove for sharing with us the social graphs for analysis and Matt Spear for his valuable comments.

References

[1] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *KDD'06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 631–636.

[2] S. H. Lee, P.-J. Kim, and H. Jeong, "Statistical properties of sampled networks," *Phys. Rev. E*, no. 73, p. 016102, 2006.

[3] J. Cho, H. Garcia-Molina, and L. Page, "Efficient crawling through URL ordering," in *WWW'98: Proceedings of the seventh international conference on World Wide Web*, 1998, pp. 161–172.

[4] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: Percolation on random graphs," *Phys. Rev. Lett.*, vol. 85, no. 25, pp. 5468–5471, Dec 2000.

[5] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *IMC'07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 29–42.

[6] D. H. Chau, S. Pandit, S. Wang, and C. Faloutsos, "Parallel crawling for online social networks," in *WWW'07: Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 1283–1284.

[7] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *WWW'07: Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 835–844.

[8] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Unbiased sampling of facebook," 2009.

[9] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu, "Link privacy in social networks," in *CIKM'08: Proceeding of the 17th ACM conference on Information and knowledge management*, 2008, pp. 289–298.

[10] J. Bonneau, J. Anderson, and G. Danezis, "Prying Data out of a Social Network," *First International Conference on Advances in Social Networks Analysis and Mining*, 2009.

[11] M. E. J. Newman, "A measure of betweenness centrality based on random walks," *Social Networks*, vol. 27, no. 1, pp. 39–54, 2005.

[12] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[13] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *CIKM'03: Proceedings of the twelfth international conference on Information and knowledge management*, 2003, pp. 556–559.